

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, ...
- PhD project: mostly in R
- 1st Postdoc: bash, R
- Current postdoc: R, bash, Python



princessMÁXIMA
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

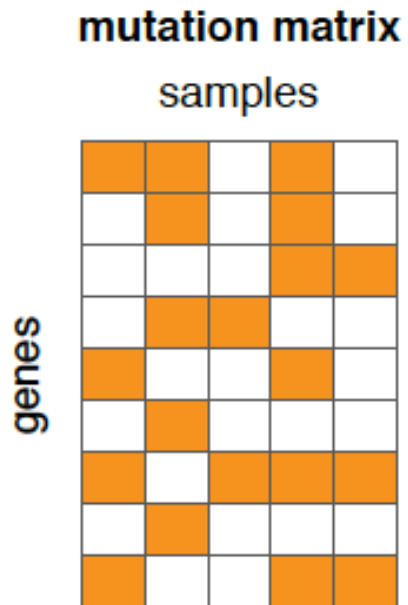
- Genetic interaction/epistasis:

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

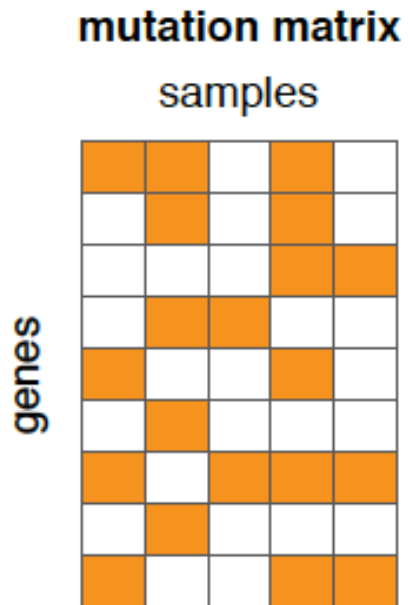
Methods to detect genetic interactions

- Create **Sample - Gene** matrix



Methods to detect genetic interactions

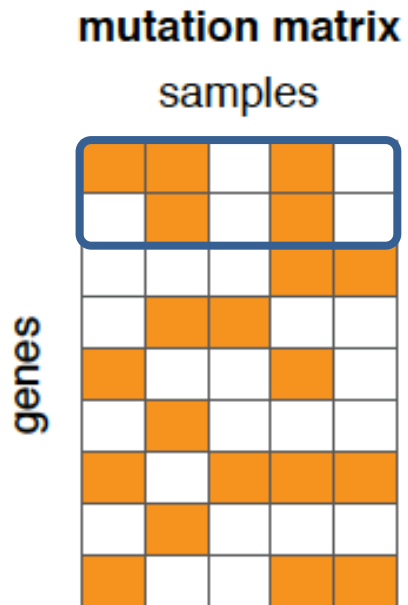
- Create **Sample - Gene** matrix
- Count per gene pair the number of **co-occurring** alterations



Orange	Orange	White	Orange	White
White	Orange	White	Orange	White
White	White	White	Orange	Orange
White	Orange	Orange	White	White
Orange	White	White	Orange	White
White	Orange	White	White	White
Orange	White	Orange	Orange	Orange
White	Orange	White	White	White
Orange	White	White	Orange	Orange

