localScore: an R package to highlight optimal and suboptimal segment in a sequence with associated p-values computation

David Robelin, Sébastien Déjean, Sabine Mercier

`localScore`: an `R` package to highlight optimal and suboptimal segment in a sequence with associated p-values computation

David Robelin[1], Sébastien Déjean[2], and Sabine Mercier[3]

https://doi.org/10.5802/fake.doi

Abstract

Highlighting atypical segments of a sequence is an important goal in very diverse domains. In the case where no prior information on the length of the segment to be highlighted is known, Karlin and Altschul defined in 1990 the local score for biological sequence analysis, and an asymptotic approximation of its distribution is proposed in 1992. There exist now many other theoretical results to establish the local score p-value in different contexts.

We developed an `R` package gathering these results for a sequence modeled by independent and identically distributed variables. It allows to compute the local score, the suboptimal scores, their position, and proposes to establish the local score p-value using the different theoretical methods available so far. An automatic analysis is also proposed to perform the most appropriate method according to the analyzed sequence.

We present here the package and different examples of application. Comparisons with other tools used depending on the context of application are also given. The `localScore` package is available on the Comprehensive R Archive Network. It is distributed under the GPL-2 licence for the core program (and various licenses for embedded Eigen library)

Keywords: statistical significance, local score, sequence analysis, atypical segment detection, Lindley-CUSUM process

[1]GenPhySE, UMR1388, INRAE Toulouse, 24, chemin de Borde-Rouge, Auzeville Tolosane, France, [2]Institut de Mathématiques de Toulouse, UMR 5219 Université de Toulouse et CNRS, Université Toulouse III Paul Sabatier, 118 route de Narbonne, Toulouse, France, [3]Institut de Mathématiques de Toulouse, UMR 5219 Université de Toulouse et CNRS, Université Toulouse II Jean Jaurès

Correspondence
machin@example.edu

Introduction

Highlighting atypical periods or segments in sequences is an issue of interest in many fields, such as Bioinformatics and Genomics, Biosurveillance, Ecology and Environmental Sciences, Epidemiology and Health Sciences, Finance, Reliability and Quality Control, Telecommunication Sciences, and many others... Karlin and Altschul, 1990 defined the local score statistic to analyze biological sequences: it corresponds to the maximum cumulative value of a given property over every possible segments in a sequence, considering segments of any position and any length (see Equation (1)). The calculation of the statistical significance of the local score is crucial in order to distinguish atypical values from ones that could have appeared by chance. Karlin and Dembo, 1992 proposed asymptotic approximations of the distribution of the local score when the length of the sequence is growing to the infinity. A generalization of this approximation for the sequence comparison case is developed in BLAST Software[1], but to our knowledge no development have been done for a single sequence analysis case. At the present time, results exist that consider independent or dependent models on the sequence. Those results include: improvements of the approximations of Karlin *et al.* (see Cellier et al., 2003 for the independent model and Grusea and Mercier, 2020 for the Markov model); exact methods (see Mercier and Daudin, 2001 for the independent model and Hassenforder and Mercier, 2007 for the Markov model); a result on the distribution for the pair of the local score value and the length of the segment that realizes the local score value (see Chabriac et al., 2014; Lagnoux et al., 2017). We developed the package `localScore` (Simon et al., 2023) for the software `R` (R Core Team, 2024). In this first version of the package, we focus on the different ways to establish the statistical distribution of the local score in a sequence modeled by independent and identically distributed (I.I.D.) random variables, and address a first result in the Markovian case.

The remainder of the article contains a brief presentation of the main theoretical backgrounds implemented in the `localScore` package. Then the package is described in Section through a standard workflow to follow and a description of the main functions. Section presents examples of using `localScore` in four different domains: biological sequence analysis and a comparison with a sliding window statistics; a signal detection context and a comparison with control charts; epidemiology and a comparison with scan statistics; a genomic sequence analysis.

Theoretical background

Let us consider a sequence as a succession of components that belong to a finite set $\mathcal{A}$. It can be for example a DNA sequences with $\mathcal{A} = \{A,\ C,\ G,\ T\}$. Let us define a score scheme or a score scale as a function s that assigns a real number to any letter of $\mathcal{A}$. The score can, for example, quantify a physico-chemical property. See the web site of Protscale[2] for an illustration of different score scales in biological sequence analysis context. Let $\mathbb{A} = (A_i)_{1 \leqslant i \leqslant n}$ be a sequence, and let us denote $X_i := s(A_i)$, for $i \geqslant 1$, the scoring sequence associated to the sequence $\mathbb{A}$ based on the score function s. Examples of scoring functions are presented in Section .

