Quick Permutation Test: feature filtering of n-gram data

Piotr Sobczyk[1]*, Michał Burdukiewicz[2], Chris Lauber[3], Paweł Mackiewicz[2]
*Piotr.Sobczyk@pwr.edu.pl

[1]Wrocław University of Technology, Institute of Mathematics and Computer Science, Poland
[2]University of Wrocław, Department of Genomics, Poland
[3]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.

	P1	P2	P3	P4	P5	P6
S1	4	3	1	3	1	3
S2	2	3	1	1	4	4
S3	4	3	1	3	4	4

Sample sequences.

1	2	3	4
2	0	3	1
2	1	1	2
1	0	2	3

Unigram counts.

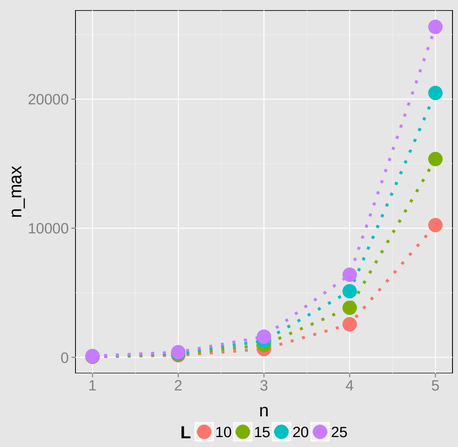
P1_1	P2_1	P3_1	P4_1	P5_1	P6_1	P1_2	P2_2	P3_2	P4_2	P5_2
0	0	1	0	1	0	0	0	0	0	0
0	0	1	1	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0

A fraction of possible unigrams with position information.

Curse of dimensionality

Number of possible positioned n-grams (not taking into account distances between elemnts of n-gram):

$$n_{max} = L \times m^n$$



Permutation test

During permutation tests 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 model and statistic independent, but computationally expensive (especially precise estimation of low p-values, because the number of permutations is inversely proportional to the interval between p-values).

QuiPT algorithm

If probability that target equals 1 is p and probability that feature equals 1 is q and feature and target are independent then each of them has the following probabilities

$$P(Target, Feature) = (1,1)) = p \cdot q$$
$$P(Target, Feature) = (1,0)) = p \cdot (1-q)$$
$$P(Target, Feature) = (0,1)) = (1-p) \cdot q$$
$$P(Target, Feature) = (0,0)) = (1-p) \cdot (1-q)$$

$$F(n_{1,1}, n_{1,0}, n_{0,1}, n_{0,0}) = \binom{n}{n_{1,1}} (p \cdot q)^{n_{1,1}} n - n_{1,1} \binom{}{n_{1,0}} (p \cdot (1-q))^{n_{1,0}} n - n_{1,1} - n_{1,0} \binom{}{n_{0,1}} ((1-p) \cdot q)^{n_{0,1}} \binom{n - n_{1,1} - n_{1,0} - n_{0,1}}{n_{0,0}} ((1-p) \cdot (1-q))^{n_{0,0}}$$

In addition to this: $n_{1,.} = n_{1,1} + n_{1,0}$ and $n_{.,1} = n_{1,1} + n_{0,1}$ are known and fixed.

Validation procedure

1. Chose randomly (without replacement) 3816 proteins without signal peptides, reshuffle 3816 proteins with signal peptides.
2. Perform 5-fold cross-validation.
3. Repeat step 1. and 2. 250 times.

Validation results

```
## Error in t(poster_data[["metrics"]][c("AUC", "H", "Gini",
"Recall", "Spec", :  object 'poster_data' not found
## Error in ncol(metrics):  object 'metrics' not found
## Error in is.factor(x):  object 'melted_metrics' not found
## Error in levels(melted_metrics[["metric"]]) <- c("AUC",
"H-measure", "Gini", :  object 'melted_metrics' not found
## Error in ggplot(melted_metrics, aes(x = metric, y = value)):
object 'melted_metrics' not found
```

The mean AUC yielded by cross-validation is

```
## Error in eval(expr, envir, enclos):  object 'poster_data' not
found
## Error in eval(expr, envir, enclos):  object 'position_data'
not found
## Error in ggplot(position_data, aes(x = Var1, y = Freq)):
object 'position_data' not found
```

Comparision with other signal peptide predictors

Benchmark data set: 140 eukaryotic proteins with signal peptide and 280 randomly chosen eukaryotic proteins without signal peptide added after 2010.

signal.hsmm1987: trained on data set of 496 eukaryotic proteins with signal peptides added before year 1987.

signal.hsmm2010: trained on data set of 3676 eukaryotic proteins with signal peptides added before year 2010.

Comparision of various software

STH

Summary

Hidden semi-Markov models can be used to accurately predict the presence of secretory signal peptides effectively extracting information from very small data sets.

Avaibility

signal.hsmm web server:

`http://michbur.shinyapps.io/signalhsmm/`

signal.hsmm R package:

`http://cran.r-project.org/web/packages/signal.hsmm/`

Bibliography

Hegde, R. S. and Bernstein, H. D. (2006). The surprising complexity of signal sequences. *Trends in Biochemical Sciences*, 31(10):563–571.

Jain, R. G., Rusch, S. L., and Kendall, D. A. (1994). Signal peptide cleavage regions. functional limits on length and topological implications. *The Journal of Biological Chemistry*, 269(23):16305–16310.

Moeller, L., Gan, Q., and Wang, K. (2009). A bacterial signal peptide is functional in plants and directs proteins to the secretory pathway. *Journal of Experimental Botany*, 60(12):3337–3352.

Nagano, R. and Masuda, K. (2014). Establishment of a signal peptide with cross-species compatibility for functional antibody expression in both escherichia coli and chinese hamster ovary cells. *Biochemical and Biophysical Research Communications*, 447(4):655 – 659.

Nielsen, H. and Krogh, A. (1998). Prediction of signal peptides and signal anchors by a hidden markov model. *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 6:122–130.