PaBScan: Selection outlier scan with population branch statistic

Tuomas Hämälä

27th of July 2017

tuomas.hamala@gmail.com

0. Quick start

```
$ git clone https://github.com/thamala/PaBScan.git

$ cd PaBScan

$ mv pabscan.* example_data

$ gcc pabscan.c -lm -o pabscan

$ ./pabscan -likes example.likes -pop1 list1.txt -pop2 list2.txt -pop3 list3.txt
-out output

$ head output.pbs

Chromo     Position   PBS1       PBS2       PBS3
1          220623     0.319226   -0.120292  0.211402
1          445587     0.053947   -0.065290  0.070898
1          734592     -0.028352  0.001156   -0.043844
1          825157     -0.173893  0.251893   0.326334
1          956672     -0.029573  -0.020313  0.212964
1          1069389    0.238198   -0.154341  0.261004
1          1154260    0.192379   -0.125199  0.281174
1          1383999    0.037803   -0.062909  0.448616
1          1587647    0.323904   -0.285808  1.113377
```

1. Background

PaBScan is a program used for detecting selection outliers with population branch statistic (PBS). This measure, introduced by Yi *et al.* (2010), is based on comparing divergence estimates between three populations, two focal ones and an outgroup. PaBScan works with diploid NGS data in either genotype call or genotype likelihood formats.

PaBScan is a command line tool written in C and is designed to run in UNIX or UNIX-like operating systems, such as macOS and Linux. The program can also run in Windows after compiling the code with e.g MinGW, but it has not been tested in that environment.

1.1 Divergence estimates

Two F_{ST} statistics have been implemented into PaBScan: one by Hudson *et al.* (1992) and one by Weir & Cockerham (1984). As default, the pairwise divergence estimates are calculated with Hudson's F_{ST}, using the formula from Bhatia *et al.* (2013):

$$F_{\mathrm{ST}} = \frac{(p_1 - p_2)^2 - \frac{p_1(1-p_1)}{n_1 - 1} - \frac{p_2(1-p_2)}{n_2 - 1}}{p_1(1-p_2) + p_2(1-p_1)}$$

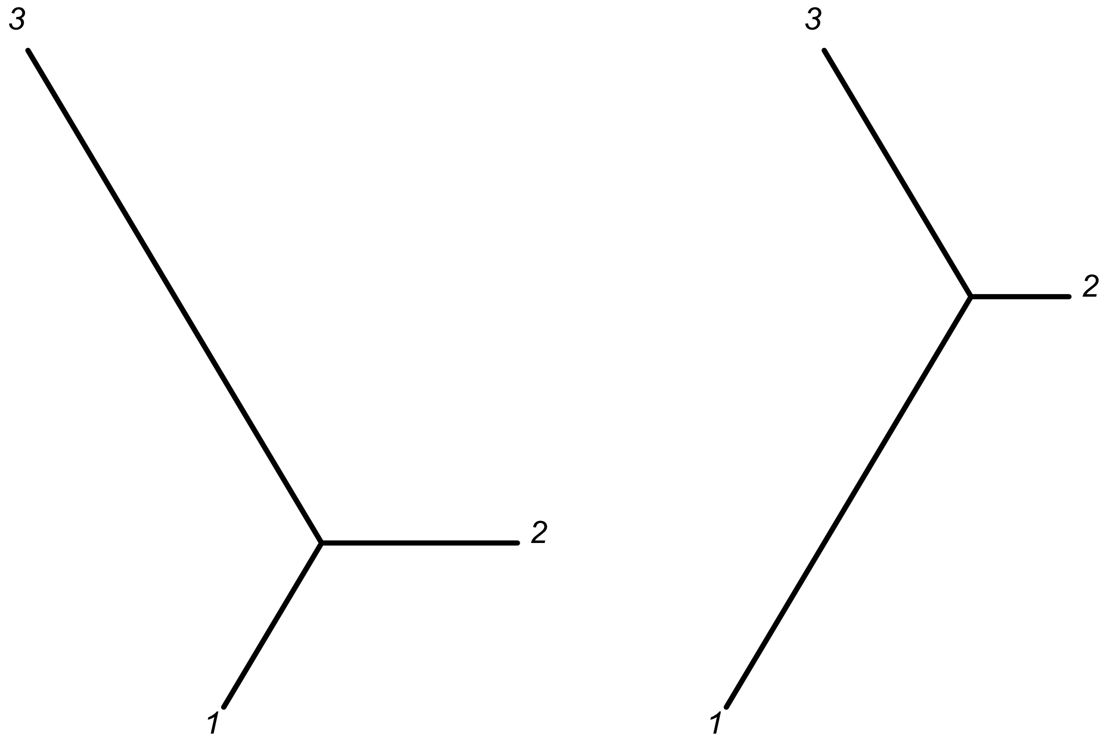
where n_{i} is the sample size and p_{i} is the minor allele frequency in the two populations to be compared. Sliding window estimates are based on the weighting method by Reynolds *et al.* (1983). With Hudson's measure, the weighted F_{ST} is: $1 - (\sum_{i=1}^{n} H_{iW} / \sum_{i=1}^{n} H_{iB})$, where n is the window size. However, being a relative measure, F_{ST} can be inflated by reduced within population nucleotide diversities, so selection scans can also be conducted using an absolute measure, d_{XY} (Nei 1987). To make this estimator compatible with genotype likelihoods, d_{XY} is calculated from allele frequencies using the following formula (here shown for window of size n):

