

# Quick Permutation Test: feature filtering of n-gram data

Piotr Sobczyk<sup>1\*</sup>, Michał Burdukiewicz<sup>2</sup>, Chris Lauber<sup>3</sup>, Paweł Mackiewicz<sup>2</sup>  
\*Piotr.Sobczyk@pwr.edu.pl

<sup>1</sup>Wrocław University of Technology, Department of Mathematics, Poland

<sup>2</sup>University of Wrocław, Department of Genomics, Poland

<sup>3</sup>Dresden University of Technology, Institute of Medical Informatics and Biometry, Poland

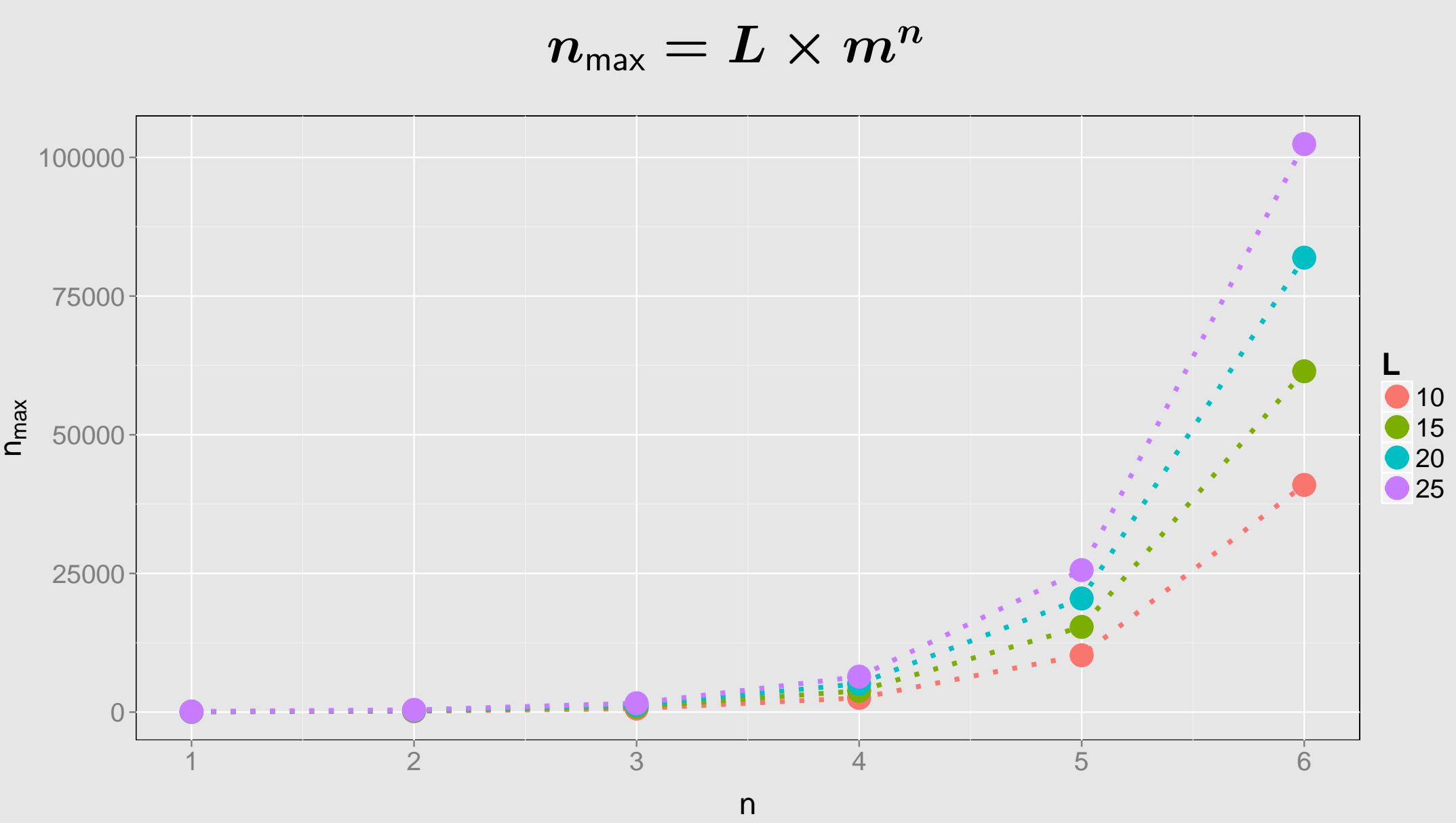
## Introduction

N-grams (k-tuples) are vectors of n characters derived from input sequence(s). They may form continuous sub-sequences or be discontinuous. Another important n-gram parameter is its position. Instead of just counting n-grams, one may want to count how many n-grams occur at a given position in multiple (e.g. related) sequences. Originally developed for natural language processing, n-grams are also used in genomics (Fang et al., 2011), transcriptomics (Wang et al., 2014) and proteomics (Guo et al., 2014).

