User Manual of *PCAdapt*
Version 1.6

Nicolas Duforet-Frebourg and Michael Blum
Université Joseph Fourier,
Centre National de la Recherche Scientifique,
Laboratoire TIMC-IMAG, Grenoble, France.

December 2014

1 Introduction

PCAdapt infers population structure and identifies outlier loci that are candidates for local adaptation. *PCAdapt* is based on a hierarchical factor model where population structure is captured using K latent factors. In order to identify candidates for local adaptation, the hierarchical factor model searches for loci that are atypically related to population structure as measured by the latent factors. Parameter inference is based on a Markov Chain Monte Carlo (MCMC) algorithm.

PCAdapt returns 1/a matrix of latent factors (also called scores) to capture population structure, 2/a matrix of factor loadings to measure the relationships between SNPs and latent factors, and 3/a list of Bayes factors. The SNPs with largest Bayes factors are the candidates for local adaptation.

We also provide an additional and faster version of *PCAdapt* that is suitable with very large datasets containing more than half a million of genetic markers. The fast version is based on Principal Component Analysis and does not use a MCMC algorithm. It also returns a matrix of scores and factor loadings. Instead of using Bayes factors to rank SNPs, outliers SNPs are found using a summary statistic based on factor loadings.

2 Algorithm and reference

The model and MCMC algorithm for the standard version based on a hierarchical factor model is described in the following paper:

Genome scans for detecting footprints of local adaptation using a Bayesian factor model, Duforet-Frebourg N, Bazin E and Blum MGB (2014) Molecular Biology and Evolution, doi: 10.1093/molbev/msu182.

The faster version, based on PCA, computes the sample covariance matrix, and computes the eigenvectors with the K largest eigenvalues.

3 Starters

3.1 Download

An archive containing the software can be downloaded at the following webpage:

`http://membres-timc.imag.fr/Michael.Blum/PCAdapt.html`

3.2 Windows OS

3.3 UNIX OS and MAC OS

Extraction and Compilation The archive of the program is provided with a `Makefile` for `UNIX` OS. Compilation proceeds as follows. First, you need to compile the local modified Lapack library [1].

```
MyMachine $> make lapack
```

Then, compile the program

```
MyMachine $> make
```

After compilation, if for some reasons, you want to clean the folder of all executables and binary files (including Lapack objects), just type

```
MyMachine $> make realclean
```

If you want to remove all executables and binary files but Lapack objects, just type

```
MyMachine $>make clean
```

Those two commands will also remove the executable `PCAdapt`. After compilation, you can run the program. You can run it without parameters, and a presentation screen will be displayed. Then the software is run as other usual software for LINUX.

With MAC OS, if you get the error message ‘Agreeing to the Xcode/iOS license requires admin privileges, please re-run as root via sudo.’ Running

```
MyMachine $> sudo xcrun cc
```

should bring up the cli version of the Xcode license agreement.

3.4 Windows OS

There is no *.exe* version for the moment.

4 Command line

Here is a complete list of the parameters of the program. When a parameter can be unspecified, it is explicitly mentioned. The basic command line to run the software is the following

```
MyMachine $> ./PCAdapt -i Genotypes -K number_of_factors
-o Output_file -s steps -b burnin -S scale
-I SVDinit_file -B Isingbeta -t transposeData
```

-i Genotypes The input file is the name, with the path, of the file containing the genotypes. Markers should be in lines, and individuals in columns. It is the same format as the *.geno* format of the software *Admixture* except that the 0/1/2 values of the genotype matrix are separated by white spaces. To convert input files to the right format, we provide conversion utilities (see subsection 5.1).

-K number_of_factors The number of latent factors K. ***Default value is 2 factors***

-o Output_files This parameter provides the name that is at the beginning of the names for output files. An extension is added to this pattern such as `.loadings` for the loading matrix. It may also contain a path to another directory. ***Default name is PCAdapt_output***.