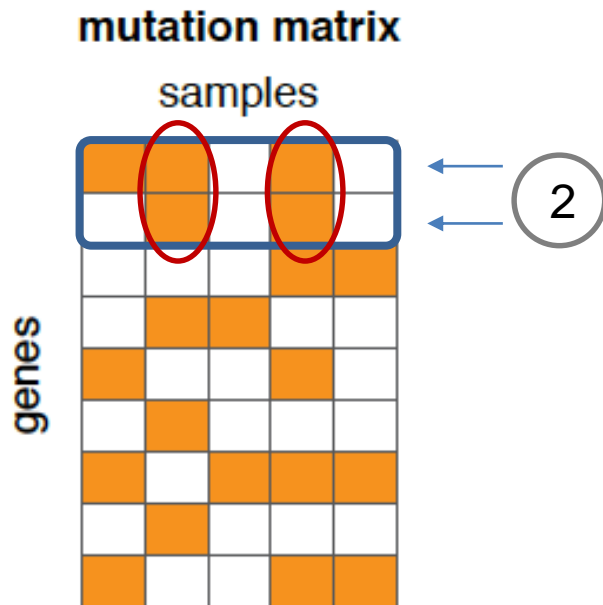
Methods to detect genetic interactions

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



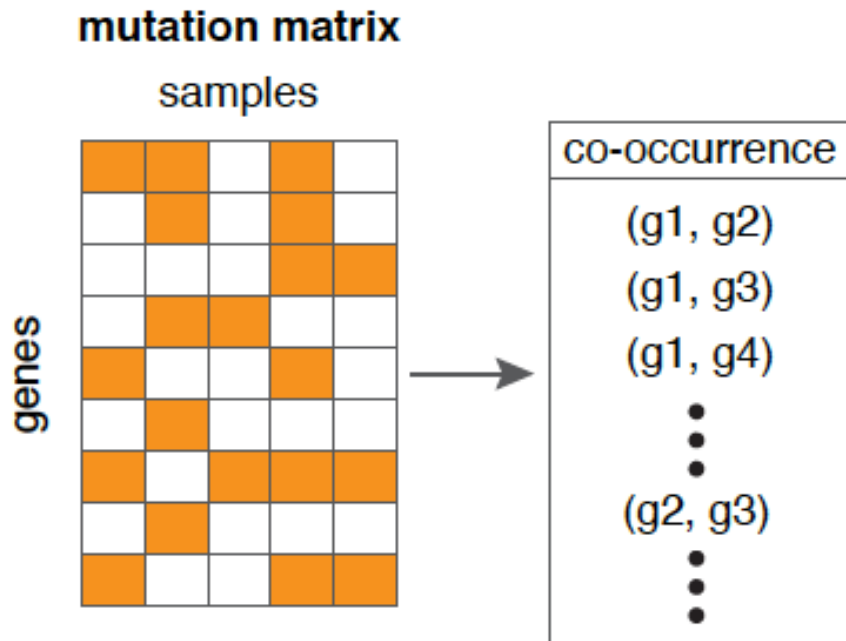
Methods to detect genetic interactions

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



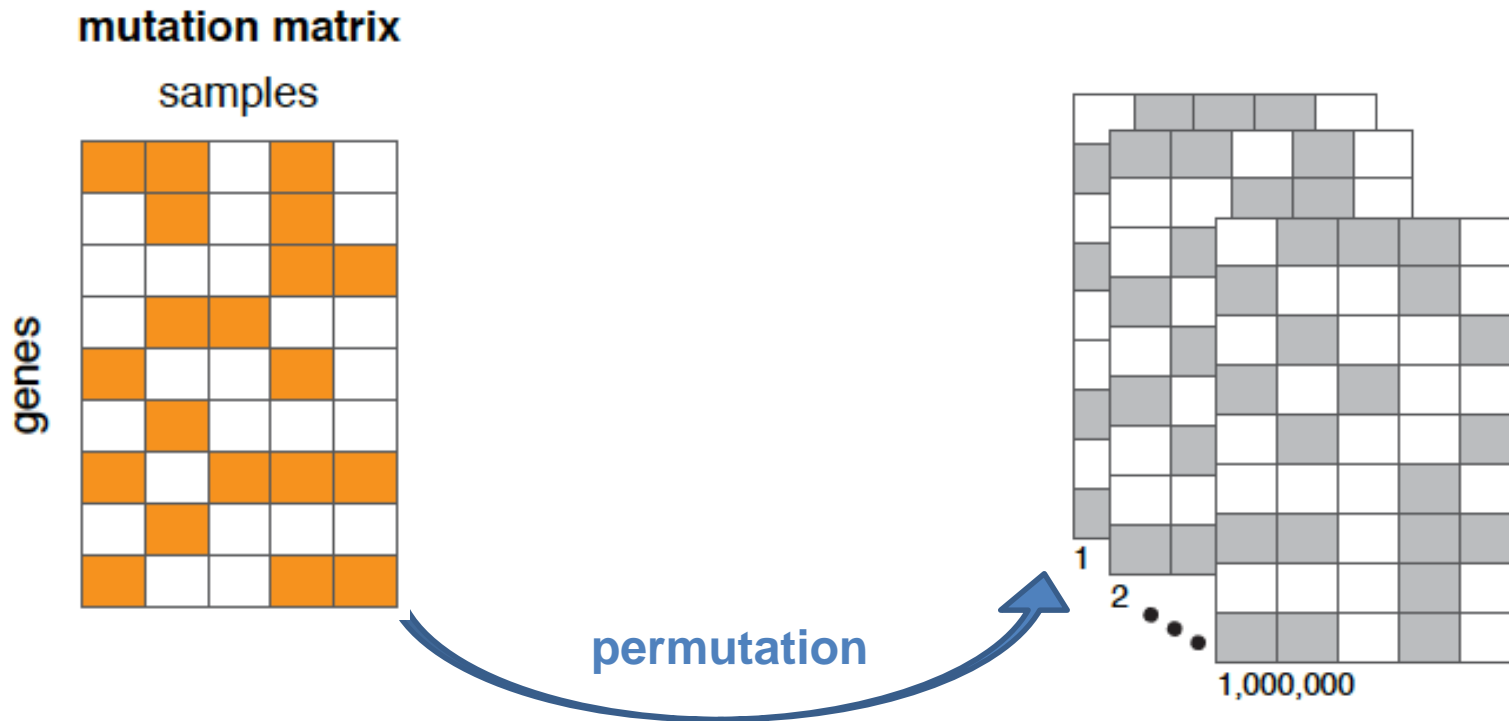
Methods to detect genetic interactions

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



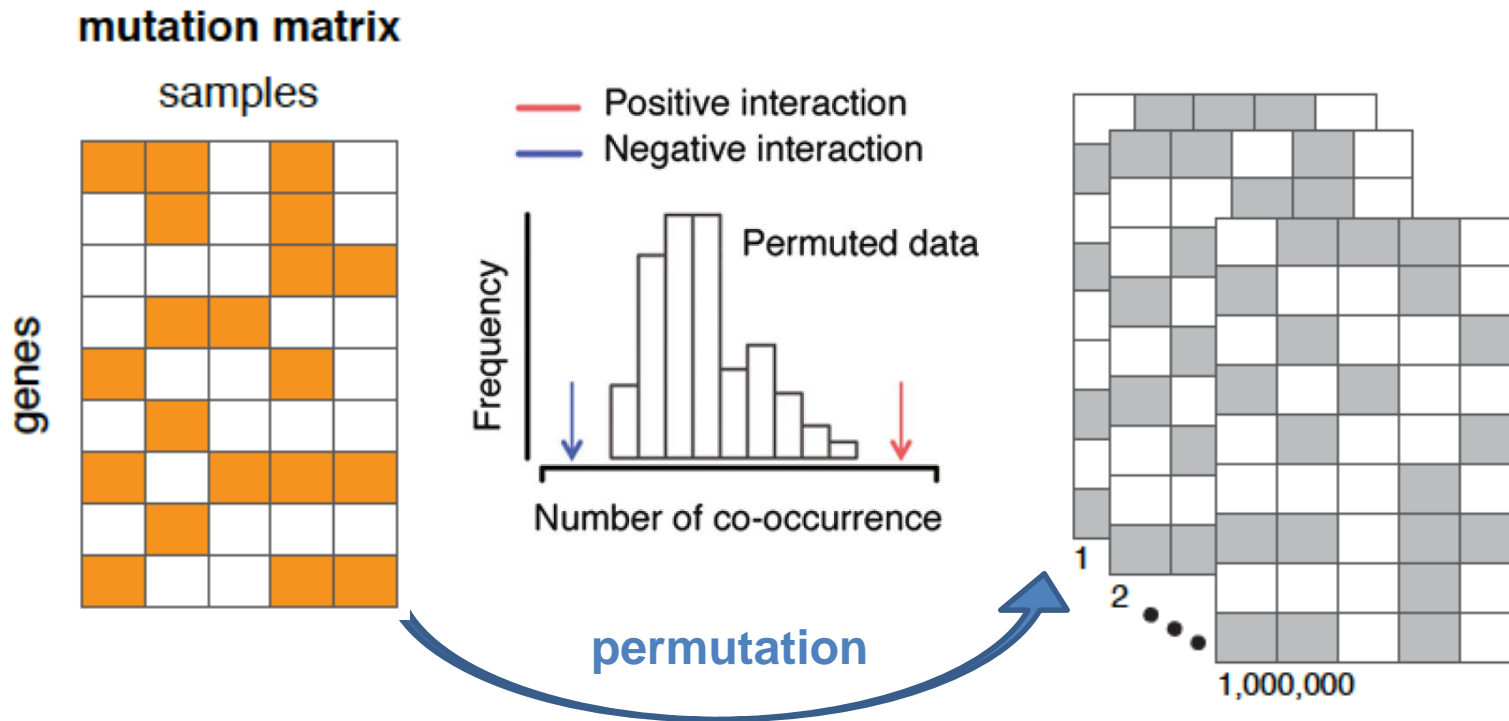
Methods to detect genetic interactions

- 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



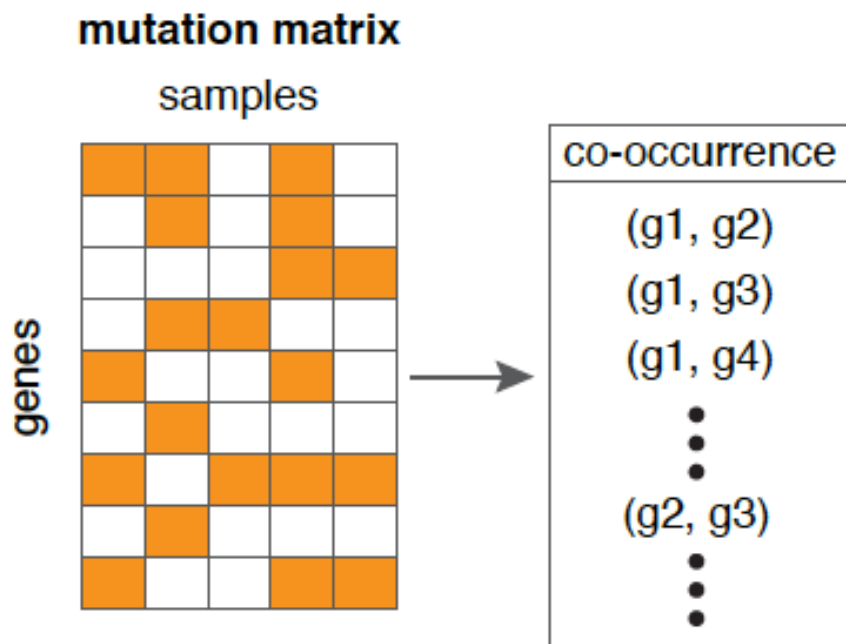
Methods to detect genetic interactions

- 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



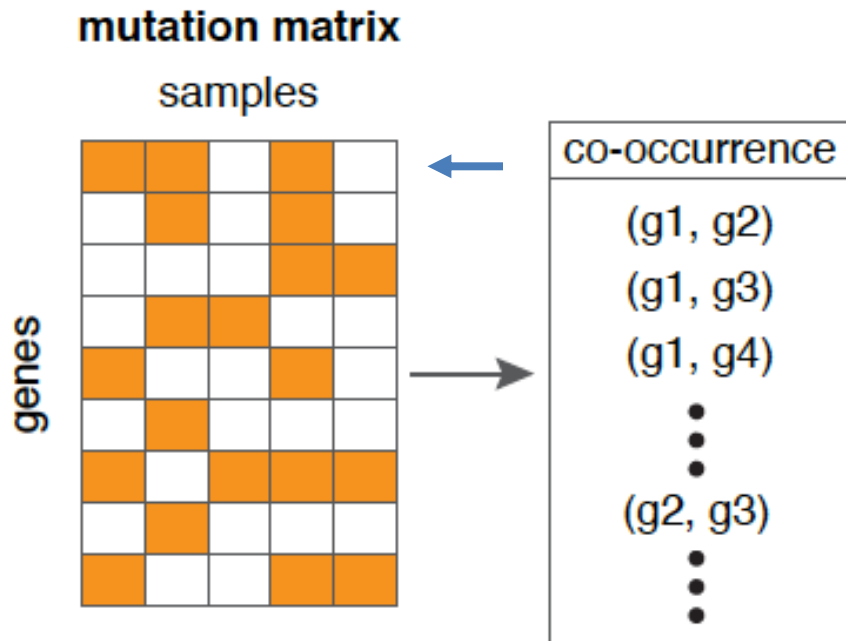
- Compare observed counts with null distribution to **infer significance** (p-value)

Implementation in R



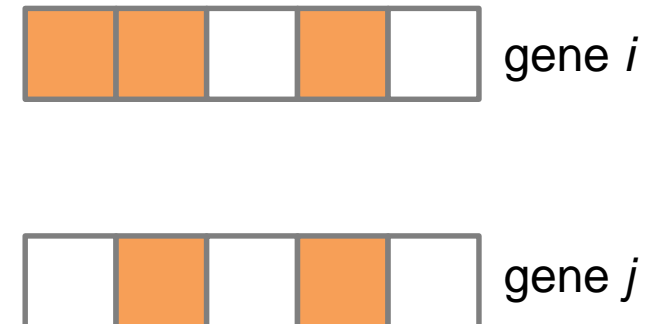
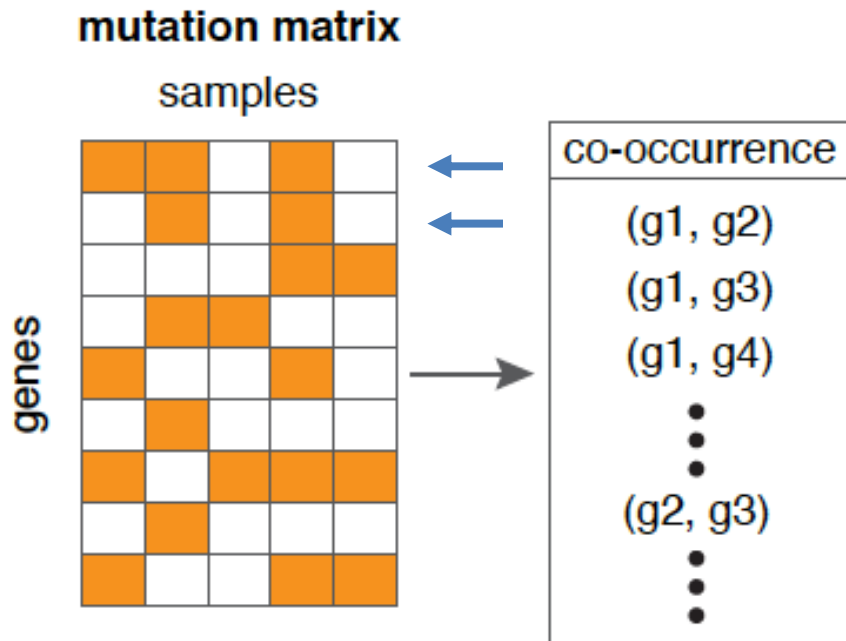
Implementation in R

for each gene i



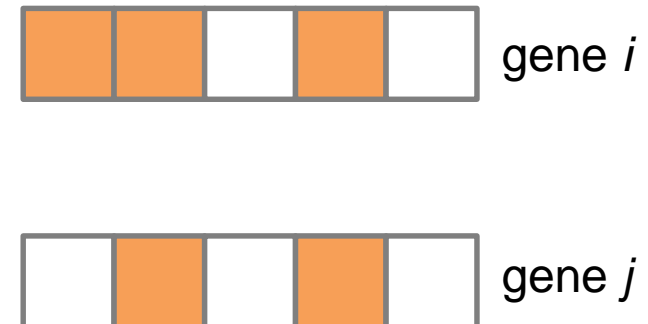
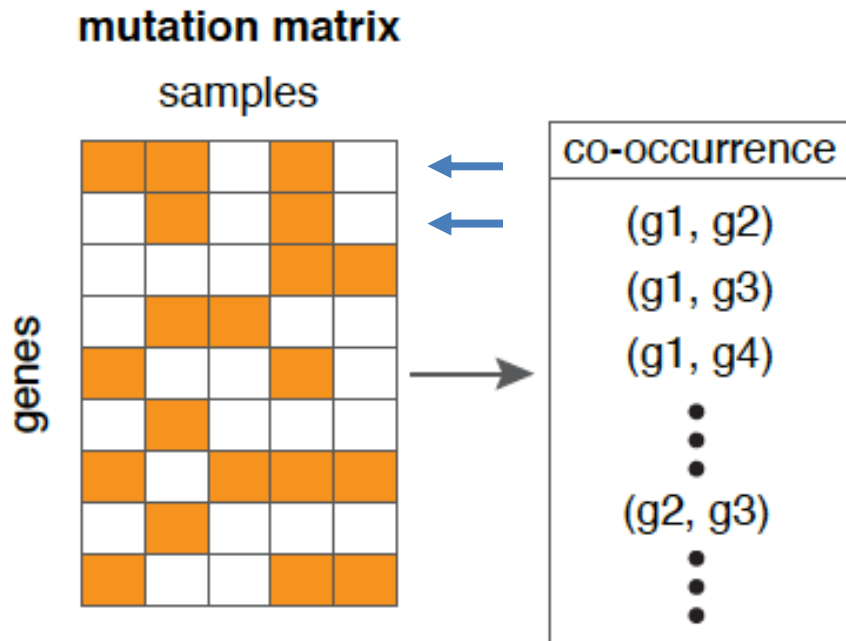
Implementation in R

```
for each gene  $i$   
  for each gene  $j \neq i$ 
```