$$d_{XY} = \frac{1}{n} \sum_{i=1}^{n} p_{i1}(1 - p_{i2}) + p_{i2}(1 - p_{i1})$$

These estimates are then transformed into relative divergence times: $T = -\ln(1 - X)$, where X is either F_{ST} or d_{XY}, and the PBS for lineage 1 is calculated as:

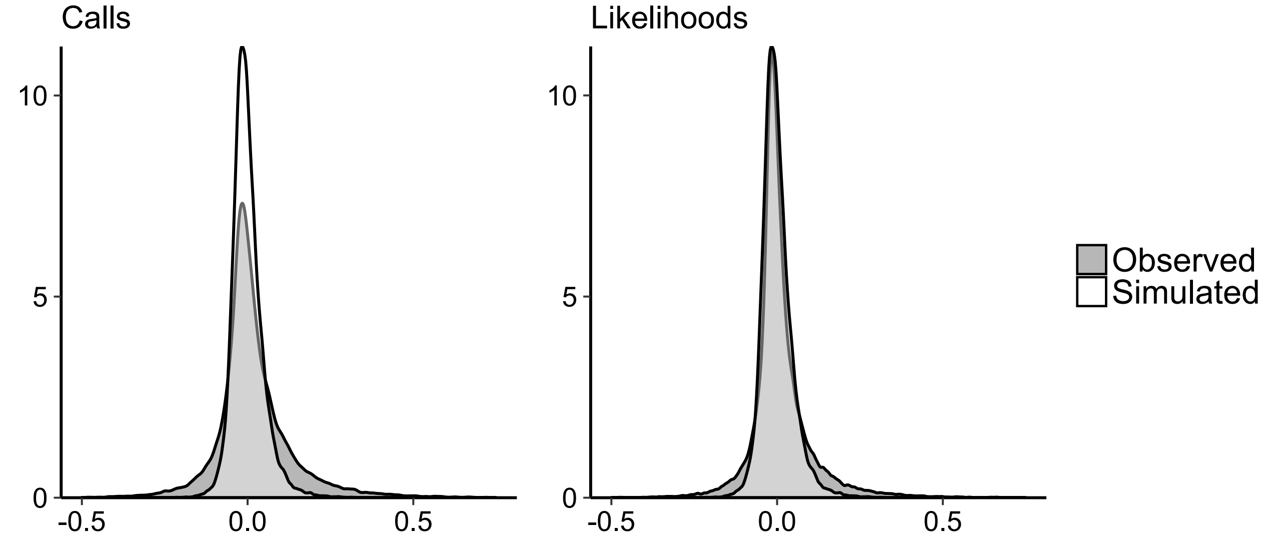
$$\mathrm{PBS} = \frac{T_{12} + T_{13} - T_{23}}{2}$$

The obtained value quantifies the magnitude of allele frequency change in population 1 since its divergence from the closely related population 2 and the outgroup 3. The figure below depicts relative branch lengths in a neutral (left) and selected (right) scenarios.



