Probabilistic 20/20 Documentation

Release 1.2.3

Collin Tokheim

Jul 17, 2019

Contents

Author Collin Tokheim

Contact ctokhei1 AT alumni.jh.edu

Source code GitHub

Q&A Biostars (tag: prob2020)

The Probabibilistic 20/20 statistical test identifies genes with signficant oncogene-like and tumor suppressor gene-like mutational patterns for small somatic variants in coding regions. Putative signficant oncogenes are found through evaluating missense mutation clustering and *in silico* pathogenicity scores. Often highly clustered missense mutations are indicative of activating mutations. While statistically signficant tumor suppressor genes (TSGs) are found by abnormally high proportion of inactivating mutations.

Probabilistic 20/20 evaluates statistical significance by employing monte carlo simulations, which incorporates observed mutation base context. Monte carlo simulations are performed within the same gene and thus avoid building a background distribution based on other genes. This means that the statistical test can be applied to either all genes in the exome from exome sequencing or to a certain target set of genes from targeted sequencing. Additionally, the direct results of somatic mutation simulations can be accessed in a Mutation Annotation Format (MAF) file.

The Probabilistic 20/20 test has nice properties since it accounts for several factors that could effect the significance of driver genes.

- gene length
- mutation context
- gene sequence (e.g. codon bias)

Contents:

CHAPTER 1

Download

1.1 Probabilistic 20/20 releases

You can download the package below, or install directly from github (see *Installation*).

- probabilistic2020-v1.2.3 7/17/2019 Minor fixes
- probabilistic2020-v1.2.2 2/18/2019 Fixed installation bug caused by cython on python3.7
- probabilistic2020-v1.2.1 2/14/2019 Fixed installation bug
- probabilistic2020-v1.2.0 10/8/2017 Improved hotmaps 1d efficiency and added capability to drop silent mutations in simulations
- probabilistic2020-v1.1.1 5/21/2017 Fixed bug for newer releases of numpy
- probabilistic2020-v1.1.0 12/7/2016 Added HotMAPS 1D algorithm to find codons with significant clustering of missense mutations
- probabilistic2020-v1.0.7 11/29/2016 Fixed simulated indel bug
- probabilistic2020-v1.0.6 11/7/2016 Major bug fix in simulations (users should upgrade version)
- probabilistic2020-v1.0.5 10/3/2016 Fixed python3 conversion error in pickle module
- probabilistic2020-v1.0.4 8/16/2016 Fixed mutation uniquing
- probabilistic2020-v1.0.3 8/5/2016 Fixed p-value calculation bug in 1.0.1
- probabilistic2020-v1.0.2 6/26/2016 Fixed installation on python 3
- probabilistic2020-v1.0.1 6/14/2016 Improved memory handling
- probabilistic2020-v1.0.0 5/1/2016 Initial release

1.2 Example data

- Example pancreatic adenocarcinoma data set

1.3 Other

- Pre-computed scores data set
- Reference SNVBox transcripts in BED format

CHAPTER 2

Installation

2.1 Python Package Installation

Using the python package installation, all the required python packages for the probabibilistic 20/20 test will automatically be installed for you. We recommend use of python version 3.6, if possible.

To install the package into python you can use *pip*. If you are installing to a system wide python then you may need to use *sudo* before the pip command.

```
$ pip install probabilistic2020
```

The scripts for Probabilstic 20/20 can then be found in *Your_Python_Root_Dir/bin*. You can check the installation with the following:

```
$ which probabilistic2020
$ probabilistic2020 --help
```

2.2 Local installation

Local installation is a good option if you do not have privilege to install a python package and already have the required packages. The source files can also be manually downloaded from github at https://github.com/KarchinLab/probabilistic2020/releases.

Required packages:

- numpy
- scipy
- pandas>=0.17.0

- pysam

If you don't have the above required packages, you will need to install them. For the following commands to work you will need pip. If you are using a system wide python, you will need to use *sudo* before the pip command.

```
$ cd probabilstic2020
$ pip install -r requirements.txt
```

Next you will need to build the Probabilistic 20/20 source files. This is can be accomplished in one command.

```
$ make build
```

Once finished building you can then use the scripts in the *probabilstic2020/prob2020/console* directory. You can check the build worked by the following:

```
$ python prob2020/cosole/probabilistic2020.py --help
```

CHAPTER 3

Quick Start

The quick start is meant to test that everything is working with the installation of the probabilistic2020 package. This provides running probabilistic2020 with the minimum number of steps to execute the statistical test. For more expansive user instructions see *Tutorial*.