Implementation in R

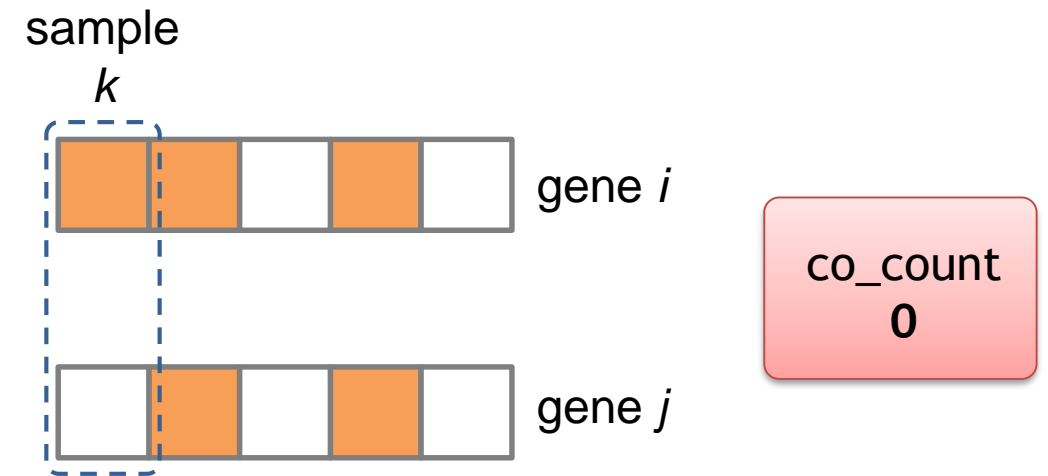
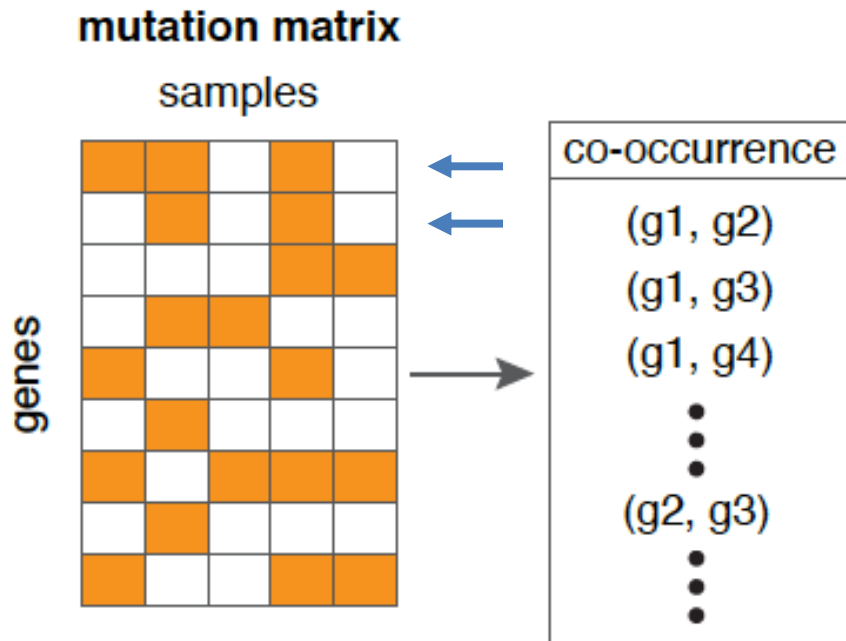
```
for each gene  $i$   
  for each gene  $j \neq i$   
     $\text{co\_count}_{i,j} = 0$ 
```



co_count
0

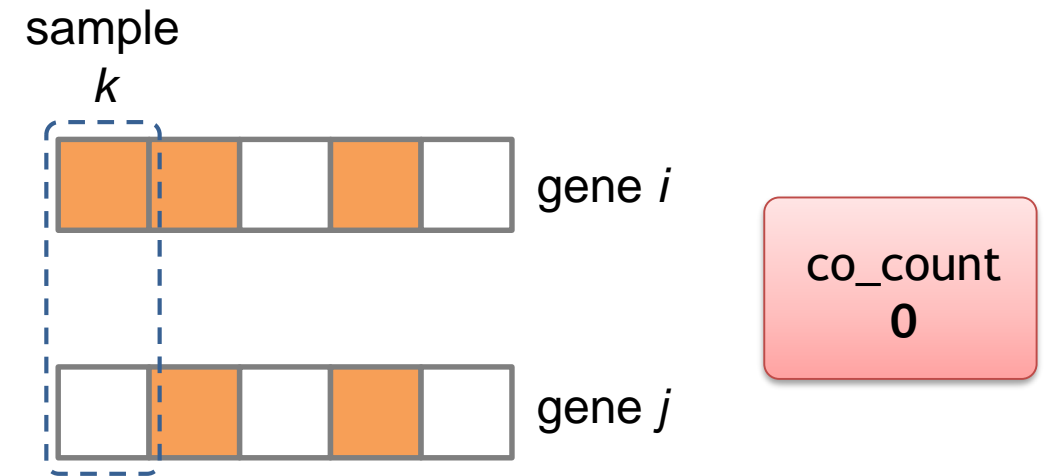
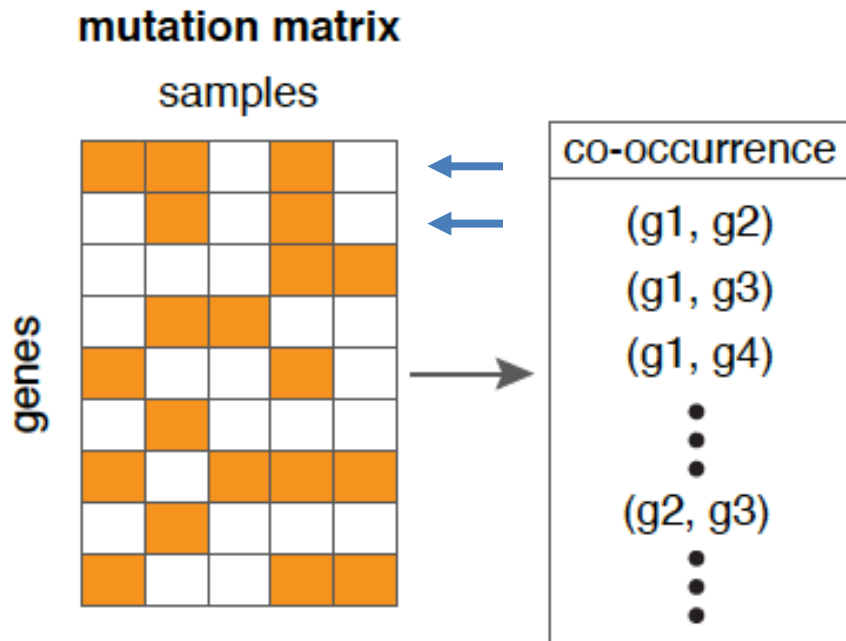
Implementation in R

```
for each gene  $i$   
  for each gene  $j \neq i$   
     $\text{co\_count}_{i,j} = 0$   
    for each sample  $k$ 
```



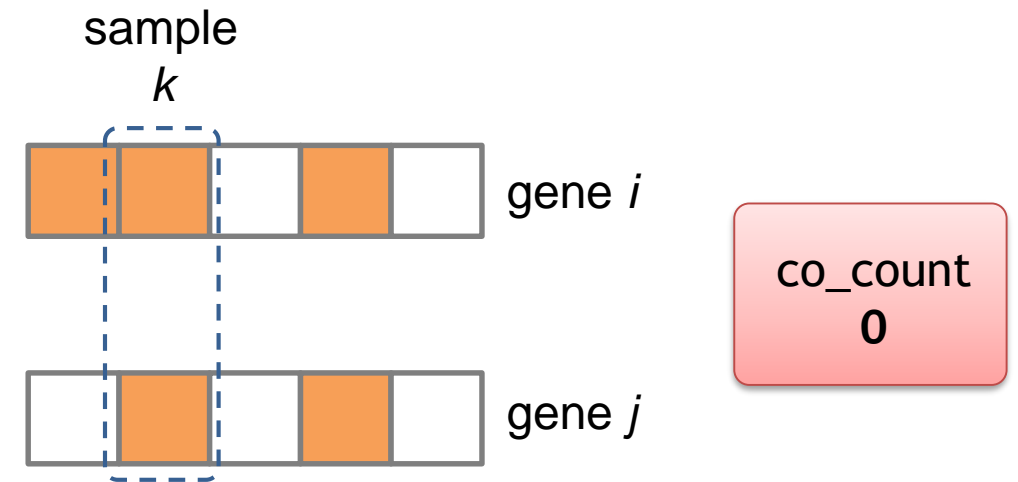
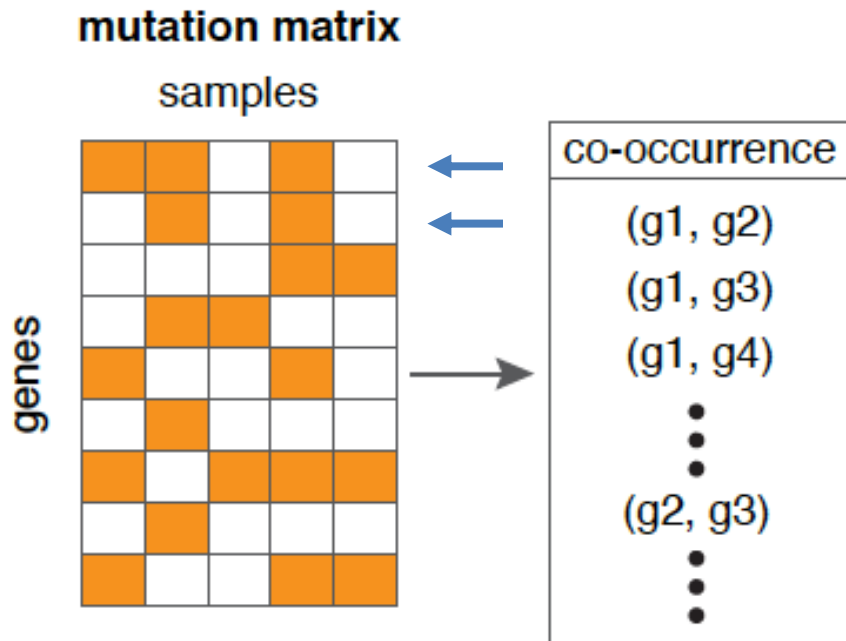
Implementation in R

```
for each gene  $i$ 
  for each gene  $j \neq i$ 
     $\text{co\_count}_{i,j} = 0$ 
    for each sample  $k$ 
      if  $\text{mtx}_{i,k} == 1 \ \& \ \text{mtx}_{j,k} == 1$  then
```



Implementation in R

```
for each gene  $i$ 
  for each gene  $j \neq i$ 
     $\text{co\_count}_{i,j} = 0$ 
    for each sample  $k$ 
      if  $\text{mtx}_{i,k} == 1 \ \& \ \text{mtx}_{j,k} == 1$  then
```