[1] https://blast.ncbi.nlm.nih.gov/
[2] https://web.expasy.org/protscale

With $X_0 := 0$, the local score M_n of one sequence $\mathbb{X}$ of length n is defined by

$$M_n := \max_{0 \leqslant i \leqslant j \leqslant n} \sum_{k=i}^{j} X_k \ . \tag{1}$$

In Mercier and Daudin, 2001 the authors proved that the local score can also be defined as: $M_n := \max_{0 \leqslant i \leqslant n} U_i$ with $U_0 := 0$ and $U_{i+1} := \max(U_i + X_{i+1}, 0)$ the Lindley, or CUSUM process, associated to the sequence $(X_i)_{1 \leqslant i \leqslant n}$. The Lindley process defines non negative excursions and the height of the highest one is equal to the local score. The other excursions are called the sub optimal segments.

The local score approach avoids the choice of a segment length when no prior information on it is available. Let us present below the two main kinds of results, approximation and the exact method, when the random variables $(A_i)_{1 \leqslant i \leqslant n}$ are independent and identically distributed (I.I.D.) and so are the $(X_i)_{1 \leqslant i \leqslant n}$.

Karlin and Dembo approximation

The asymptotic approximation of Karlin and Altschul, 1990 and Karlin and Dembo, 1992 corresponds to an asymptotic result converging to a Gumbel distribution when the length of the sequence n tends to infinity. This result stands on the two following hypotheses: The average score must be non positive, $\mathbb{E}[X] < 0$, and a non negative score must be possible, $\mathbb{P}(X > 0) > 0$. We have

$$\lim_{n \to +\infty} P\left(M_n \leqslant \frac{\ln n}{\lambda} + x\right) = e^{-K^* e^{-\lambda x}} \tag{2}$$

where λ and K^* depend on the score distribution. The parameter λ corresponds to the single root of a polynomial of degree equal to the amplitude of the scores (maximum score minus minimum score) and checking $\mathbb{E}[\exp(\lambda X)] = 1$. The existence of λ is ensured by the assumption $\mathbb{E}[X] < 0$. The set of other roots is also used for the calculation of K^* notably by means of a square matrix called Vandermonde comprising at each line a geometric progression associated with one of the roots of the polynomial. This gives the following approximation for an observed local score a

$$P(M_n \leqslant a) \approx e^{-K^* n e^{-\lambda a}} \ . \tag{3}$$

The computation of the approximation given in (3) is very accurate for sequence length larger than thousands and very fast to obtain but must be avoided for sequence shorter than a hundred components.

Karlin by Monte Carlo

The Karlin and Dembo approximation in (2) calculates the value of two parameters λ and K^* in function of the values and distribution of the scores. Here we propose to estimate them by a Monte Carlo approach. This method is useful in the case of a too long score sequence to perform a direct and efficient Monte Carlo of the local score distribution as it does not need to simulate

full length sequences. Formula (3) can be linearized in λ and K^* using logarithms as long as n is large enough. That leads to the following formula:

$$\ln\{-\ln\{P(M_n \leqslant a)\}\} \approx \ln K^* - \lambda a + \ln n \,. \tag{4}$$

Given the previous formula, the Karlin by Monte Carlo procedure consists in:

(1) Choosing a sequence length n_{sim} for the simulation big enough to have a satisfying Karlin and Dembo approximation and small enough to be computed with reasonable resources.
(2) Simulating sequences of size n_{sim}.
(3) Calculating the local score of each sequence in order to derive empirical distribution function of the local score for sequence of size n_{sim}.
(4) Deriving estimation of λ and K^* by a linear regression on the empirical distribution function using Formula (4), i.e., $\hat{\lambda} = -\hat{b}$ and $\hat{K^*} = \exp(\hat{a})/n_{sim}$ where $\hat{a}$ and $\hat{b}$ are respectively the slope and the intercept of the regression.
(5) Using Karlin and Dembo approximation to calculate the p-value of the local score observed on the full sequence of size n.