3.1 Installation

Please see the *Installation*.

3.2 Downloading Example

Download the quick start example data, and extract the resulting tarball.

```
$ wget http://karchinlab.org/data/2020+/pancreatic_example.tar.gz
$ tar xvzf pancreatic_example.tar.gz
$ cd pancreatic_example
```

3.3 Input files

3.3.1 Gene BED annotation

BED gene annotation files should contain a single reference transcript per gene. The name field in the BED file should contain the gene name (not the transcript). An example BED file containg the annotations for the largest transcripts in SNVBox is named snvboxGenes.bed.

3.3.2 Gene FASTA

Gene sequences are extracted from a genome FASTA file, and is a step that only needs to be done once. This has already been done for the example BED file provided, but if you were to use a different transcript annotation then you would need to follow the *Gene FASTA*.

3.3.3 Mutation Annotation Format (MAF) file

Mutations are saved in a MAF-like format. Not All fields in MAF spec are required, and columns may be in any order. Mutations for pancreatic adenocarcinoma are in the file pancreatic_adenocarcinoma.txt.

3.4 Running the Example

To execute the statistical test for TSG-like genes by examining elevated proportion of inactivating mutations, the **tsg** sub-command for **probabilistic2020** is used. To limit the run time for this example, you can limit the number of iterations to 10,000 with the **-n** parameter. You can further speed up the example by using multiple computer cores with the **-p** parameter.

```
$ probabilistic2020 tsg \
    -n 10000 \
    -i snvboxGenes.fa \
    -b snvboxGenes.bed \
    -m pancreatic_adenocarcinoma.txt \
    -o pancreatic_output_comparison.txt
```

Your results should match those found in the file pancreatic_output.txt. Particularly, TP53, SMAD4, ARID1A, and SMARCA4 should have a significant inactivating Benjamini-Hochberg (BH) q-value of less than .1.

CHAPTER 4

Tutorial

Probabilistic 20/20 consists of two broad statistical tests (oncogene-like and tsg-like) and somatic mutation simulation framework. Internally, the simulation framework is used to establish statistical significance in the hypothesis test through the **probabilistic2020** command. However, the simulation framework through the **mut_annotate** command can also be used to create a simulated MAF file where aferwords all mutations are distributed like passengers based on uniform null distribution. Moreover, a set of mutational features for each gene representative of driver genes (used in 20/20+) can also be created.

4.1 Input formats

4.1.1 Mutations

Mutations are provided in a Mutation Annotation Format (MAF) file. Columns can be in any order, and only a few columns in the MAF file are needed. The following is a list of the required columns.

- Hugo_Symbol (or named "Gene")
- Chromosome
- Start_Position
- End_Position
- Reference_Allele
- Tumor_Seq_Allele2 (or named "Tumor_Allele")
- Tumor_Sample_Barcode (or named "Tumor_Sample")
- Variant_Classification

The remaining columns in the MAF specification can be left empty or not included. Since a MAF file has many additional annotation columns, removing additional columns will reduce the memory usage of probabilistic2020.

Only coding variants found in the Variant_Classification column will be used, which includes the following: 'Missense_Mutation', 'Silent', 'Nonsense_Mutation', 'Splice_Site', 'Nonstop_Mutation', 'Translation_Start_Site',

'Frame_Shift_Ins', 'Frame_Shift_Del', 'In_Frame_Ins', 'In_Frame_Del', 'Frame_Shift_Indel', or 'In_Frame_Indel'. Note, although 'In_Frame_Indel' and 'Frame_Shift_Indel' are not official MAF specification values, for the purpose of this program represent either and insertion or deletion. Other values for the Variant_Classification column are assumed to be non-coding, and dropped from the analysis.

4.1.2 Gene BED file

A single reference transcript for each gene is stored in BED12 format. Instead of using the transcript name for the name field in the BED file, the gene symbol which matches the MAF file should be used. In the example data, the longest CDS transcript from SNVBox was used.