Implementation in R

```
for each gene  $i$ 
```

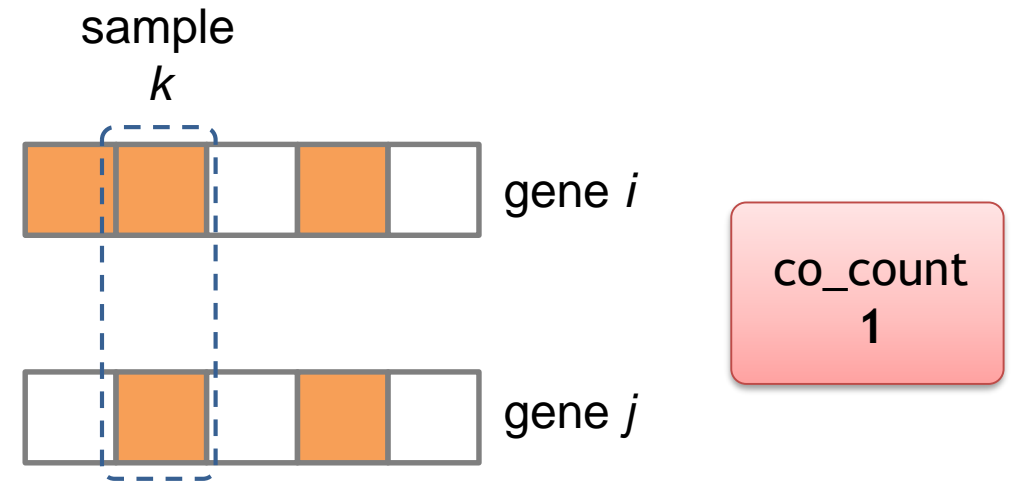
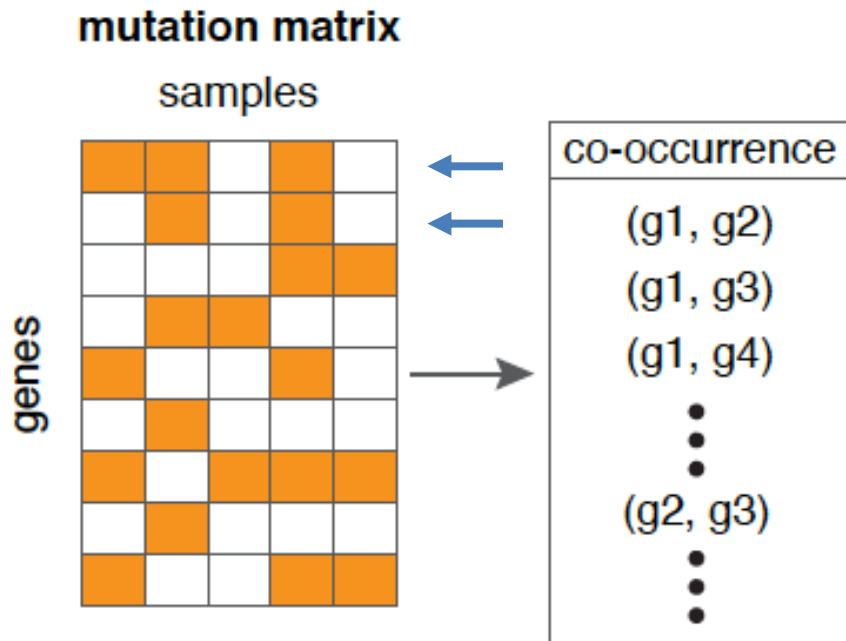
```
  for each gene  $j \neq i$ 
```

```
     $\text{co\_count}_{i,j} = 0$ 
```

```
    for each sample  $k$ 
```

```
      if  $\text{mtx}_{i,k} == 1 \ \& \ \text{mtx}_{j,k} == 1$  then
```

```
         $\text{co\_count}_{i,j} = \text{co\_count}_{i,j} + 1$ 
```



Implementation in R

```
for each gene  $i$ 
```

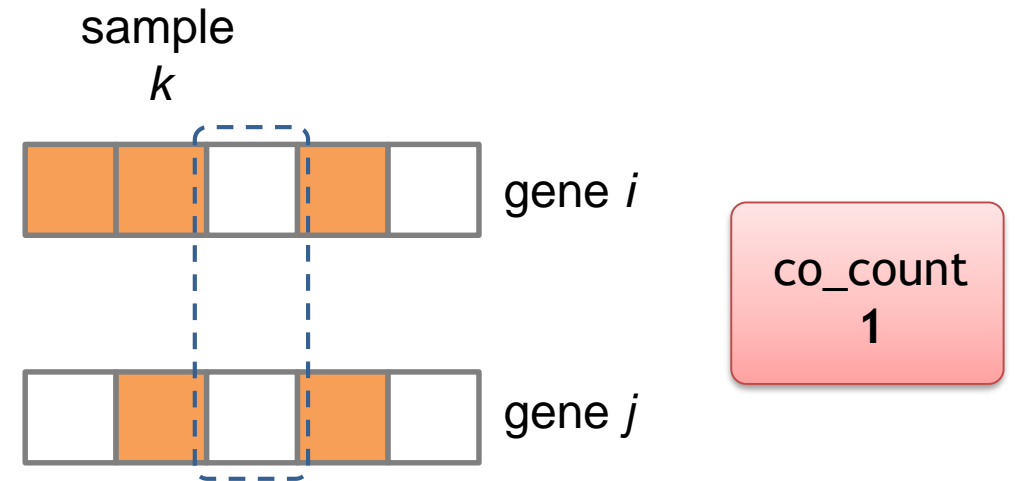
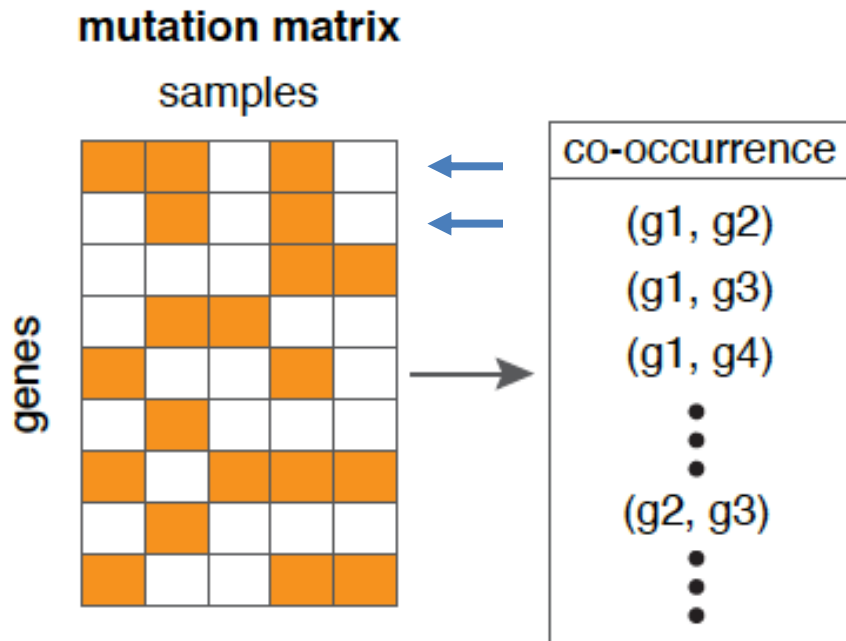
```
  for each gene  $j \neq i$ 
```

```
     $\text{co\_count}_{i,j} = 0$ 
```

```
    for each sample  $k$ 
```

```
      if  $\text{mtx}_{i,k} == 1 \ \& \ \text{mtx}_{j,k} == 1$  then
```

```
         $\text{co\_count}_{i,j} = \text{co\_count}_{i,j} + 1$ 
```



Implementation in R

```
for each gene  $i$ 
```

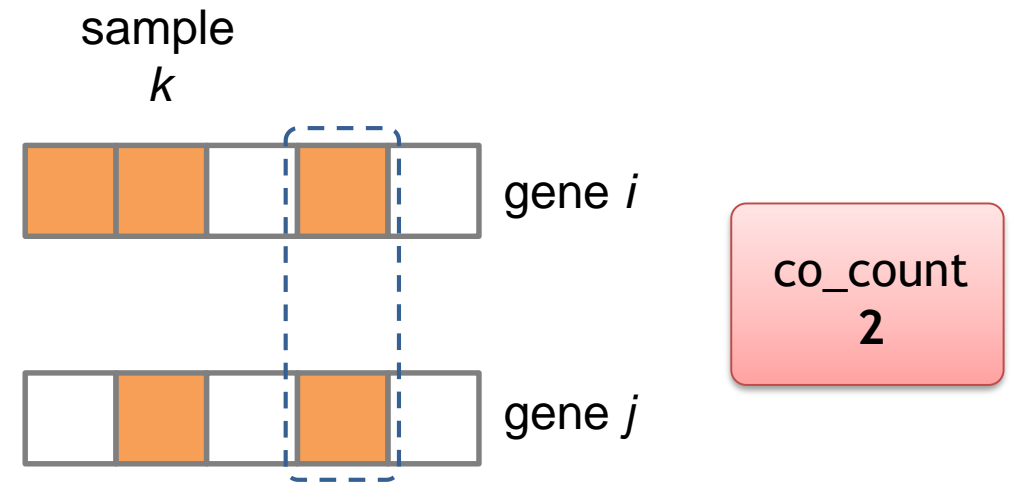
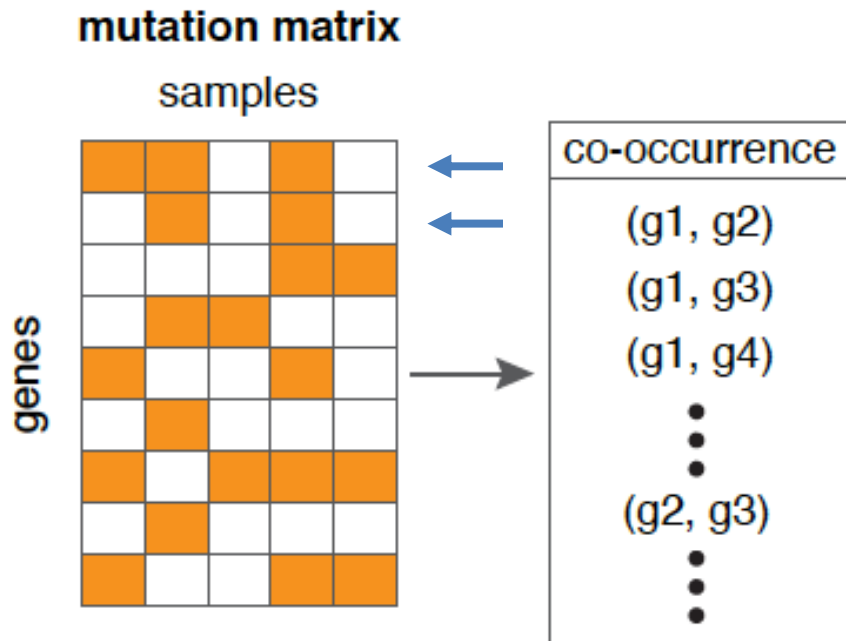
```
  for each gene  $j \neq i$ 
```

```
     $\text{co\_count}_{i,j} = 0$ 
```

```
    for each sample  $k$ 
```

```
      if  $\text{mtx}_{i,k} == 1 \ \& \ \text{mtx}_{j,k} == 1$  then
```

```
         $\text{co\_count}_{i,j} = \text{co\_count}_{i,j} + 1$ 
```