	P1	P2	P3	P4	P5	P6		1	2	3	4														
S1	3	1	2	3	2	2		1	3	2	0														
S2	3	3	2	2	4	2		0	3	2	1														
S3	2	4	4	4	4	1		1	1	0	4														
Sample sequences.								Unigram counts.																	
P1	1	P2	1	P3	1	P4	1	P5	1	P6	1	P1	2	P2	2	P3	2	P4	2	P5	2	P6	2	P1	3
0		1		0		0		0		0		0		0		1		0		1		1		1	
0		0		0		0		0		0		0		0		1		1		0		1		1	
0		0		0		0		1		1		0		0		0		0		0		0		0	
A fraction of possible unigrams with position information.																									

## Curse of dimensionality

Even when we limit ourselves to only continuous positioned n-grams, feature space grows rapidly with the number of elements in n-gram ( $n$ ) and length of the sequence ( $L$ ).  
Number of possible positioned n-grams:



## Feature selecting permutation tests

Model and statistic independent permutation tests can be used to filter features obtained through counting n-grams. During a permutation test class labels are randomly exchanged during computation of significance statistic. p-values are defined as:

$$\text{p-value} = \frac{N_{T_P > T_R}}{N}$$

where  $N_{T_P > T_R}$  is number of times when  $T_P$  (permuted test statistic) was more extreme than  $T_R$  (test statistic for non-permuted data). Permutation tests are computationally expensive (especially precise estimation of low p-values, because the number of permutations is inversely proportional to the interval between p-values).

## QuiPT idea

If probability that target equals 1 is  $p$  and probability that feature equals 1 is  $q$  then we can compute the probability of given observations, eg.

$$P(\text{Target}, \text{Feature}) = (1, 1) = p \cdot q$$

Therefore another view at permutation test is that we get a contingency table, which needs to be tested for independance.