1.2 Genotype likelihoods

A common issue with NGS data is low and often highly variable sequencing coverage. This fact combined with the strict filtering associated with variant calling (most protocols require the called genotype to be ten times more likely than the other ones) can lead to a scenario where heterozygote calls are clearly underrepresented in areas of low coverage, biasing the sampling distributions and thus effecting outlier detection. To prevent this, PaBScan utilizes a maximum likelihood model by Kim *et al.* (2011) to estimate allele frequencies directly from genotype likelihoods. This approach bypasses the need for genotype calling, leading to unbiased allele frequency estimates even at very low coverage (~2×). Below is a real data example showing PBS distributions estimated from SNP calls and genotype likelihoods compared against simulated neutral samples. The data has a median coverage of 12× and it was filtered using the following settings: mapping quality 30, site quality 20, genotype quality 20 (SNP calls only), minimum coverage 4×.



1.3 Outlier detection

For outlier detection, PaBScan utilises a simulation and Monte Carlo based tests. In the latter method, permutations are repeated *n* number of times, each time randomizing the alleles among individuals. The highest PBS values at each site or window is retained, providing an approximation of maximum genome-wide estimates under neutrality. The *P*-values are then defined by comparing the observed PBS estimates against quantiles of the simulated distribution. This randomization based approach is robust, applicable to all data, and in most cases more accurate than drawing thresholds directly from the sampling distribution. However, if user has knowledge about demography and recombination parameters, PaBScan provides an option to use simulated neutral data in ms format (Hudson 2002) as null distribution.

2. Usage

2.1 Downloading and compiling

PaBScan can be cloned from Github:

```
$ git clone https://github.com/thamala/PaBScan.git
```

Or downloaded as a ZIP package.

The program is then compiled using the following command (the header file `pabscan.h` needs to be in the same folder):

```
$ gcc pabscan.c -lm -o pabscan
```

2.2 Input data

PaBScan supports three input data formats: genotype likelihoods in Beagle format, genotype calls in VCF 4.1 format and genotype calls in native PaBScan format.

Example of a Beagle format input file, shown here for one individual (Ind0) and three markers (Chromosome 1, base pairs 24, 47 and 91):

```
marker     allele1    allele2    Ind0       Ind0       Ind0
1_24       3          0          0.999878   0.000122   0.000000
1_47       2          0          0.999992   0.000008   0.000000
1_91       2          0          1.000000   0.000000   0.000000
```

Beagle format files can be produced e.g. with ANGSD (Korneliussen *et al.* 2014): http://www.popgen.dk/angsd/index.php/Beagle_input

VCF 4.1 is the de facto genotype call format and it can be produced e.g. with GATK (McKenna *et al.* 2010) or Freebayes (Garrison & Marth 2012).

The flexibility of the VCF format can sometimes cause issues with algorithmic reading, so there is an option to use native PaBScan format as an input (shown here for six individuals and three markers):

```
CHR       BP        Ind0      Ind1      Ind2      Ind3      Ind4      Ind5
1         69270     11        10        00        00        11        11
1         69761     11        11        11        11        11        10
1         183598    10        00        00        00        00        10
```

A VCF file can also be transformed into a native format by using `-vcfp` input argument. This way the user has an option to check that the data has been read correctly, and the native format can be used as a faster input option in subsequent runs.

All input data files are assumed to be sorted according to chromosome and position. VCFtools (Danecek *et al.* 2011) or Picard (http://broadinstitute.github.io/picard/) can be used to sort VCF files.

2.3 Population lists

Three lists are required to define which individuals belong to which populations. Each list should be a plain text file (.txt) with UNIX line endings (LF) containing individual names corresponding to names found in data input files (one name per line). The focal populations are defined with lists `-pop1` and `-pop2` and the outgroup is defined with a list `-pop3.`

2.4 Simulated neutral data

Simulated neutral samples can be used in the outlier detection. Data is required to be in ms format (Hudson 2002), which may be produced also by other simulation programs, such as MSMS (Ewing & Hermisson 2010) and SLiM 2 (Haller & Messer 2017). Simulated data must have the same number of individuals as the observed data and they must be in the order defined by the populations lists. For example, if `-pop1` list has ten names, first ten individuals in the ms file are assumed to belong to population 1. The neutral PBS distributions can be printed to file by using `-msp` argument.