Implementation in R

```
for each gene  $i$ 
```

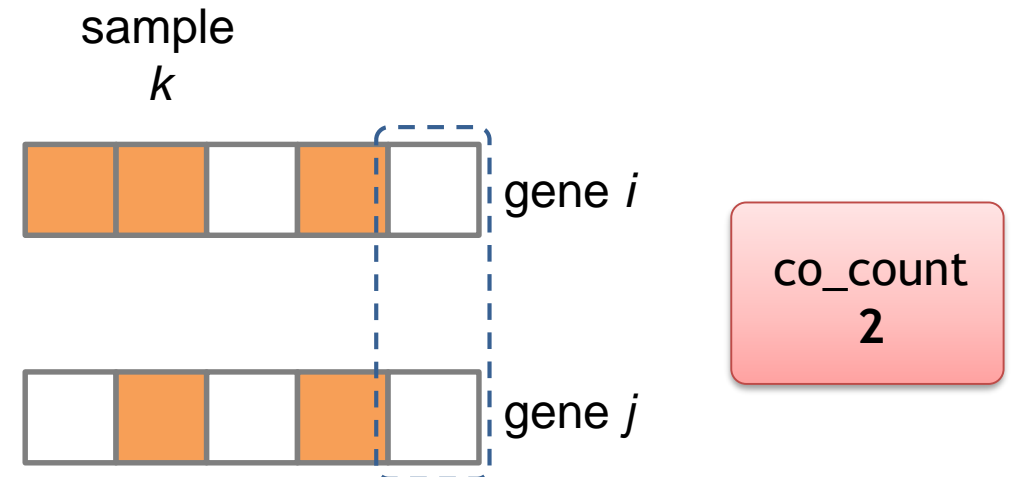
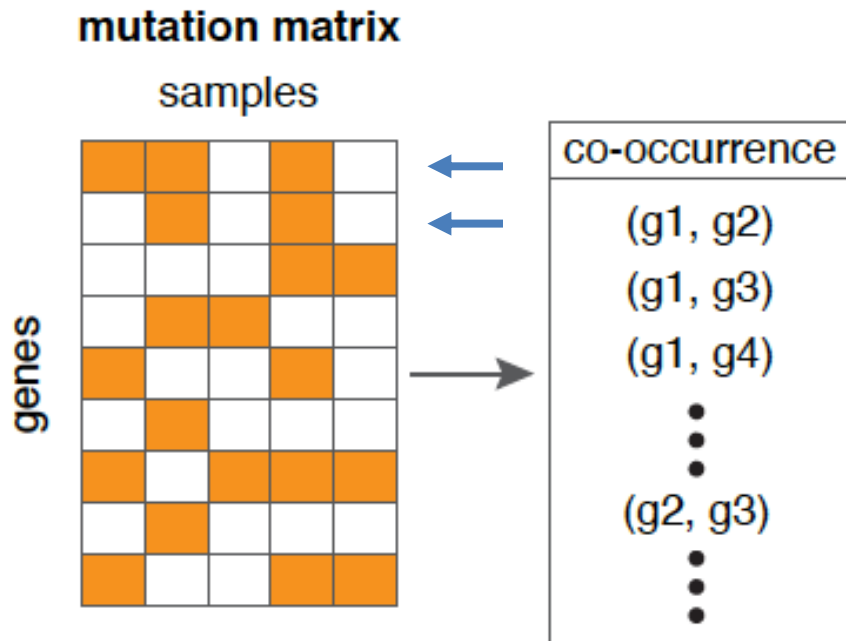
```
  for each gene  $j \neq i$ 
```

```
     $\text{co\_count}_{i,j} = 0$ 
```

```
    for each sample  $k$ 
```

```
      if  $\text{mtx}_{i,k} == 1 \ \& \ \text{mtx}_{j,k} == 1$  then
```

```
         $\text{co\_count}_{i,j} = \text{co\_count}_{i,j} + 1$ 
```



Implementation in R

```
for each gene  $i$ 
```

```
  for each gene  $j \neq i$ 
```

```
     $\text{co\_count}_{i,j} = 0$ 
```

```
    for each sample  $k$ 
```

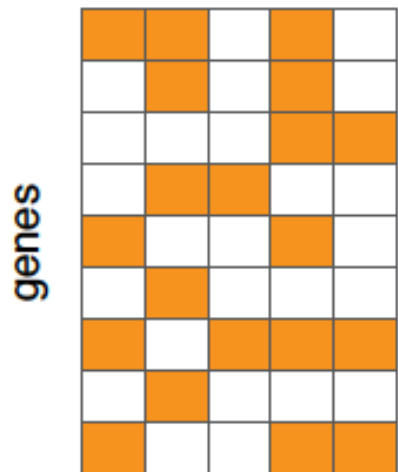
```
      if  $\text{mtx}_{i,k} == 1 \ \& \ \text{mtx}_{j,k} == 1$  then
```

```
         $\text{co\_count}_{i,j} = \text{co\_count}_{i,j} + 1$ 
```



mutation matrix

samples



co-occurrence

(g1, g2)

(g1, g3)

(g1, g4)

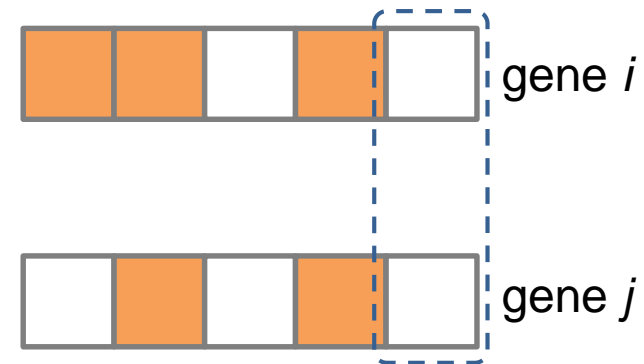
⋮

(g2, g3)