Daudin

For a an observed local score value, the exact method in the I.I.D. case is based on an appropriate stopped process constructed to be a Markov chain and taking its values in $\{0, \dots, a\}$. Let us denote $P = (P_{ij})_{0 \leqslant i,j \leqslant a}$ its corresponding transition matrix. Mercier and Daudin, 2001 proved that

$$(\forall a \geqslant 0) \quad \mathbb{P}(M_n \geqslant a) = (P^n)_{(0,a)} \,. \tag{5}$$

There is no restriction on the sign of the average score for the use of the exact method. This method is accurate and very fast for n up to several thousands but must be avoided for very long sequence of order a million because it could be too time and space consuming. As there exists a limit to the exponentiation of P for n tending to infinity, it is not necessary to use the correct value of n for sequence of length larger than a hundred of thousands, to obtain an accurate value, but a smaller value could be used.

Improved approximation of Karlin et al. in the I.I.D. case

An improved approximation of the one proposed in Karlin and Dembo, 1992 is proposed in Cellier et al., 2003. As with the Karlin *et al.* method, it is necessary to calculate the roots of the same polynomial which are then used in several steps in order to calculate the additive correcting terms to improve the approximation of Karlin *et al.* We have for large a values

$$P(M_n \leqslant a) \approx (1 - \sum_{i=1} K_i R_i^a)^{\frac{n}{\mu}+1} \tag{6}$$

with $(R_i)_i$ the roots of module strictly less than 1 of a polynomial directly defined with the score distribution. The degree of this polynomial is equal to the range of the possible scores. Based on the two hypothesis used in the work of Karlin and Dembo, there exits a unique positive real root with module less than 1, $e^{-\lambda}$, with λ defined in Equation (2). The parameters K_i and μ are also derived from the score distribution and the computation based on the Vandermonde

matrix and some equation system resolutions. The improved approximation in (6) is accurate and fast for values of n from several hundreds, but must be avoided for sequence length lower than one hundred.

All the above theoretical results must be considered complementary for practical application depending on the score scheme, with its range, the sign of the average score, and the length of the sequence to be analyzed.

Software features and contents

Workflow

A tentative workflow using `localScore` could be:

(1) Transform the component of a given sequence set into score sequences through a given score function.
(2) Learn the distribution of the scores on the score sequences.
(3) Compute the local score of each sequence.
(4) Compute the corresponding p-values using the automatic method for the computed local score value, the corresponding sequence length and the global score distribution.

Main functions

Following the workflow presented above, here are the main functions that can be used in each step.

To get a score sequence: The transformation of a component sequence, as a DNA one, into a score sequence can be done with the `CharSequence2ScoreSequence` function. Integer or real scores can be considered.

To learn a distribution: Learning distribution of the components of the sequences or the given scores can be performed using several functions. For instance, the empirical distribution from one numerical sequence or a list of sequences is built by `scoreSequences2probabilityVector`.

To compute the local score: The function `localScoreC` (*respectively* `localScoreC_double`) calculates the local score for a sequence of integer (*resp.* real) scores. It provides the local score and all suboptimal segments with associated scores. Functions `suboptimalSegment` or `Lindley` can be used to obtain the others localizations of the different realizations of the local score.

To compute the corresponding p-values: Then, the following functions propose different methods to compute p-values associated to the local score of a sequence:

- `karlin`: The Karlin *et al*'s approximation (see (3)). This method needs a non positive average score, $\mathbb{E}[X] < 0$, and integer scores, and is more adapted for long sequences with length larger than a few thousand components, depending on the expectation of the score distribution.
- `mcc`: An improved approximation of the previous one presented in Cellier et al., 2003. This method also needs a non positive average score, $\mathbb{E}[X] < 0$, and integer scores, and is more adapted for sequences with length from a few hundreds components, depending on the expectation of the score distribution.
- `daudin`: An exact method for integer scores is also incorporated and can be used whatever the sign of the expected score (see (5)). This method is computationally adapted for not too long sequences, but several thousands of components can be easily handled.

Table 1 – Adequate methods to compute the local score p-value depending on the average score value $\mathbb{E}[X]$ and the sequence length n order ; with E : `daudin()` ; MCC : `mcc()` ; K : `karlin()` ; MC : `monteCarlo()` ; MC-K : `karlinMonteCarlo()`.

n	< 100	$10^2 \leqslant \cdot < 10^3$	$10^3 \leqslant \cdot < 10^4$	$\geqslant 10^4$
$\mathbb{E}[X] < 0$	E ; MC	E ; MCC ; MC	E ; MCC ; MC	MCC ; K ; MC ; MC-K
$\mathbb{E}[X] \geqslant 0$	E ; MC	E ; MC	E	

The implementation is based on the exponentiation of a square matrix of size a, with a a given local score value.