-s steps The number of steps in the MCMC, including the burn-in. ***Default value is 400 steps.***

-b burnin The number of MCMC steps that are discarded. ***Default value is 200 steps*** to discard. Should be smaller than the parameter *steps*.

-I SVDinit_file (optional) Initialization of the factor matrix in the MCMC algorithm can be performed with PCA if the -I option is provided. If the argument -I is not provided, the MCMC use a random matrix as initial factor matrix. The name of the file containing the genotype matrix should be provided. It can be the same as the input file but it can also be a different file containing a subsample of the loci (e.g. a single chromosome).

-B Isingbeta This value is a real constant β for the Ising model. Larger values of the parameter favor outlier loci to cluster along the genome. ***Default value is 0*** meaning that the positions of the loci in the genome do not influence the algorithm.

-S scale data Set this parameter to 1 if you want to scale the markers, meaning they will all have a variance equal to 1. Set it to 0 to let the data unscaled. ***Default value is 1***.

-t transposeData If your genotypes are written with individuals in rows and SNPs in columns, no worries, transposition is operated in the software with the parameter `-t 1`. Can not be used with the faster version of *PCAdapt*.

-p proportion You can specify the proportion of top Bayes Factor SNPs you want to be written in the file `.topBF`. ***Default value is .01***.

1	0	0	0	1
9	9	1	0	0
1	0	2	0	0
0	0	0	2	0

Table 1: Example of an output file with 5 individuals and 4 SNPs.

5 Files

5.1 Input File

Genotypes The only input file is a matrix with one SNP per row, and individuals in columns. It is the same format as the *.geno* format of the software *Admixture* except that the 0/1/2 values of the genotype matrix are separated by white spaces. In case of a genotypes file with SNPs in columns, use the `-t 1` option in the command line (this can not be used with the faster version of *PCAdapt*). Missing values are coded with the value 9. Missing values are imputed in the algorithm.

Conversion from format .ped and .vcf Two programs are compiled with *PCAdapt*, *vcf2pcadapt* and *ped2pcadapt*. These two programs convert your data files from vcf (or ped) format to the format required to run PCAdapt. The command lines are

```
./vcf2pcadapt inputfile.vcf outputfile
./ped2pcadapt inputfile.ped outputfile
```

In both cases the output file is not needed. If omitted, it creates a file with the extension *.pcadapt* which is the genotype file you can directly use as input for *PCAdapt*. Others information files are written such as *.vcfsnp* and *.removed*.

5.2 Output Files

After a run, several output files are written. The main one is a file with the name specified with the `-o` option.

logBF	*logPO*	$P(Z=1\|O)$	$P(Z=2\|O)$	$P(Z=3\|O)$	$P(Z=4\|O)$
−0.264489	0.001975	0.321885	0.217664	0.380371	0.080080
0.022499	0.022499	0.009566	0.437138	0.548415	0.004882
...	...	...	...	...	...

Another file with extension `.stats` contains several statistics of the run. The *error* is the mean quarred error of the factorization model. This parameter is important to choose a value of K, as described in section 8. *pi* is the proportion of outlier in the genome. *c2* is the vector of all inflation coefficient in the K directions. *rho2* is a vector of estimated variance of each of the K directions. This vector is important to rank the Principal components with their variances.

```
  error :   0.163724
     pi :   0.006348
     c2 :   5.007744   6.139931   4.033926   9.372417
   rho2 :
0.137416   0.061379   0.051500   0.011266
```

The files with extensions `.scores` and `.loadings` contain the matrix of scores (the values of the latent factors) and of loadings. The dimension of the matrix of scores is $n \times K$ where n is the number of individuals and K is the dimension of the model specified by the user and the dimension of the matrix of loadings is $p \times K$ where p is the number of SNPs. The files with extensions `.BF` contains the list of SNPs with the largest Bayes Factor (top 1% by default), the corresponding Bayes factors, and the latent factor with which they are atypically related.