⋮

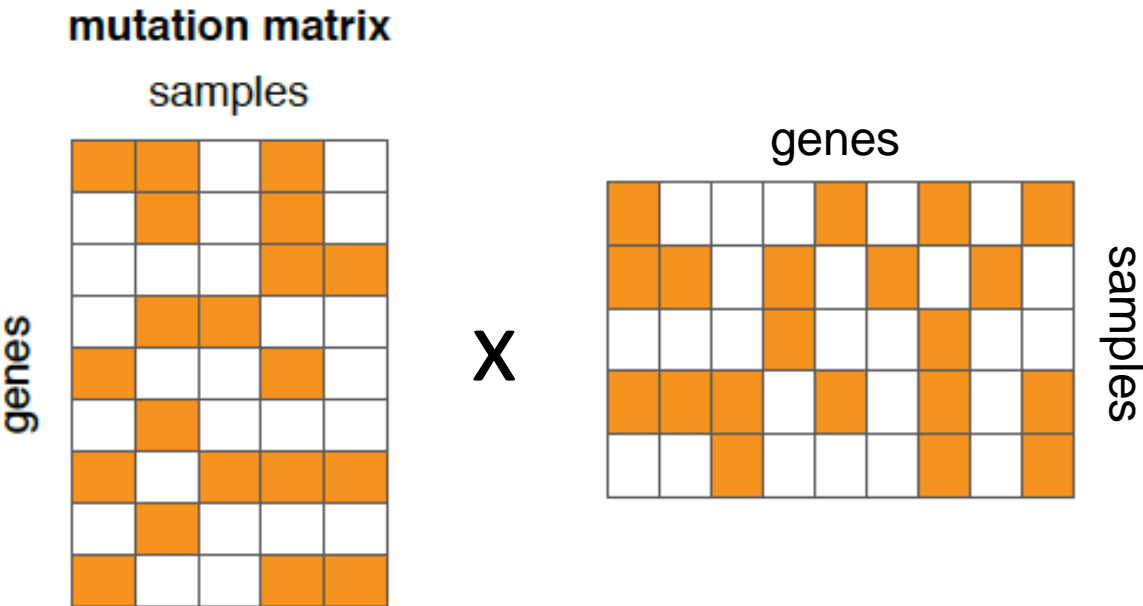
⋮

sample
 k

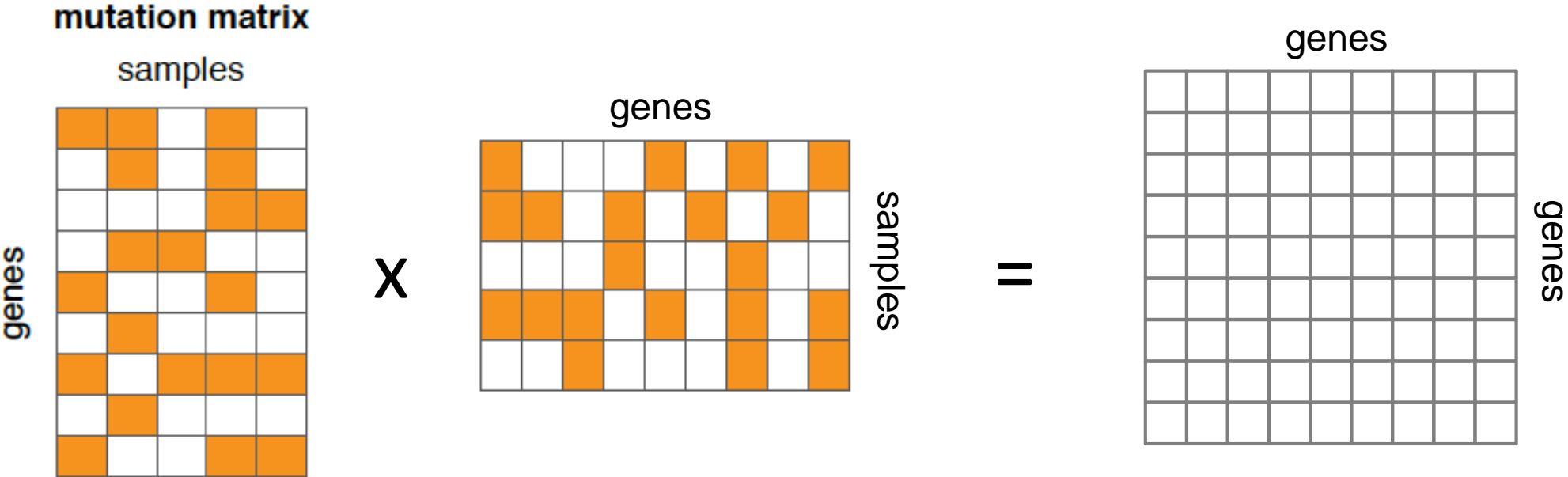


co_count
2

Matrix multiplication



Matrix multiplication



Matrix multiplication



mutation matrix

samples

genes

X

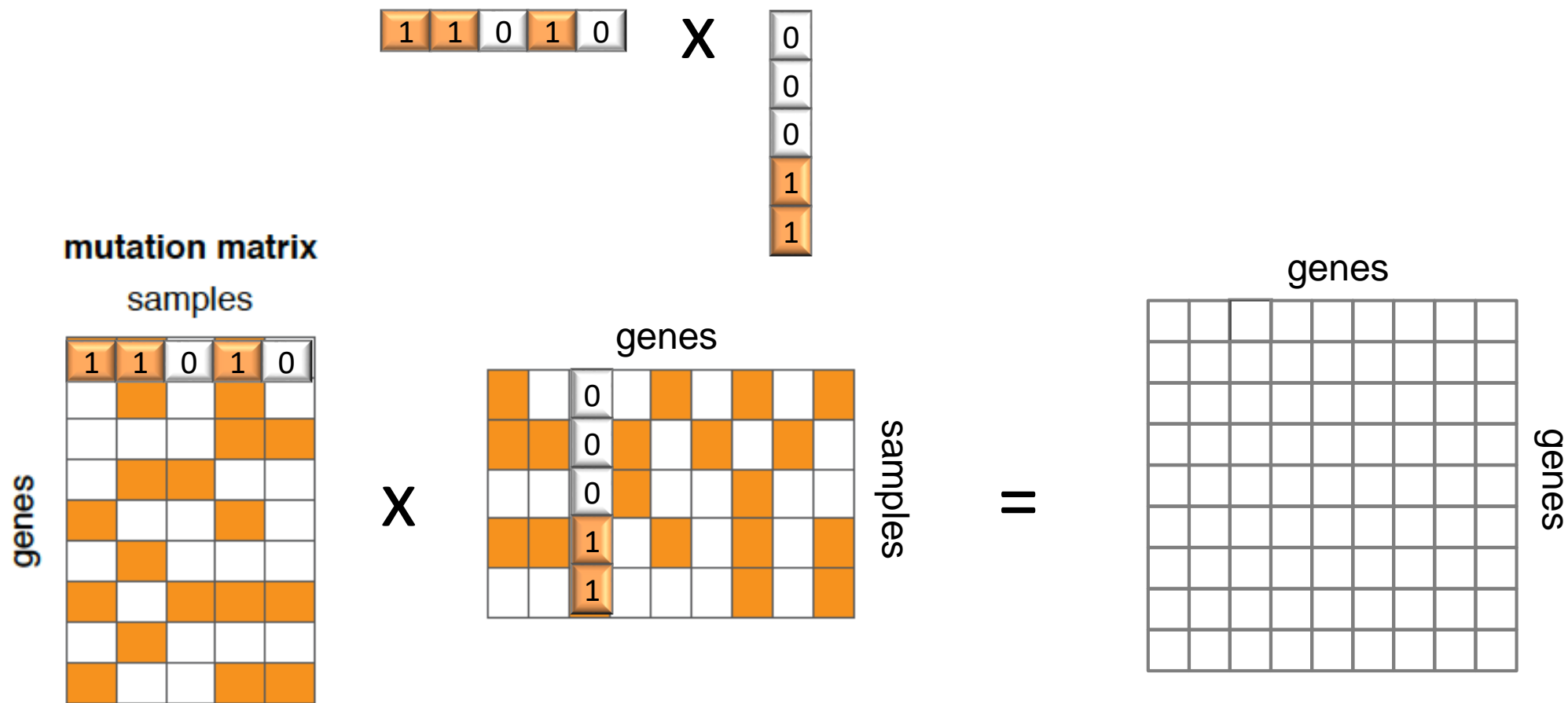
genes

samples



genes

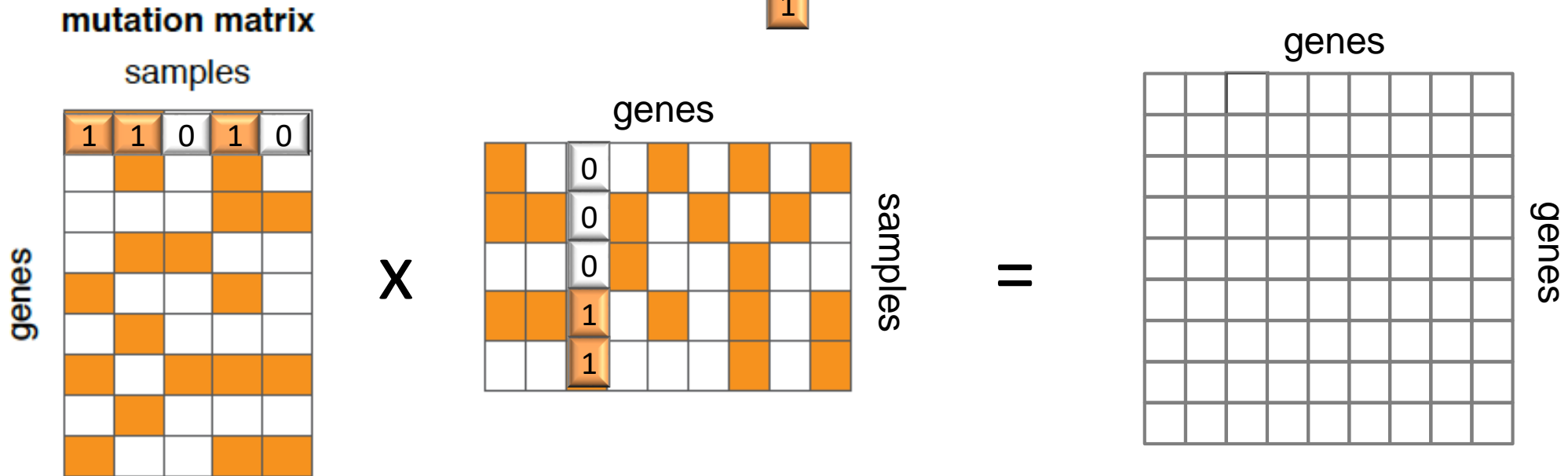
genes

Matrix multiplication



Matrix multiplication


 \times

 $=$
 $1 \times 0 + 1 \times 0 + 0 \times 0 + 1 \times 1 + 1 \times 0$
 $=$
 1

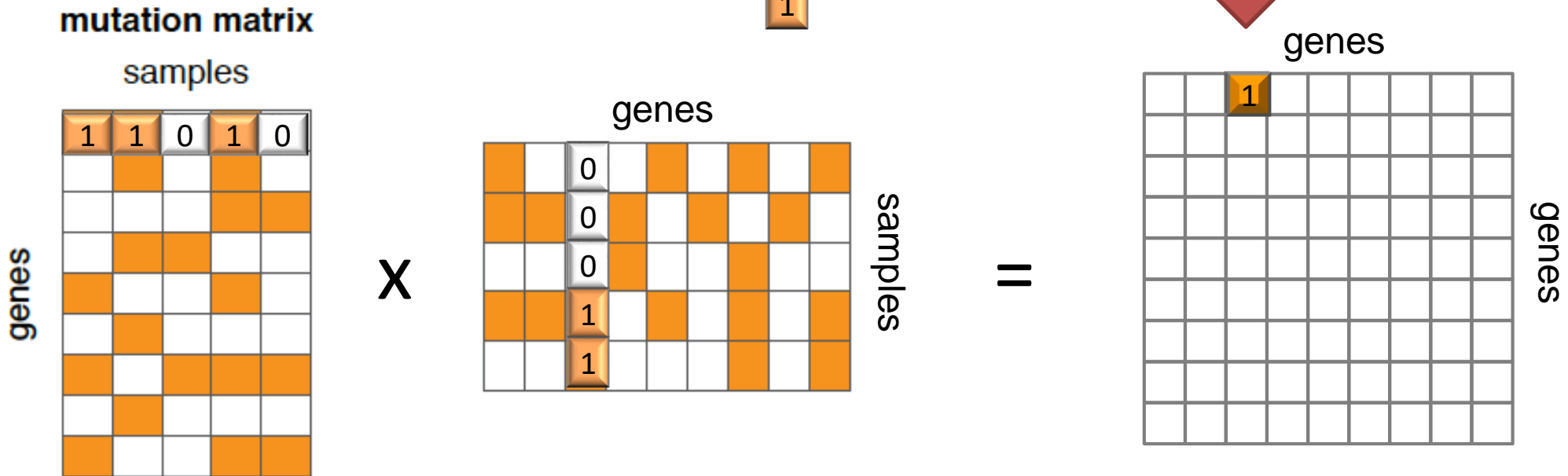


Matrix multiplication

Diagram illustrating the dot product of two 5-bit vectors:

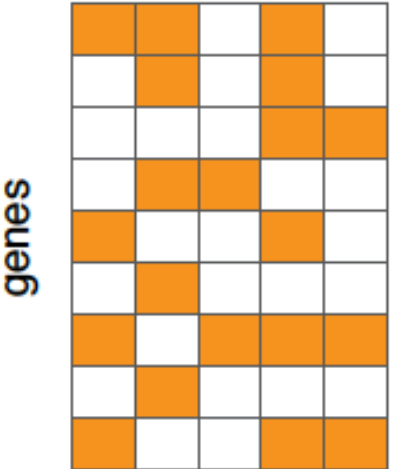
$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = 1 \times 0 + 1 \times 0 + 0 \times 0 + 1 \times 1 + 1 \times 0 = 1$$

A red arrow points from the result '1' to the next slide.