4.1.3 Gene FASTA

Gene sequences are extracted from a genome FASTA file, and is a step that only needs to be done once. To do this, you need a BED file with names corresponding to genes, and a genome FASTA (e.g. hg19). You can download hg19 from here. Creating the gene sequence FASTA is then done by the *extract_gene_seq* script:

```
$ extract_gene_seq -i hg19.fa -b snvboxGenes.bed -o snvboxGenes.fa
```

In this case the BED file is created using SNVBox, a genome FASTA file for hg19 (hg19.fa), and the resulting coding sequences for the gene are stored in snvboxGenes.fa.

4.1.4 Pre-computed scores (optional)

Two pre-computed scores are used to evaluate missense pathogenicity scores and evolutionary conservation. Both are provided in the example data, matching the reference transcript annotation from SNVBox. Including the score information is useful, but optional. The pre-computed missense pathogenicity scores are from the VEST algorithm. The evolutionary conservation scores are calculated as the entropy of a specific column in the protein-translated version of UCSC's 46-way vertebrate alignment.

4.2 Running the statistical test

The statistical tests account for gene sequence and mutational base context. Each gene is represented by a single reference transcript (above is longest CDS SNVBox transcript). By default the relevant sequence context for mutations are utilized from CHASM paper (denoted by **-c 1.5** parameter). This includes some common dinucletoide contexts like CpG, and otherwise just a single base. Ultimately a multiple testing corrected q-value is reported using the Benjamini-Hochberg (BH) method.

Technical detail: Running on the obtained pan-cancer data may take several hours to run on a single core. Specifying the **-p** parameter to use multiple processors will speed up run time if available. Lowering the number of iterations (default: 100,000) will decrease run time, but also decrease the resolution of p-values.

4.2.1 Running oncogene sub-command

The oncogene sub-command examines missense position clustering (by codon) and elevated *in silico* pathogenicity scores (VEST). The score directory contains pre-computed values for VEST scores. The p-values will be combined using fisher's method to report a single p-value with a BH FDR. In the below example, the command is parallelized onto 10 processors with the **-p** parameter. Lower this if the compute is not available.

```
$ probabilistic2020 oncogene \
    -i genes.fa \
    -b genes.bed \
    -s score_dir \
    -m mutations.txt \
    -c 1.5
    -p 10 \
    -o oncogene_output.txt
```

Where genes.fa is your gene FASTA file for your reference transcripts in genes.bed, mutations.txt is your MAF file containing mutations, score_dir is the directory containing the pre-computed VEST scores, and oncogene_output.txt is the file name to save the results.

Output format

The oncogene statistical test will output a tab-delimited file having columns for the p-values and Benjamini-Hochberg q-values:

- "entropy"
- "vest" (only included if score_dir provided)
- "combined" (only included if score_dir provided)

The entropy columns evaluate missense clustering at the same codon by using a normalized missense position entropy statistic. Low values for entropy correspond to increased clustering of missense mutations. The vest columns examine whether missense mutations tend to have higher *in silico* pathogenicity scores for missense mutations than expected. The "combined" columns, combine the p-values from VEST scores and missense clustering using Fisher's method.

4.2.2 Running tsg sub-command

The **tsg** sub-command evaluates for elevated proportion of inactivating point mutations to find TSG-like genes.

```
$ probabilistic2020 tsg \
    -i genes.fa \
    -b genes.bed \
    -m mutations.txt \
    -p 10 \
    -c 1.5 \
    -o tsg_output.txt
```

Where genes.fa is your gene FASTA file for your reference transcripts in genes.bed, mutations.txt is your MAF file containing mutations, and tsg_output.txt is the file name to save the results.

Output format

The tsg statistical test examines inactivating single nucleotide variants (nonsense, splice site, lost start, and lost stop). Both the p-value ("inactivating p-value") and the Benjamini-hochberg q-value ("inactivating BH q-value") are reported for a higher than expected fraction of inactivating mutations. Mutations which could not be placed onto the reference transcript will be indicated in the "SNVs Unmapped to Ref Tx" column.

