Calculating rfPred scores with package rfPred

Hugo Varet, Fabienne Jabot-Hanin and Jean-Philippe Jais

October 13, 2014

1 Background

Exome sequencing is becoming a standard tool for gene mapping of monogenic diseases. Given the vast amount of data generated by Next Generation Sequencing techniques, identification of disease causal variants is like finding a needle in a haystack. The impact assessment and the prioritization of potential pathogenic variants are expected to reduce work in biological validation, which is long and costly.

One of the possible approaches to determine the most probable deleterious variants in individual exomes is to use protein function alteration prediction algorithms. This package proposes a new method [1] based on five previously described algorithms (SIFT [2], Polyphen2 [3], LRT [4], PhyloP [5] and MutationTaster [6]) compiled in the dbNSFP database [7]. A functional meta-score is derived from a random forest method trained on a dataset of 61,500 non-synonymous SNPs. On Two independent validation datasets, the random forest method appears to be globally better than each of the algorithms separately or in combination in a logistic regression model. rfPred scores have been pre-calculated and made available for all the possible non-synonymous SNPs of human exome.

2 Computing rfPred scores

After having launched the *R* software, the user must load the *rfPred* package with:

```
> library(rfPred)
```

Several packages whose *rfPred* is depending on will also be loaded (especially *Rsamtools*). The package contains one main function - `rfPred_scores()` - which returns the rfPred scores. It works as follows: the user gives as input a list of variants for which he/she wants the corresponding scores and the function looks for these variants in a data base which stores the pre-calculated scores. It avoids re-calculating the scores for each request. The list of variants can ben either a *data.frame*, or the path to a VCF (Variant Call Format) file as a *character* string or a *GRanges* object (see package *GenomicRanges* on *Bioconductor* [8]). For example, the list of variants can be the data frame `variant_list_Y`:

```
> data(variant_list_Y)
> print(variant_list_Y)
```

```
  chr      pos ref alt  uniprot
1   Y 21869371   C   A Q9BY66-2
2   Y  6736929   T   G   Q99218
3   Y  6736330   C   G   Q99218
4   Y 21869957   C   G   Q9BY66
5   Y  6736294   G   C   Q99218
```

`variant_list_Y` is a *data.frame* which contains four or five columns respecting this order: the chromosome, the HG19 position on the chromosome, the reference nucleotid allele, the alteration nucleotid allele and optionally the protein (Uniprot accession number). Here, this *data.frame* is part of the package, but it is possible to import data from an Excel or CSV file in *R* (see *R* documentation). The user can then run the function:

```
> result=rfPred_scores(variant_list=variant_list_Y,
+                      data="http://www.sbim.fr/rfPred/all_chr_rfPred.txtz",
+                      index="http://www.sbim.fr/rfPred/all_chr_rfPred.txtz.tbi")
> print(result)
```

```
  chromosome position_hg19 reference alteration proteine aaref aaalt rfPred_score
1          Y       6736294         G          C   Q99218     N     K       0.1318
2          Y       6736330         C          G   Q99218     Q     H        0.114
3          Y       6736929         T          G   Q99218     N     H        0.058
4          Y      21869371         C          A Q9BY66-2     A     S       0.0938
5          Y      21869957         C          G   Q9BY66     V     L       0.3432
  SIFT_score MutationTaster_score Polyphen2_score      PhyloP_score LRT_score aapos
1       0.86              0.08857           0.364 0.956751604061215 0.9963805   133
2       0.92             0.001443           0.115 0.167097520013057 0.9932325   121
3       0.99             0.000555           0.828 0.948123579209099 0.4416835    42
4       0.96             0.008922            0.81 0.938202628327774 0.9960675   193
5       0.98             0.460181           0.995 0.916829367481492  0.999998    60
```

The argument `data` is the path to the data base (TabixFile) and the argument `index` is the path to the TabixFile index. Note that the default paths are those to send queries on the server (see section 3 for details) and thus need an Internet connection.

It is also possible to look for variants without specifying the protein. In this case, the user gives only the columns 1-4 as input:

```
> result2=rfPred_scores(variant_list=variant_list_Y[,1:4],
+                       data="http://www.sbim.fr/rfPred/all_chr_rfPred.txtz",
+                       index="http://www.sbim.fr/rfPred/all_chr_rfPred.txtz.tbi")
> print(result2)
```

```
   chromosome position_hg19 reference alteration proteine aaref aaalt rfPred_score
1           Y       6736294         G          C   Q99218     N     K       0.1318
2           Y       6736294         G          C Q99218-1     N     K       0.1564
3           Y       6736330         C          G   Q99218     Q     H        0.114
4           Y       6736330         C          G Q99218-1     Q     H       0.1156
5           Y       6736929         T          G   Q99218     N     H        0.058
6           Y      21869371         C          A   B7ZLX1     A     S       0.1016
7           Y      21869371         C          A   E9PFH2     A     S       0.1274
8           Y      21869371         C          A   Q9BY66     A     S        0.099
9           Y      21869371         C          A Q9BY66-2     A     S       0.0938
10          Y      21869957         C          G   B7ZLX1     V     L       0.4906
11          Y      21869957         C          G   E9PFH2     V     L       0.4906
12          Y      21869957         C          G   Q9BY66     V     L       0.3432
13          Y      21869957         C          G Q9BY66-2     V     L       0.4816
   SIFT_score MutationTaster_score Polyphen2_score      PhyloP_score LRT_score aapos
1        0.86              0.08857           0.364 0.956751604061215 0.9963805   133
2        0.95              0.08857           0.249 0.956751604061215 0.9963805   133
3        0.92             0.001443           0.115 0.167097520013057 0.9932325   121
4        0.92             0.001443           0.132 0.167097520013057 0.9932325   121
5        0.99             0.000555           0.828 0.948123579209099 0.4416835    42
6        0.96             0.008922           0.696 0.938202628327774 0.9960675   193
7        0.92             0.008922            0.07 0.938202628327774 0.9960675   193
8        0.75             0.008922            0.88 0.938202628327774 0.9960675   193
9        0.96             0.008922            0.81 0.938202628327774 0.9960675   193
10          1             0.460181           0.997 0.916829367481492  0.999998    60
11          1             0.460181           0.997 0.916829367481492  0.999998    60
```

```
12       0.98              0.460181           0.995 0.916829367481492  0.999998     60
13          1              0.460181           0.998 0.916829367481492  0.999998     60
```