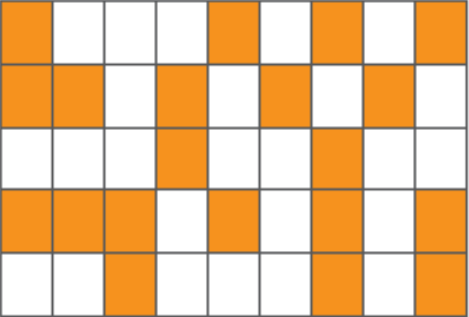
Matrix multiplication

mutation matrix
samples



X

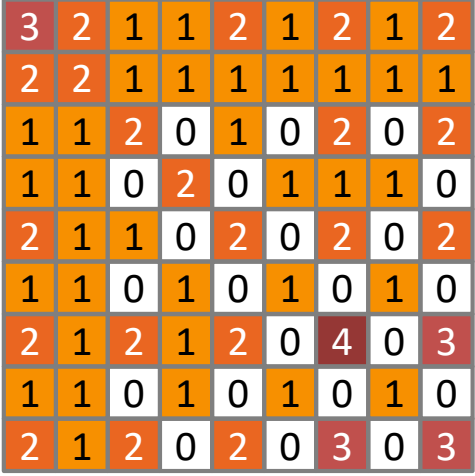
genes



samples

=

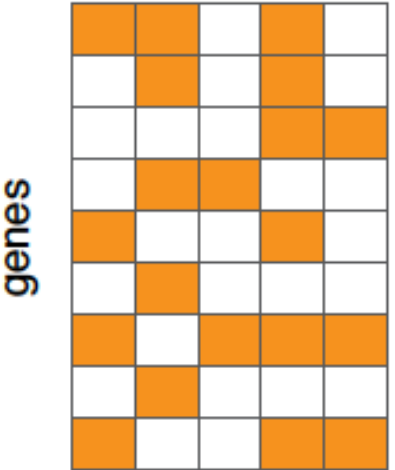
genes



genes

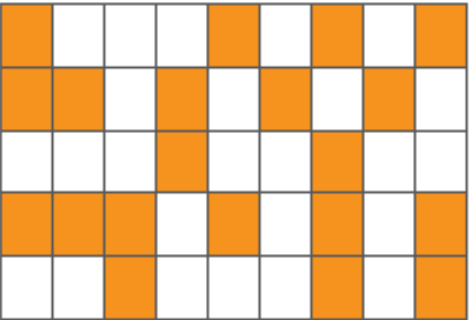
Matrix multiplication

mutation matrix
samples



X

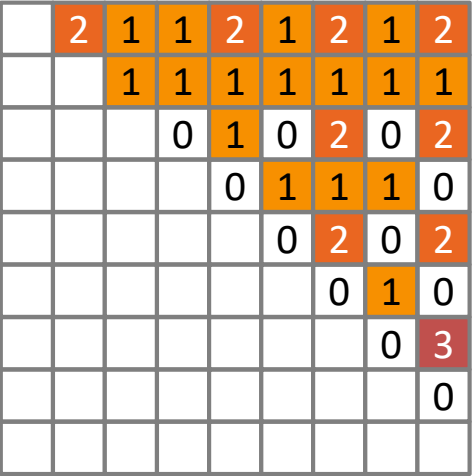
genes



samples

=

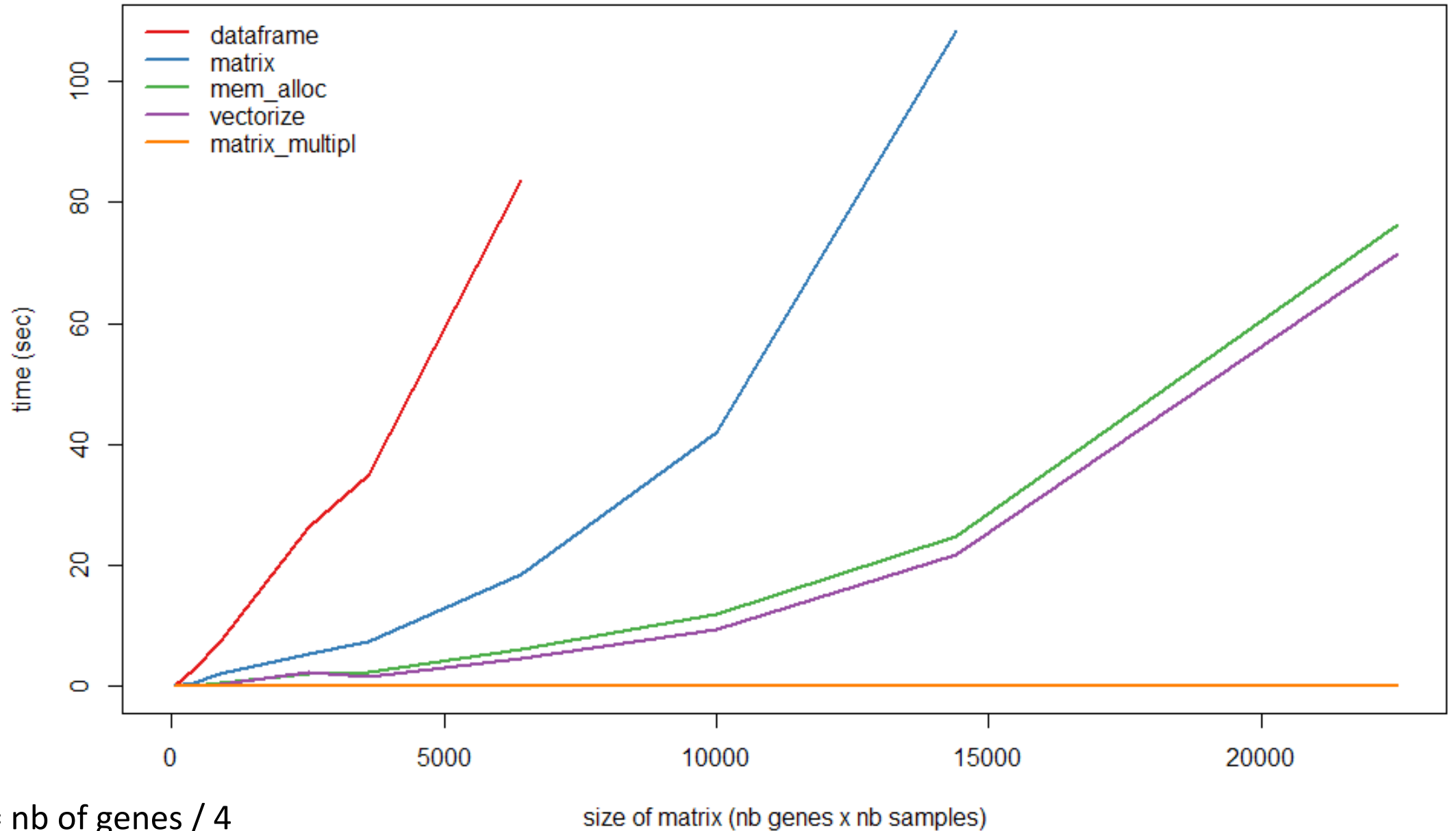
genes



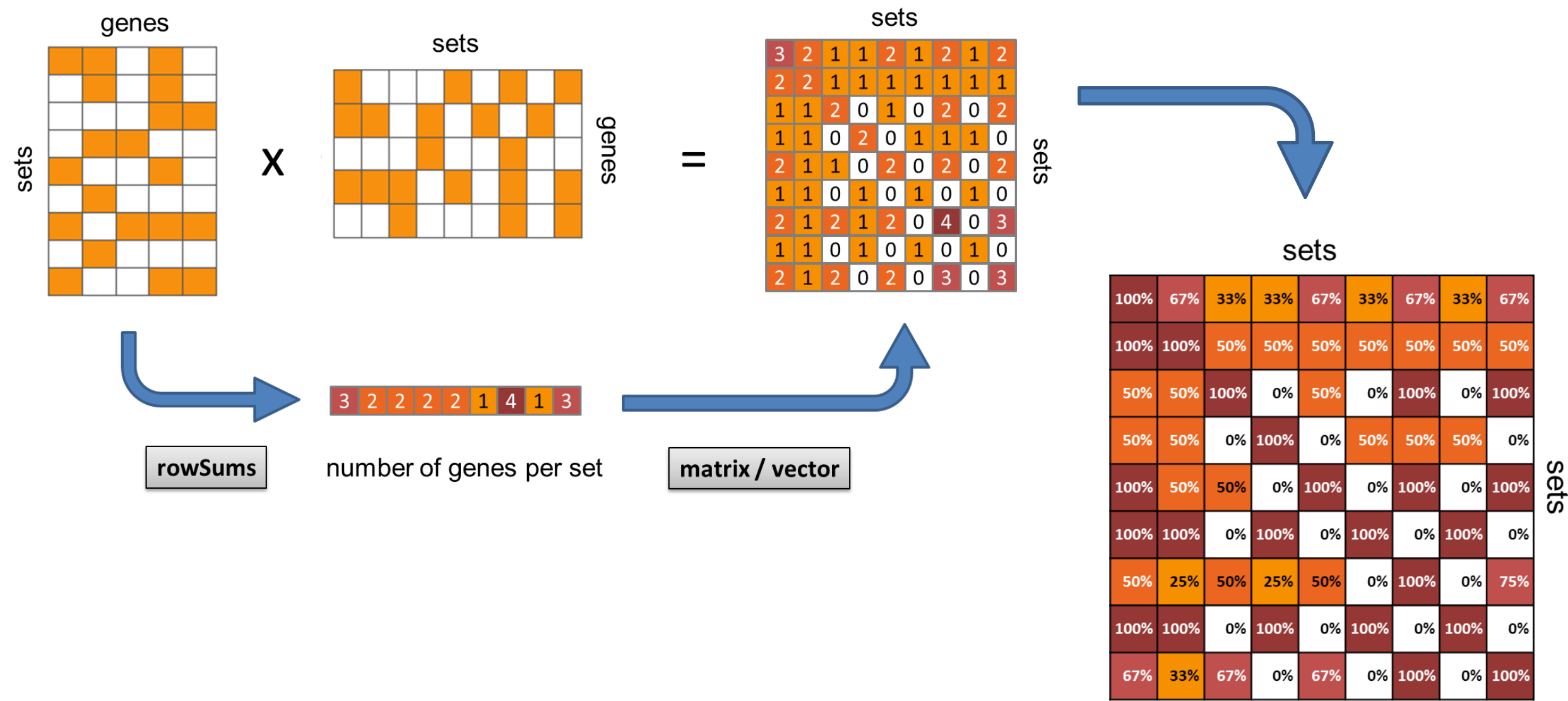
genes

co-occurrence
(g1, g2)
(g1, g3)
(g1, g4)
⋮
(g2, g3)
⋮

Benchmarking



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

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  $\text{sum}(\text{col1}_i, \text{col2}_i, \text{col3}_i, \text{col4}_i) > 4$ 
         $\text{output}_i = \text{"greater"}$ 
    else
         $\text{output}_i = \text{"less_or_equal"}$ 
```

Example 2: which rows have sum > 4?

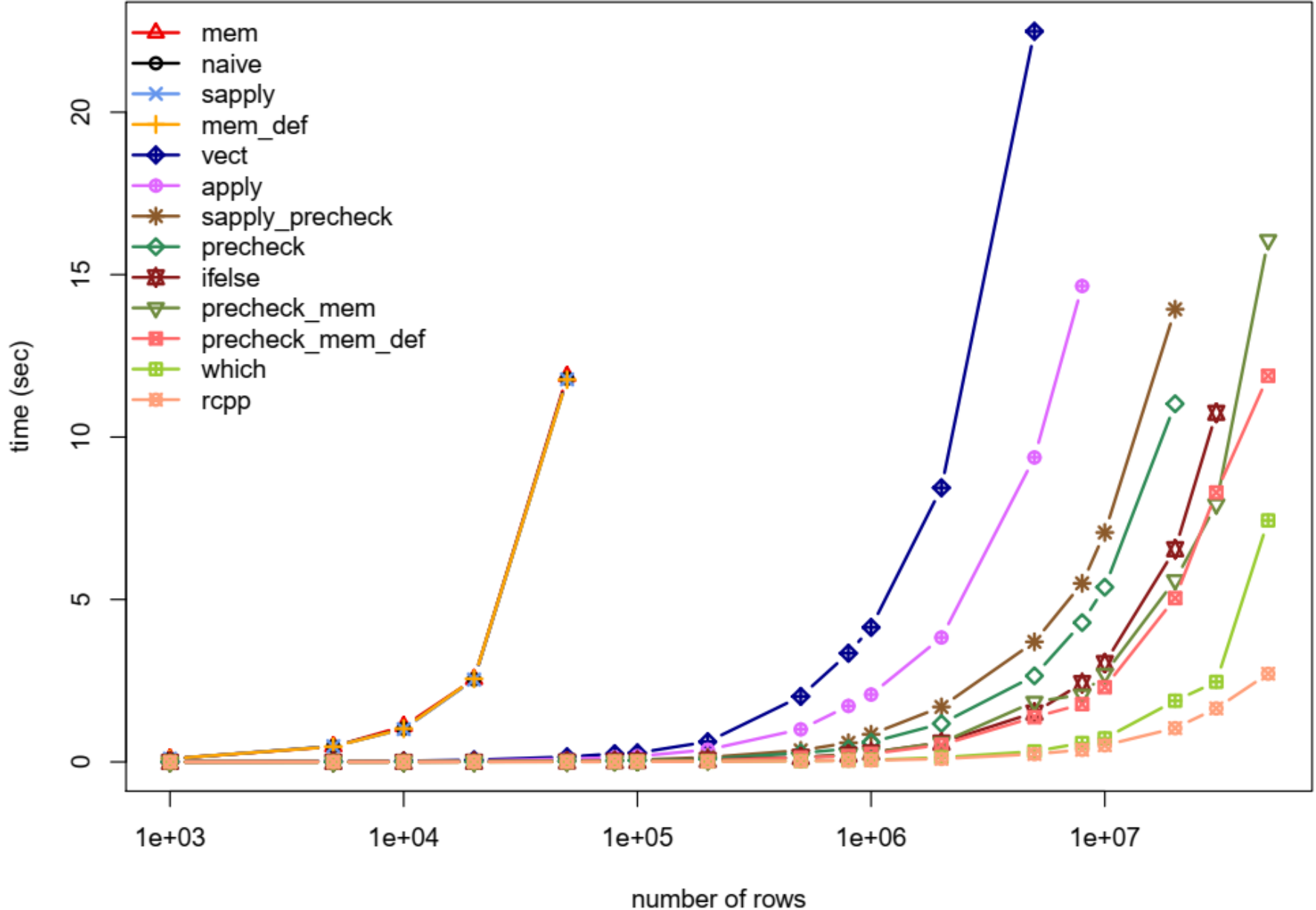
col1	col2	col3	col4
1.598	-0.325	1	9.647
1.110	2.491	5	1.712
1.298	-0.352	4	1.566
1.824	1.116	2	1.207
0.943	1.148	4	2.776
1.214	0.190	4	4.054
1.259	-2.696	2	0.046
0.214	0.050	1	0.621