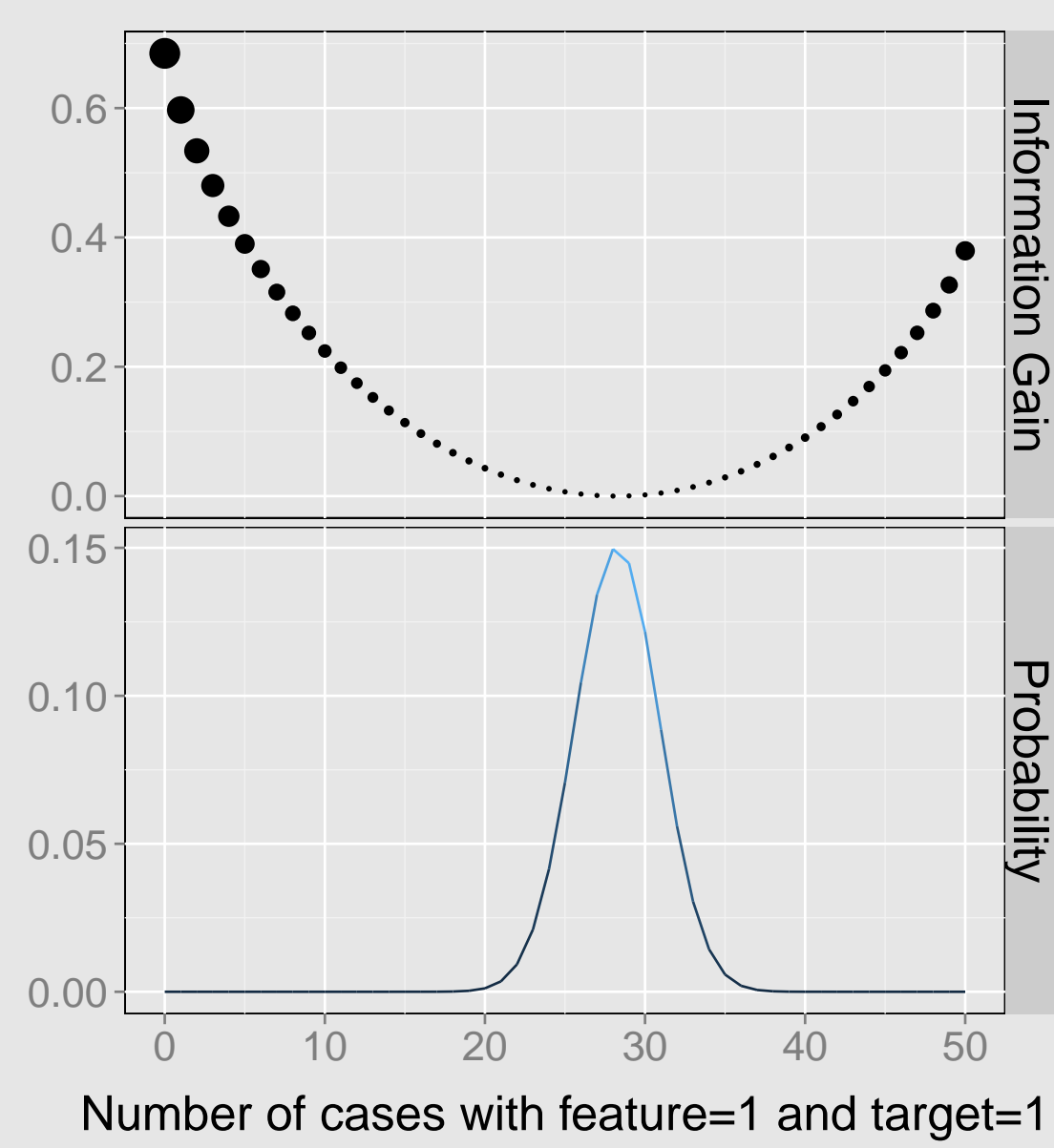
## Independence test

$F(n_{1,1}, n_{1,0}, n_{0,1}, n_{0,0}) = \binom{n}{n_{1,1}} (p \cdot q)^{n_{1,1}} \binom{n-n_{1,1}}{n_{1,0}} (p \cdot (1-q))^{n_{1,0}} \binom{n-n_{1,1}-n_{1,0}}{n_{0,1}} ((1-p) \cdot q)^{n_{0,1}} \binom{n-n_{1,1}-n_{1,0}-n_{0,1}}{n_{0,0}} ((1-p) \cdot (1-q))^{n_{0,0}}$   
This distribution comes with two constraints:  $n_{1,\cdot} = n_{1,1} + n_{1,0}$  and  $n_{\cdot,1} = n_{1,1} + n_{0,1}$ . Thus, conditioning on  $n_{1,1}$ , we get hypergeometric distribution.  
This is in fact exact two-sided Fisher's test. Information Gain is used here as a way of deciding which contingency tables are more extreme.

## Computational cost

The cost of performing QuiPT is equal to computing Information Gain and probability of occurence for  $n_{1,1} + n_{0,1}$  contingency tables. Suppose we consider 6-grams build on sequences of length 25 build of four characters. Then there are around 100,000 n-grams, features to test. This means that for Benjamini-Hochberg procedure, we need to calculate p-values with accuracy of  $0.05 \times 10^{-5}$ . This requires at least 2 million permutations. Each permutation, apart from reshuffling labels, requires computation of IG. Since n-gram features are very sparse vectors, QuiPT needs to evaluate only few contingency tables.