Note: save a logfile Note that if you wish to save a journal of the MCMC run, you can still redirect the flow in a log file typing:

```
MyMachine $> ./PCAdapt ... > myPCAdaptRun.log
```

6 Command line of *PCAdapt fast*

A fast version of *PCAdapt* is also implemented in the software. The faster version is based on Principal Component Analysis whereas the standard version is based on a Bayesian implementation of a hierarchical factor model. The fast version of the software is particularly well suited for datasets containing millions of SNPs. To run this version just add the keyword `fast` after `PCAdapt`.

```
 MyMachine $> ./PCAdapt fast -i Genotypes_1 Genotypes_2 ...
 -o Output_file -K number_of_factors -S scale
```

Several input files If your data are separated in several files, such as *ch1.geno ch2.geno...*, this version handles several inpufiles.

Output files Output files for this fast version are slightly different. The main output file is:

h'	axis	h	mAF	miss
44.458618	1	0.48682	0.276250	0.000000
44.521441	1	0.47852	0.342500	0.000000
15.391069	2	0.25864	0.196250	0.000000
...	...	...	...	...

There are two important statistics that are returned by the algorithm. The h' statistic (column 1) sums the squared loadings for the K PCs at each SNPs. The h statistic (column 3) corresponds to the communality statistic, as it is called in factor analysis, and measures the proportion of variance of a SNP that is accounted by the first K PCs. Although the h' statistic can provide less false discoveries than h in some settings where K is optimally chosen, we find that it is extremely sensitive to the choice of K. We therefore rather recommend using the h statistic in practice. Additionally, when individuals cluster into populations, the h statistic provides almost the same ranking as the common F_{ST} statistic (when using the populations to label the individuals). The third column gives the principal component (or latent factor) to which the SNPs is the most associated to. The two last columns indicate the minor allele frequencies and the number of missing values for each SNP.

Depending on the context, it might be a valid strategy to perform one genome scan for each PC. Population structure corresponding to each PC can be visualized using the *.scores* file. The correlations between the SNPs and the PCs (loadings) are then used to rank the SNPs for each PC (*loadings* file). This is a perfectly valid strategy if interested in biological adaptation that occurs along one of the PC axes only.

7 Example on a 4-population model

7.1 Run *PCAdapt*

In the *Example* folder of the archive *PCAdapt.tar.gz*, we provide a file to run *PCAdapt* on a first simple example. The data were simulated using the software *ms* [3], and *SimuPOP* [4]. We simulated a 4-population divergence model following the topology presented in [2]. 100 individuals are sampled in each of the 4 population and $6,000$ SNPs are available. Among the $6,000$ snps, 400 SNPs have been under selective pressures for the last generations, before the first split backward in time. To run *PCAdapt* with $K = 3$, type

```
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 3
-o results4pops_K3 -s 200 -b 100
```

7.2 Run *PCAdapt fast*

Another data set `Data4pops_2` is available in the *Example* folder. This file contain 4,000 additional SNPs for the 400 individuals sampled in `Data4pops_1`. None of these additional markers are involved in local adaptation. The fast version of the algorithm can be used with the two data files to learn population structure and the statistic computed for every SNP. To run the software with the two genotype datasets available in the *Example* folder, type

```
MyMachine $>./PCAdapt fast -i Example/Data4pops_1 Example/Data4pops_2 -K 3
```

8 A walkthrough example of how to use *PCAdapt*

8.1 Choice of K

We consider the example already described in subsection 7.1. To run *PCAdapt* for several values of K, type

```
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 1
-o results4pops_K1 -s 200 -b 100 -I Example/Data4pops_1
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 2
-o results4pops_K2 -s 200 -b 100 -I Example/Data4pops_1
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 3
-o results4pops_K3 -s 200 -b 100 -I Example/Data4pops_1
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 4
-o results4pops_K4 -s 200 -b 100 -I Example/Data4pops_1
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 5
-o results4pops_K5 -s 200 -b 100 -I Example/Data4pops_1
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 6
-o results4pops_K6 -s 200 -b 100 -I Example/Data4pops_1
```