```
for each row  $i$   
  if  $\text{sum}(\text{col1}_i, \text{col2}_i, \text{col3}_i, \text{col4}_i) > 4$   
     $\text{output}_i = \text{"greater"}$   
  else  
     $\text{output}_i = \text{"less_or_equal"}$ 
```

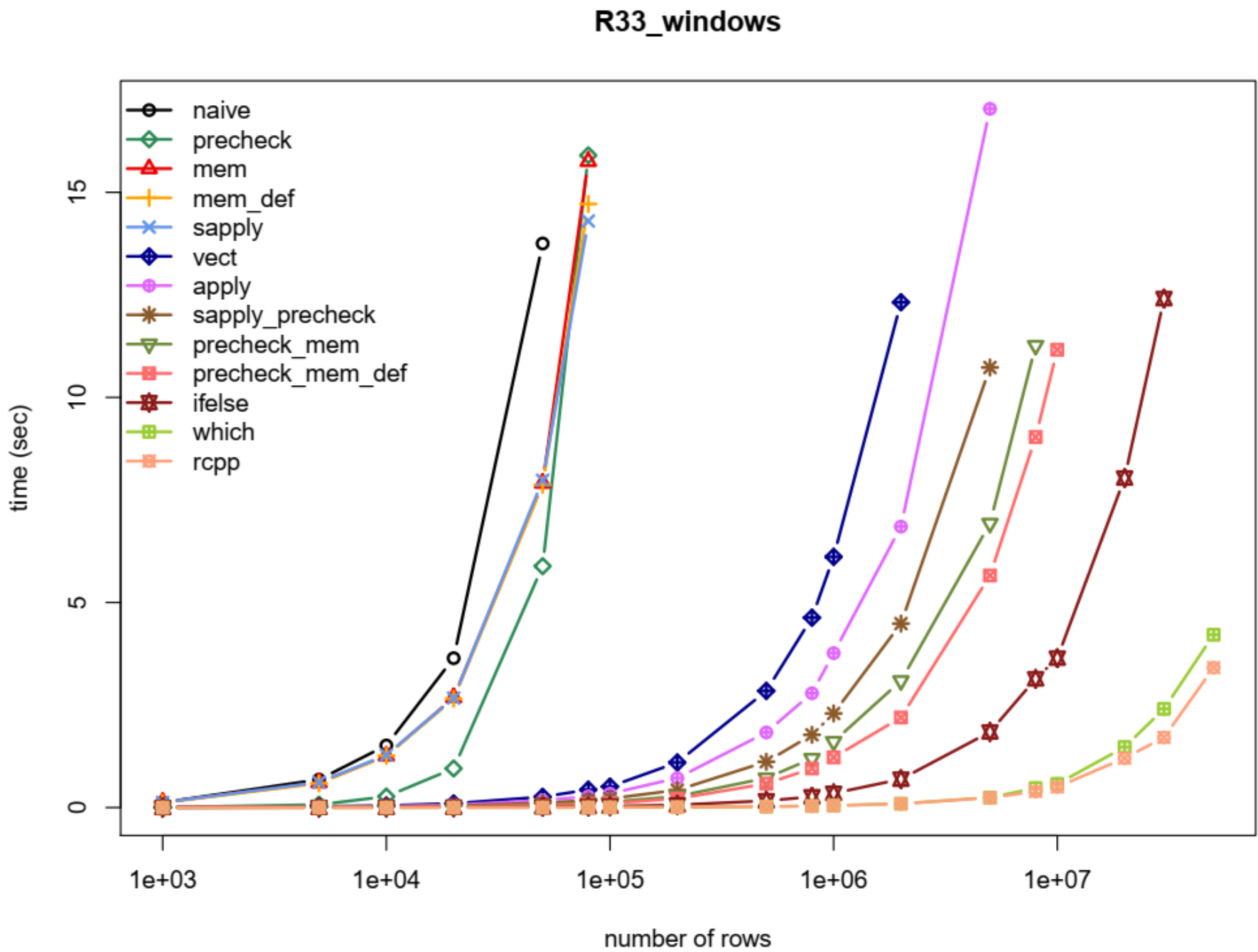


Benchmarking

R34_mac

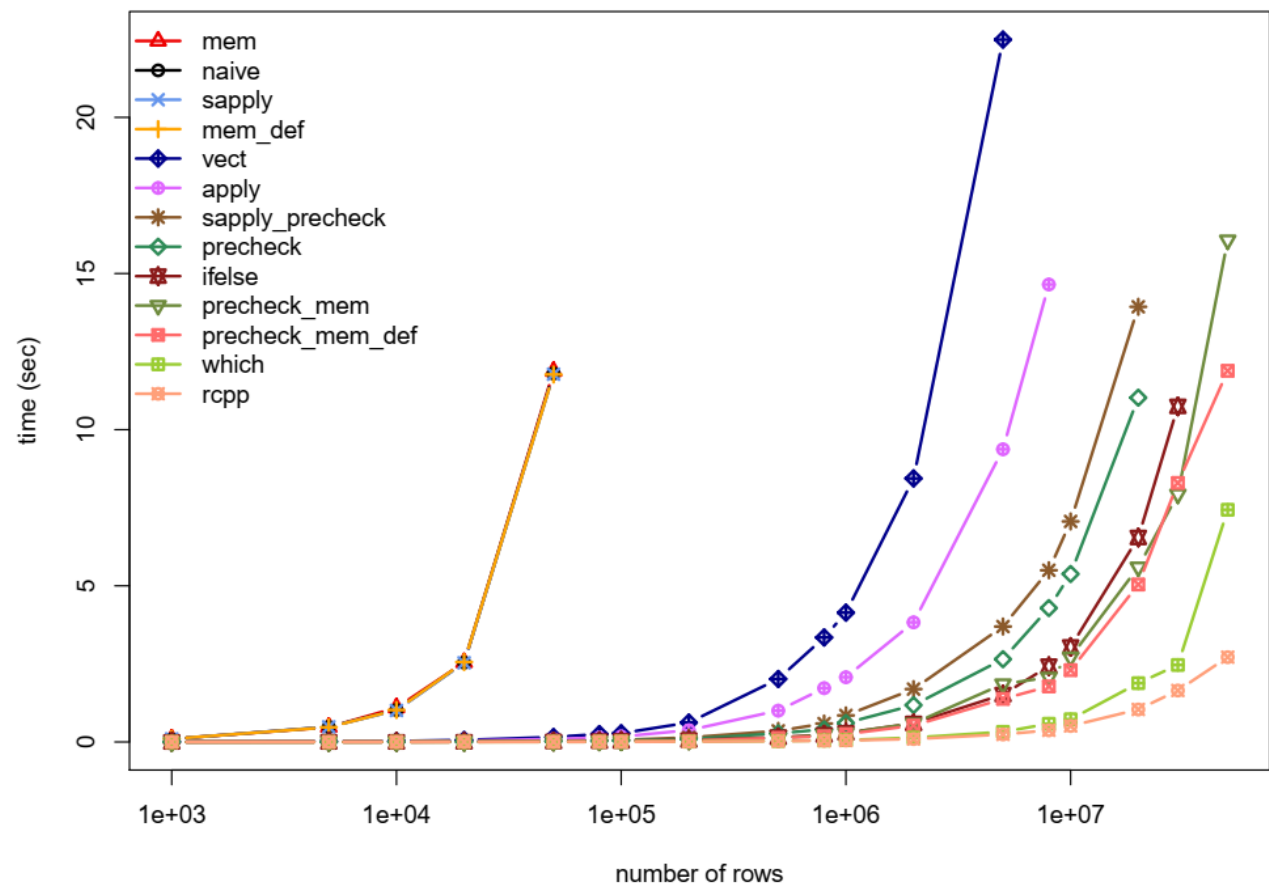


Benchmarking

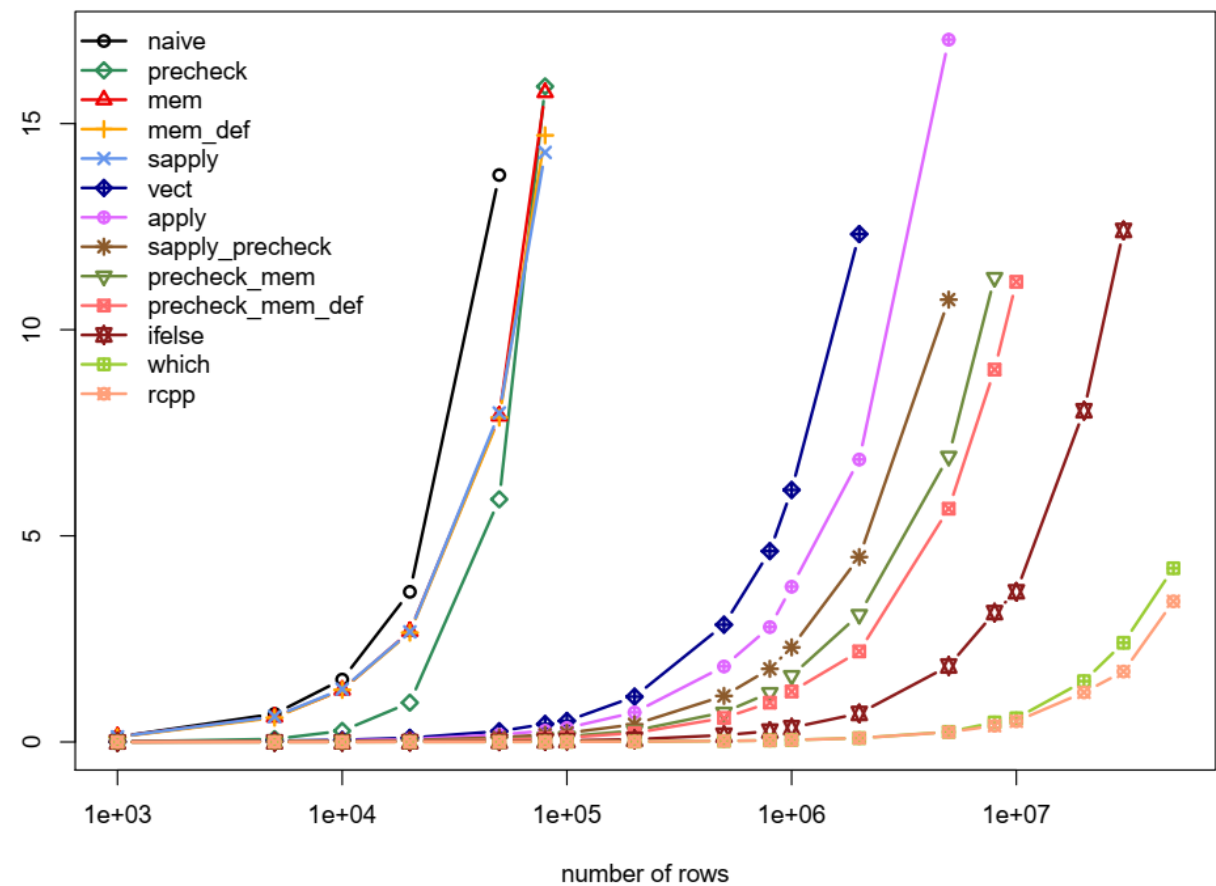


Benchmarking

R34_mac



R33_windows



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