2.5 Output

PaBScan run without `-win` `-ms` or `-mc` options produces the simples output format, with chromosome, position in base pair and PBS estimates for the three populations:

```
Chromo     Position   PBS1       PBS2       PBS3
1          220623     0.319226   -0.120292  0.211402
1          445587     0.053947   -0.065290  0.070898
1          734592     -0.028352  0.001156   -0.043844
```

If `-win` (defined in number of SNPs) is used, start, middle and end positions of the sliding window, along with window length in number of base pairs, are also printed to file:

```
Chromo     Beginning  Middle     End        Length     PBS1       …
1          220623     477608     734592     513969     0.107262   …
1          825157     947273     1069389    244232     0.009635   …
1          1154260    1370954    1587647    433387     0.202560   …
```

And when `-ms` or `-mc` argument is present, *P*-values are also printed:

```
Chromo     Beginning  Middle     End        Length     PBS1       P1         …
1          220623     477608     734592     513969     0.107262   0.042000   …
1          825157     947273     1069389    244232     0.009635   0.460000   …
1          1154260    1370954    1587647    433387     0.202560   0.008000   …
```

2.6 Parameters

```
Required parameters:
           -likes [file]*        Genotype likelihoods in Beagle format
           -vcf [file]*          Genotype calls in VCF 4.1 format
           -vcfp [file]*         Same as '-vcf', except prints out a native PaBScan file (suffix .pabscan)
           -in [file]*           Genotype calls in native PaBScan format
           -pop1 [file]          List of individuals (one per line) from focal population one
           -pop2 [file]          List of individuals (one per line) from focal population two
           -pop3 [file]          List of individuals (one per line) from the outgroup population
           -out [string]         Name for the output file (suffix .pbs)

           *one of these is required

Optional parameters:
           -win [int]            SNP based window size (def 1)
           -step [int]           SNP based step size (def 1)
           -ms [file]            Simulated neutral data in ms format
           -msp [file]           Same as '-ms', except prints the null-distributions to file (suffix .nulldist)
           -perm [int]           Number of permutation cycles for Monte Carlo testing (cannot be used with '-ms')
           -div [int]            Divergence measure: [0] Hudsons's $F_{ST}$ [1] Weir & Cockerham's $F_{ST}$ [2] $d_{xy}$ (def 0)
           -min [int]            Minimum number of individuals required per population (def 1)
           -maf [double]         Minimum minor allele frequency required per site (def 0.01)

Usage example:
./pabscan -likes example.likes -pop1 list1.txt -pop2 list2.txt -pop3 list3.txt -out output -win 50 -step 51 -ms
example.ms -div 2 -min 5 -maf 0.05
```

3. References

Bhatia G, Patterson N, Sankararaman S, Price AL (2013) Estimating and interpreting F_{ST}: the impact of rare variants. *Genome research*, **23**, 1514–21.

Danecek P, Auton A, Abecasis G *et al.* (2011) The variant call format and VCFtools. *Bioinformatics*, **27**, 2156–2158.

Ewing G, Hermisson J (2010) MSMS: A coalescent simulation program including recombination, demographic structure and selection at a single locus. *Bioinformatics*, **26**, 2064–2065.

Garrison E, Marth G (2012) Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907 [q-bio.GN]*.

Haller BC, Messer PW (2017) SLiM 2: Flexible, Interactive Forward Genetic Simulations. *Molecular Biology and Evolution*, **34**, 230–240.

Hudson RR (2002) Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, **18**, 337–338.

Hudson RR, Slatkin M, Maddison WP (1992) Estimation of levels of gene flow from DNA sequence data. *Genetics*, **132**, 583–589.

Kim S, Lohmueller KE, Albrechtsen A *et al.* (2011) Estimation of allele frequency and association mapping using next-generation sequencing data. *BMC Bioinformatics*, **12**, 231.

Korneliussen TS, Albrechtsen A, Nielsen R *et al.* (2014) ANGSD: Analysis of Next Generation Sequencing Data. *BMC Bioinformatics*, **15**, 1471–2105.

McKenna A, Hanna M, Banks E *et al.* (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome research*, **20**, 1297–303.

Nei M (1987) *Molecular evolutionary genetics*. Columbia university press.

Reynolds J, Weir BS, Cockerham CC (1983) Estimation of the coancestry coefficient: Basis for a short-term genetic distance. *Genetics*, **105**, 767–779.

Weir BS, Cockerham CC (1984) Estimating *F*-statistics for the analysis of population structure. *Evolution*, **38**, 1358–1370.

Yi X, Liang Y, Huerta-Sanchez E *et al.* (2010) Sequencing of 50 Human Exomes Reveals Adaptation to High Altitude. *Science*, **329**, 75–78.