Investigating the mean squared error as a function of K, we find that $K = 3$ is the most adequate value (figure 1, see [2] for a discussion about the choice of K). The figure can be obtain by running the R script `get_errors.R`.

```
MyMachine $> R
```

and in the **R** command line, type

```
> source("Rscripts/get_errors.R").
```

8.2 Consistency of the factors over the runs

Because of variations in the MCMC algorithm, the latent factors can rotate among different runs. We recommend to check if is rotation of the factors between runs.

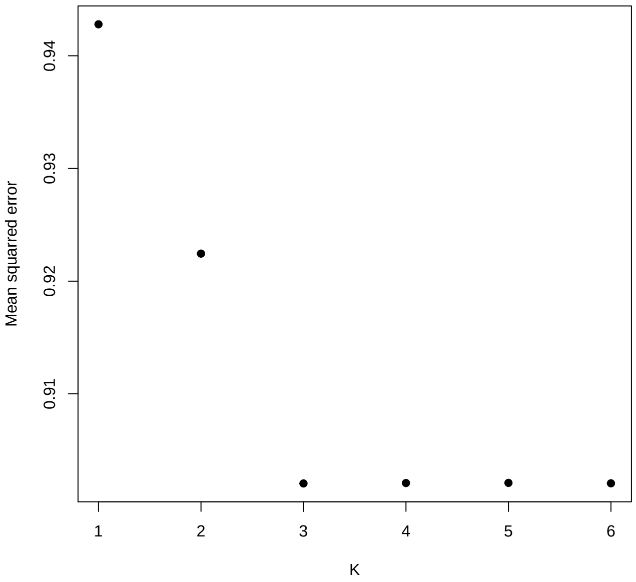

Figure 1: Mean squared error explained by the number of latent factors.

Once the value of K is chosen (here $K = 3$), different runs of MCMC can be performed by typing

```
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 3
-o results4pops_K3_1 -s 200 -b 100
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 3
-o results4pops_K3_2 -s 200 -b 100
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 3
-o results4pops_K3_3 -s 200 -b 100
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 3
-o results4pops_K3_4 -s 200 -b 100
MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 3
-o results4pops_K3_5 -s 200 -b 100
```

Then compute a squared correlation coefficient to ascertain that the factors contain the same genetic structure, open a R session:

```
MyMachine $> R
```

and in the **R** command line, type

```
> source("Rscripts/check_rotation.R").
```

The R^2 values calculated in the object `cor_runs` indicate if for two runs the learned directions are the same. If the R^2 values are below a certain threshold for a certain run, you can ignore the run. If there is no consistency between most of the runs, a way to get consistent results is to use the SVD initialization with the option "-I".

```
   MyMachine $> ./PCAdapt -i Example/Data4pops_1 -K 3 -S 0
-o results4pops_K3 -s 200 -b 100 -I Example/Data4pops_1
```

8.3 Graphical display of the results

To display population structure as encoded by the matrix of latent factors or scores and to display a Manhattan plot using the Bayes factors, run **R**

```
MyMachine $> R
```

and in the **R** command line, type

```
> source("Rscripts/Display.R").
```

The script displays the latent factors (population structure) and the Bayes factors along the genome (Manhattan plot).

9 Running *PCAdapt fast* with large datasets

PCAdapt can handle very large datasets, with millions of markers using the provided fast version. Instead of returning Bayes factors, it returns a statistic called the communality h to rank the SNPs. In case of discrete populations, it provides the same ranking as the standard F_{ST} statistic.

```
   MyMachine $> ./PCAdapt fast -i Example/Data4pops_1 -K 3
-o results4pops_K3_fast -p .02
```

This version can handle several input files, such as files containing markers from different chromosomes

```
   MyMachine $> ./PCAdapt fast -K 3 -o results4pops_K3_fast
-i Example/Data4pops_1 Example/Data4pops_2 -p .02
```

The `-p` option indicates which proportion of the top scoring markers (as function of h') must be written in separate output files with the extension `.top`. Here we return the top 2% of the SNPs with the highest values of the h' statistic. This option should not be used since we do not recommend to use the h' statistic anymore.