## Contingency table representation

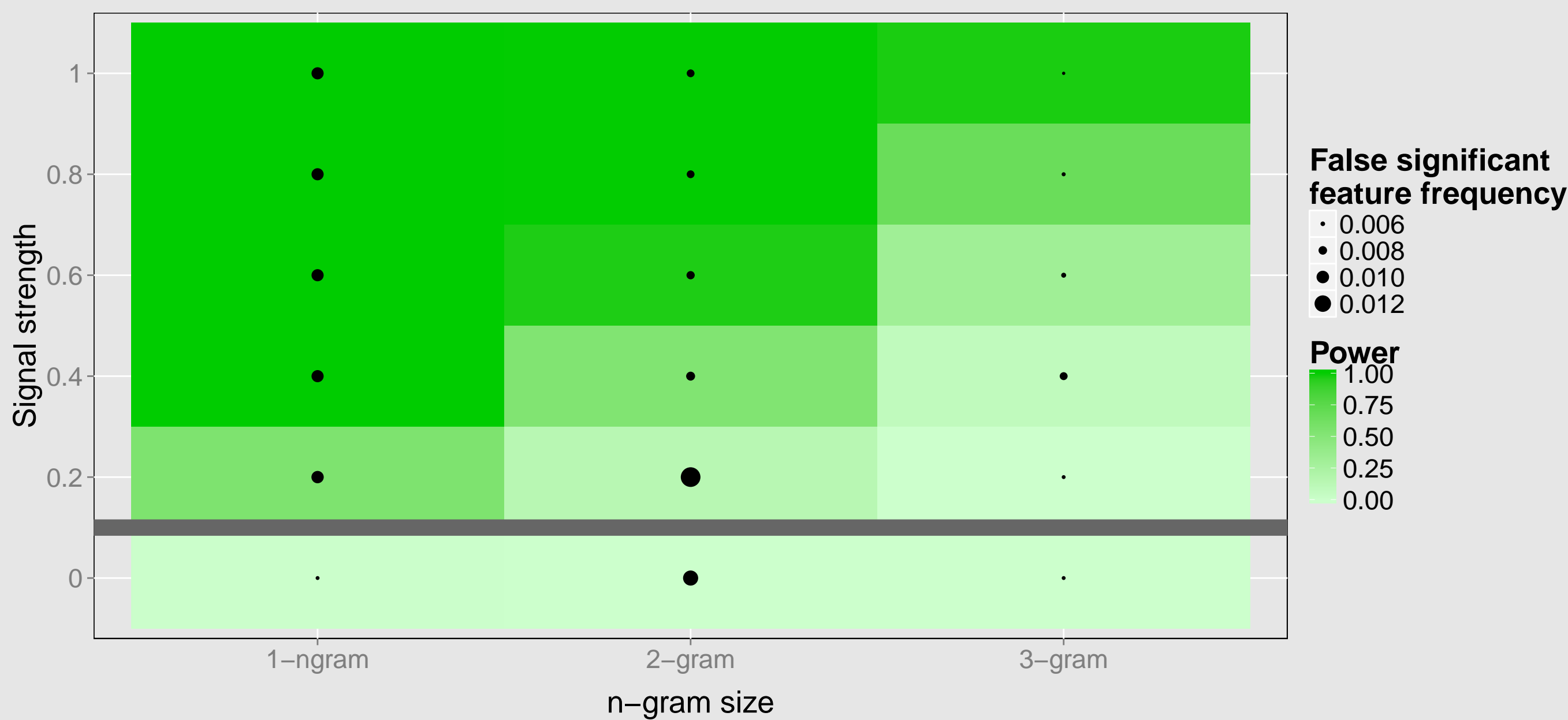


	Target	Feature	Freq
1	0	0	40
2	1	0	10
3	0	1	25
4	1	1	40

## Simulation scheme

1. Random 4000 sequences (20 nucleotides each). The half of the sequences has label 0.
2. Choose a single position between 3 and 18 (to avoid border cases).
3. Resample nucleotides at chosen position. The dominant nucleotide has probability of occurence  $p_d = 0.25$ . Other nucleotides have probability of occurence  $p_o = (1 - p_d)/3$ .
4. Perform QuiPT (Information Gain) and choose significant features (with p-value  $< 0.001$ ).
5. Iterate steps 1-4 over other values of  $p_d$  - 0.38, 0.51, 0.65, 0.78, 0.91.
6. Repeat steps 1-5 200 times.

## Power and False discoveries



## Summary

Quick permutation test is a powerful and quick equivalent of permutation test in binary feature-binary target testing scenario.

## Availability

biogram R package:  
<http://cran.r-project.org/web/packages/biogram/>

## Bibliography

Fang, Y.-C., Lai, P.-T., Dai, H.-J., and Hsu, W.-L. (2011). Meinfoxtex 2.0: gene methylation and cancer relation extraction from biomedical literature. *BMC Bioinformatics*, 12(1):471.  
Guo, S.-H., Deng, E.-Z., Xu, L.-Q., Ding, H., Lin, H., Chen, W., and Chou, K.-C. (2014). inuc-pseknrc: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. *Bioinformatics*, 30(11):1522–1529.  
Wang, Y., Liu, L., Chen, L., Chen, T., and Sun, F. (2014). Comparison of metatranscriptomic samples based on *italic*<sub>k</sub>-*italic*<sub>i</sub>tuple frequencies. *PLoS ONE*, 9(1):e84348.