The object `result2` contains more lines (13) than the object `result` (5) because the user does not impose any matching on the protein.

3 TabixFile and TabixFile index

The user can search in the TabixFile/index located on the server (http://www.sbim.fr/rfPred/) or can download them (about 3.3 Go) in order to use them locally on his/her computer. In this case, the arguments `data` and `index` must be the paths to the TabixFile and index on the user's computer. Note that the function runs faster using the data base locally than on the server since it is not necessary to go through the Internet. Moreover, we limited requests on the server to variants lists with less than 1000 rows.

4 Exporting the results

In order to share the results with non-*R* users, the `rfPred_scores()` function allows one to export the results in a CSV file thanks to the `file.export` argument:

```
> result3=rfPred_scores(variant_list=variant_list_Y,
+                       data="http://www.sbim.fr/rfPred/all_chr_rfPred.txtz",
+                       index="http://www.sbim.fr/rfPred/all_chr_rfPred.txtz.tbi",
+                       file.export="results.csv")
```

The CSV file "results.csv" will be created in the current working directory of the *R* session.

5 Number of cores

Computers with multi-core processors are able to run several tasks simultaneously. The `n.cores` argument of the `rfPred_scores()` function exploits this possibility. Specifying `n.cores=4` instead of `n.cores=1` divides the computational time by about 2.5. For example, the function ran during 106 second on a 121,606 rows variants list using a 3.3 GHz Intel®Core™i5 computer. However, for small requests, one should use only one core to avoid the overhead time due to opening hidden *R* sessions and lauching the dependent packages.

6 rfPred random forest model

One may want to compute the rfPred score from his/her own SIFT, Polyphen2, LRT, PhyloP and MutationTaster scores. This functionnality does not appear within the package but we have made the model available on our website (http://www.sbim.fr/rfPred/) in a .RData file (about 52 Mo). The *R* object `rfPred_model` has to be given as input of the `predict` function of the *randomForest* package in addition of the 5 scores. The scores used to build the random forest are [9]:

- SIFT_score $= 1-$original SIFT score;
- Polyphen2_score $=$ original HVAR Polyphen2 score;
- MutationTaster_score $=$ original MutationTaster score;
- PhyloP_score $= 1 - 0.5 \times 10^{\text{phyloP}}$ if phyloP ≥ 0 or $0.5 \times 10^{\text{-phyloP}}$ if phyloP < 0;
- LRT_score $= 1 - 0.5 \times$ LRToriginal if LRT_Omega < 1 or $0.5 \times$ LRToriginal if LRT_Omega ≥ 1.

Note that all these scores have to be stored in a *data.frame* whose column names are textually `SIFT_score`, `PhyloP_score`, `Polyphen2_score`, `LRT_score` and `MutationTaster_score` in order to match with the model parameters.

References

[1] Jabot-Hanin F and Varet H and Tores F and Jais JP, *rfPred: a new meta-score for functional prediction of missense variants in human exome*. (submitted) 2013.

[2] Kumar P and Henikoff S and Ng PC, *Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm*. Nat Protoc 2009, VOL.4 NO.8.

[3] Adzhubei IA and Schmidt S and Peshkin L et al, *A method and server for predicting damaging missense mutations*. Nat Methods 2010 7(4):248-249.

[4] Muse SV and Gaut BS, *A Likelihood Approach for Comparing Synonymous and Nonsynonymous Nucleotide Substitution Rates, with Application to the Chloroplast Genome*. Mol Biol Evol 1994 1(5):175-724.

[5] Margulies EH and Cooper GM and Asimenos G et al, *Analyses of deep mammalian sequence alignments and constraint predictions for 1% of the human genome*. Genome Res 2007 17:760-774

[6] Schwarz JM and Rodelsperger and Schuelke M and Seelow D, *MutationTaster evaluates diseasecausing potential of sequence alterations*. Nat Methods 2010 7(8)

[7] Liu X and Jian X and and Boerwinkle E, *dbNSFP: a lightweight database of human non-synonymous SNPs and their functional predictions*. Hum Mutat 2011 32:894-899.

[8] Aboyoun P and Pages H and Lawrence M, *GenomicRanges: Representation and manipulation of genomic intervals*. R package version 1.12.4.

[9] dbNSFP v1.3 READ-ME, http://dbnsfp.houstonbioinformatics.org/dbNSFPzip/dbNSFPv1.3.readme.txt.