This proportion is calculated on the entire data set (whatever the number of input files). To observe *PCAdapt fast* results, run **R**

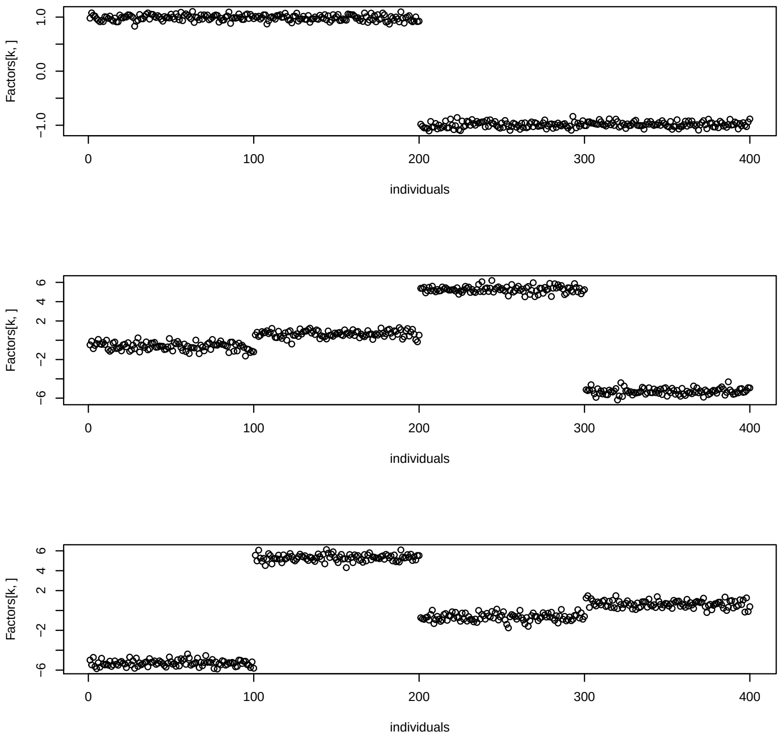

Figure 2: Factors

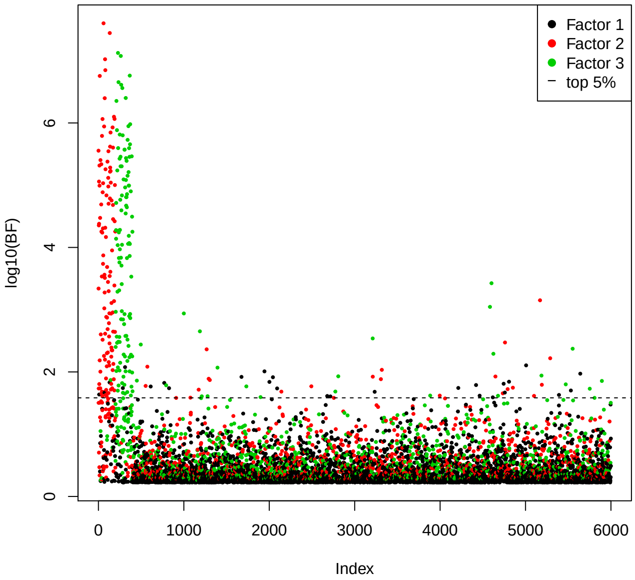

Figure 3: Bayes Factors

```
MyMachine $> R
```

and in the **R** command line, type

```
> source("Rscripts/Display_fast.R").
```

The script displays the latent factors (population structure) and the summary statistic h along the genome (Manhattan plot).