- `monteCarlo`: A classical Monte Carlo method
- `karlinMonteCarlo` and `karlinMonteCarlo_double`: A mix between the Karlin *et al.*'s and the Monte Carlo method. It allows an approximated distribution with a lower time computation than the empirical Monte Carlo method, for very long sequences. This mixed method also needs $\mathbb{E}[X] < 0$.

We also developed the function `automatic_analysis` for users with less experience. This function, as its name indicates, automatically picks the adequate p-value method for the user's input according to the configuration described in Table 1. The function calculates the p-value based on the length of each of the sequences given as input. It can either use an empirical score distribution based on the input or a distribution provided by the user. By setting the `method_limit`, the user can also decides up to what sequence length the computation-intensive methods (`daudin`, `exact_mc`) should be used to calculate the p-value.

Inputs / outputs

Inputs. When starting the workflow, the first input is a sequence. It can be imported in `R` from an ASCII file using the standard reading functions such as `read.table` and related functions. For users interested in analyzing biological sequences composed of nucleotides or amino acids, the package can also handle FASTA files as inputs. In FASTA files, every sequence is preceded by a title (marked by a ">") and a line break. One sequence takes one line, followed by a line break and a line only containing a tab.

Furthermore, if no sequences are passed to the `automatic_analysis` function, it let the user pick a FASTA file. In this case, and if the user hasn't provided any score system (as it can be done by passing a named list with the appropriate scores for each character), the second file dialog pops up. The latter allows to choose a file containing the score, and if the user provides an extra column for the probabilities, they are used, too - see Section File Formats in the vignette for details.

Score files can also be imported in a standard way from an ASCII file. Such a file must contain a header and each row contains a letter and its score. Optionally, a probability for each score can also be provided.

Numerical outputs. The main numerical output is given by the `localScoreC` function. It contains a list with the following attributes:

- The local score value and the begin and end index of the segment realizing this optimal score.
- All the local maxima of the Lindley process (non negative excursion) and their begin and end index.

- The record times of the Lindley process but only the ones corresponding to the begin index of non negative excursions.

Every method calculating *p*-values only provide the value obtained.

Graphical outputs. Graphical outputs can be optionally displayed by the `monteCarlo` and the `karlinMonteCarlo` functions. They represent the distribution of all local scores simulated and the cumulative distribution.

Example data

Some data we propose to analyze in Section are already embedded in the package for illustration purpose. `Seq1093` is a real biological sequence with 1093 characters referring to Q60519 queries in UniProt Data base[1]. `SeqListSCOPe` contains 285 protein sequences with length from 31 to 404. They are referred as `CF_scop2dom_20140205aa` in the Structural Classification Of Proteins database (SCOP)[2]. `SJSyndrome.data` corresponds to a dataset of 824 lines, each describing a Stevens-Johnson syndrome appearance described by 15 covariates including Case ID, Initial FDA Received Date, days since last fda. The third column corresponds to the number of days between two adverse events. `Aeso` consists of individual dates of birth over 35 cases of the birth defects oesophageal and tracheo-oesophagean fistula observed in a hospital in Birmingham.

Illustrations

We illustrate the use of the `localScore` package on four examples in different fields. First, one of the biological sequences embedded in the package is used as a toy example to show a basic use of the package. In the same vein, we illustrate how to deal simultaneously with a set of sequences. Then, we analyze two medical data sets to show how local score can be used to detect eventual shift in sequential observations. Subsection deals with the study of a chromosome to associate genomic regions with phenotype differentiation. We also present, for each case, results of other methods.

Biological sequences

We first describe how to analyze one single sequence then we show how to deal with a set of several sequences at once.