4.2.3 Running hotmaps1d sub-command

The **hotmaps1d** sub-command evaluates particular amino acid residues for elevated cluster of missense mutations in the protein sequence.

```
$ probabilistic2020 hotmaps1d \
    -i genes.fa \
    -b genes.bed \
    -m mutations.txt \
    -w 3 \
    -p 10 \
    -c 1.5 \
    -o hotmaps1d_output.txt
```

Where genes.fa is your gene FASTA file for your reference transcripts in genes.bed, mutations.txt is your MAF file containing mutations, and hotmaps1d_output.txt is the file name to save the results. HotMAPS 1D also takes a window size for examining missense mutation clustering. In the above example, the parameter **-w 3** considers 3 residues on either side of each mutated residue. A large number of mutations in this small window may indicate the mutations form a "hotspot", and likely contain driver mutations at the mutated residue. The window size can be changed depending on the preferred granularity of the analysis.

Output format

The hotmaps1d statistical test examines the position of missense mutations in sequence. Both the p-value ("p-value") and the Benjamini-hochberg q-value ("q-value") are reported for a higher than expected ammount of missense mutations within a given window around a mutation. The "mutation count" column reports how many missense mutations were observed at the particular codon, and the "windowed sum" column reports how many missense mutations were observed in a sequence window encompassing the particular codon.

4.3 Simulating somatic mutations

The probabilistic2020 package also allows saving the results of underlying simulation of somatic mutations. The simulations need a set of observed mutations to create simulated mutations. Briefly, for each gene, SNVs (single nucleotide variants) are moved with uniform probability to any matching position in the gene sequence, holding the total number of SNVs fixed. A matching position was required to have the same base context (e.g. **-c 1.5** = C*pG, CpG*, TpC*, G*pA, A, C, G, T) as the observed position. This method of generating a null distribution controls for the particular gene sequence, gene length and mutation base context. To simulate small insertions/deletions (indels), indels are moved to different genes according to a multinomial model where the probability is proprotional to the gene length. This can be done for both creating a simulated MAF file or simulated features calculated from the mutations.

Simulations are performed with the **mut_annotate** command. The **–seed** parameter will pass a seed to the pseudo random number generator. If you are performing several simulations for MAF files and features, then it is critical that every time the seed for each simulation match.

4.3.1 Simulated MAF

MAF output is designated with the **–maf** flag, but is a substantially reduced version then a typical MAF file because it only contains the relevant columns noted in the mutations input format section. To indicate mutations for each gene should be simulated once, the **-n 1** parameter is used. If zero is supplied for this parameter, then simulations are not performed and rather the observed mutations are just annotated as a MAF file on the corresponding reference transcripts in genes.bed. The pseudo random number generator seed can be passed with the **–seed** argument.

```
$ mut_annotate \
     --maf \
     -n 1 \
     -i genes.fa \
     -b genes.bed \
     -m mutations.txt \
     -p 10 \
     -c 1.5 \
     -o maf_output.txt
```

4.3.2 Simulated Features

Simulated features which serve as input to 20/20+ can also be generated.

```
$ mut_annotate \
     --summary \
     -n 1 \
     -i genes.fa \
     -b genes.bed \
     -m mutations.txt \
     -p 10 \
     -c 1.5 \
     -o summary_output.txt
```

CHAPTER 5

FAQ

Who should I contact if I encounter a problem?

If you believe your problem may be encountered by other users, please post the question on biostars. Check to make sure your question has not been already answered by looking at posts with the tag prob2020. Otherwise, create a new post with the prob2020 tag. We will be checking biostars for questions. You may also contact me directly at ctokheim AT jhu dot edu.

What python version should I use?

We recommend using python 3.6, as the package has been extensively tested on this version of python.

CHAPTER 6

Citation

Collin J. Tokheim, Nickolas Papadopoulos, Kenneth W. Kinzler, Bert Vogelstein, and Rachel Karchin. Evaluating the evaluation of cancer driver genes. PNAS 2016 ; published ahead of print November 22, 2016, doi:10.1073/pnas.1616440113

If you use the hotmaps1d command to find codons were missense mutations are significantly clustered, please cite the HotMAPS paper:

Tokheim C, Bhattacharya R, Niknafs N, Gygax DM, Kim R, Ryan M, Masica DL, Karchin R (2016) Exome-scale discovery of hotspot mutation regions in human cancer using 3D protein structure Cancer Research. Apr. 28.pii: canres.3190.2015.