10 A walkthrough example of how to use *PCAdapt fast*

10.1 choice of K

We consider the example dataset contained in files *Example/Data4pops_ 1* and *Example/Data4pops_ 2*, reprensenting for example chromosome 1 and 2 of the dataset. Run *PCAdapt fast* for a large value of K.

```
MyMachine $> ./PCAdapt fast -i Example/Data4pops_1
MyMachine $> Example/Data4pops_2 -K 10 -o results4pops_K10
```

run **R**

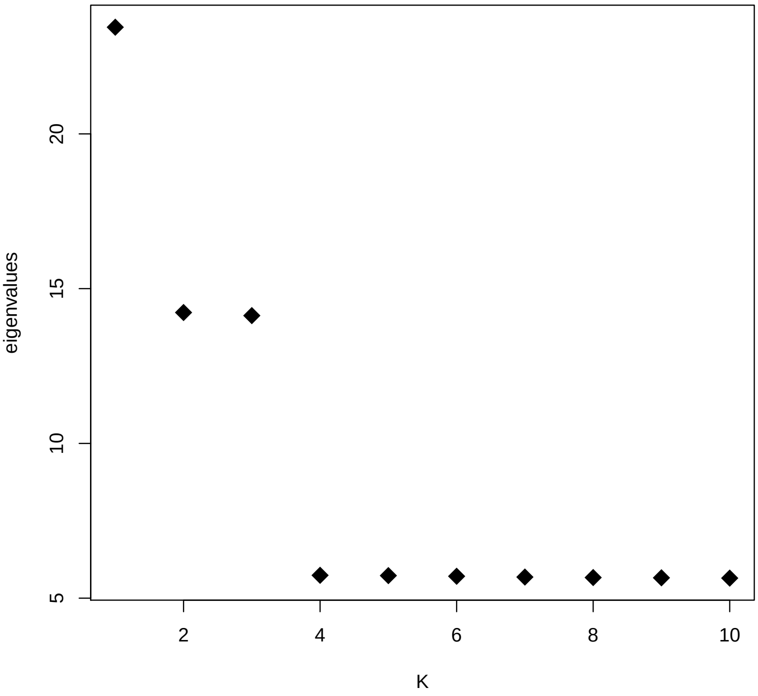

Figure 4: Screeplot

```
MyMachine $> R
```

and in the **R** command line, type

```
> source("Rscripts/Display_Sigma_K.R").
```

In figure 4, we display the scree plot of principal component analysis, which shows the decay of the eigenvalues of the covariance matrix (or of the correlation matrix, with the $-S1$ option) as a function of K. The choice of K here is $K = 3$, because a plateau is reached at $K = 4$. Be careful, when using the mean squarred error (MSE) of the Bayesian version (Figure 1), we recommend to use the first value of K where the plateau of MSE is reached, whereas with the *fast* version, we recommend to stop just before the plateau of eigenvalues. The difference comes from the fact that the MSE would be of $1-\sum_{i=1}^{K}\lambda_i^2/p$ (for the scaled version of PCA, with the `-S 1` option) when using the *fast* version of *PCAdapt*.

10.2 Graphical display of the results

To observe *PCAdapt fast* results, run **R**

```
MyMachine $> R
```

and in the **R** command line, type

```
> source("Rscripts/Display_fast.R").
```

The script displays the scores of the PCs (population structure) and the summary statistic h for each marker of the genome (Manhattan plot). Compared to the Bayesian version, the fast version does not return Bayes factors but a statistic h to rank the SNPs.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[2] Blum M.G.B Duforet-Frebourg N., Bazin E. Genome scans for detecting footprints of local adaptation using a bayesian factor model. *Molecular Biology and Evolution*, 2014.

[3] R.R. Hudson. Generating samples under a wright–fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, 2002.

[4] Bo Peng and Marek Kimmel. simupop: a forward-time population genetics simulation environment. *Bioinformatics*, 21(18):3686–3687, 2005.