One single sequence. Several sequences are already embedded in the package. Let us use the `Seq1093` object, corresponding to the protein Q60519 SEM5B_MOUSE[3]. With 1093 characters, we consider it as a sequence for which quite all the possible proposed methods can be used to establish the statistical significance (see Table 1).

```
R> library(localScore)
R> data(Seq1093)
R> MySeq <- Seq1093
R> nchar(MySeq)
```

[1] https://www.uniprot.org
[2] https://scop.mrc-lmb.cam.ac.uk/
[3] https://www.uniprot.org/uniprot/Q60519

The function `CharSequence2ScoreSequence` converts the character sequence into a score sequence using the `HydroScore` object providing the correspondence between a letter and its score according to Kyte & Doolittle hydrophobic score scale (Kyte and Doolittle, 1982).

```
R> data(HydroScore)
R> SeqScore <- CharSequence2ScoreSequence(MySeq, HydroScore)
```

Then the local score computation can be performed.

```
R> ResLocalScoreMySeq <- localScoreC(SeqScore)
R> ResLocalScoreMySeq
$localScore
value begin   end
   65   956  1001

$suboptimalSegmentScores
      value begin  end
 [1,]    40     1   20
 [2,]    10    71   73
 [3,]    23    80   99
 [4,]     3   114  114
 [5,]     3   124  124
 [6,]     4   128  128
 [7,]    23   130  150
 [8,]    16   181  195
 [9,]     2   217  217
[10,]     4   224  224
[...]
[70,]     2  1054 1054
[71,]     3  1056 1056
[72,]     4  1059 1059
[73,]     4  1064 1064
[74,]     4  1074 1074
[75,]     3  1079 1079
[76,]     2  1083 1083
[77,]     6  1089 1090

$RecordTime
 [1]    1   71   80  114  124  128  130  181  217  224  229
[12]  294  301  304  312  379  384  387  390  411  413  416
[23]  523  533  548  554  563  566  574  588  633  637  644
[34]  654  661  680  691  697  711  715  725  740  742  746
[45]  748  754  756  766  773  778  794  799  807  811  813
[56]  822  832  844  848  864  879  883  890  924  931  934
[67]  951  956 1048 1054 1056 1059 1064 1074 1079 1083 1089
```

We retrieve only the local score value for further use when calculating the p-value.

```
R> LocalScoreMySeq <- ResLocalScoreMySeq$localScore[1]
```

The function `scoreSequences2probabilityVector` builds an empirical distribution from the sequence.

```
R> ProbDistribution <- scoreSequences2probabilityVector(SeqScore)
R> round(ProbDistribution, 3)
    -5    -4    -3    -2    -1     0     1     2     3     4     5
0.074 0.203 0.020 0.075 0.212 0.078 0.000 0.071 0.094 0.144 0.028
```

The exact method (see (5)) can then be used to compute the *p*-value.

```
R> ResDaudin <- daudin(localScore = LocalScoreMySeq,
+    sequence_length = length(SeqScore),
+    score_probabilities = ProbDistribution,
+    sequence_min = min(SeqScore),
+    sequence_max = max(SeqScore))
R> ResDaudin
[1] 0.072
```

The approximate method of Karlin *et al.* (see (3)) can be performed equivalently with the `karlin` function.

```
R> ResKarlin <- karlin(localScore = LocalScoreMySeq,
+    sequence_length = length(SeqScore),
+    score_probabilities = ProbDistribution,
+    sequence_min = min(SeqScore),
+    sequence_max = max(SeqScore))
R> ResKarlin
[1] 0.076
```

The two *p*-values are rather close (0.072 for the exact method, 0.076 for the approximate one).

In comparison, here are the results obtained with ProtScale Expasy web tool on the same sequence. ProtScale computes and represents the profile on a selected protein produced by any amino acid scale and accumulating the score values over a sliding window of a chosen size. Note that the possible window size are restricted to odd values from 3 to 21. We used the hydropathicity scale proposed by Kyte and Doolittle, 1982. We chose a size equal to 21 which is the closest value to the length of the optimal segment given by the local score approach without any prior information on it. The results are presented in Figure 1. We can observe one main peak and the numerical output (not shown) gives us a window value equal to 2.195 and a center index equal to 989 (begin index 979; end index 999). This segment corresponds to the one highlighted by the local score but with a length equal to 45 with begin index 956 and end index 1001. The local score *p*-value allows us to say that this region is not statistically significant. For a window size equal 9 corresponding to the one given by default, we can observe several picks with a similar value before the one we discuss previously. We have also represented the corresponding Lindley process using the {`lindley` function.

```
R> LindleySeqScore <- lindley(SeqScore)
R> plot(LindleySeqScore, type="l")
```

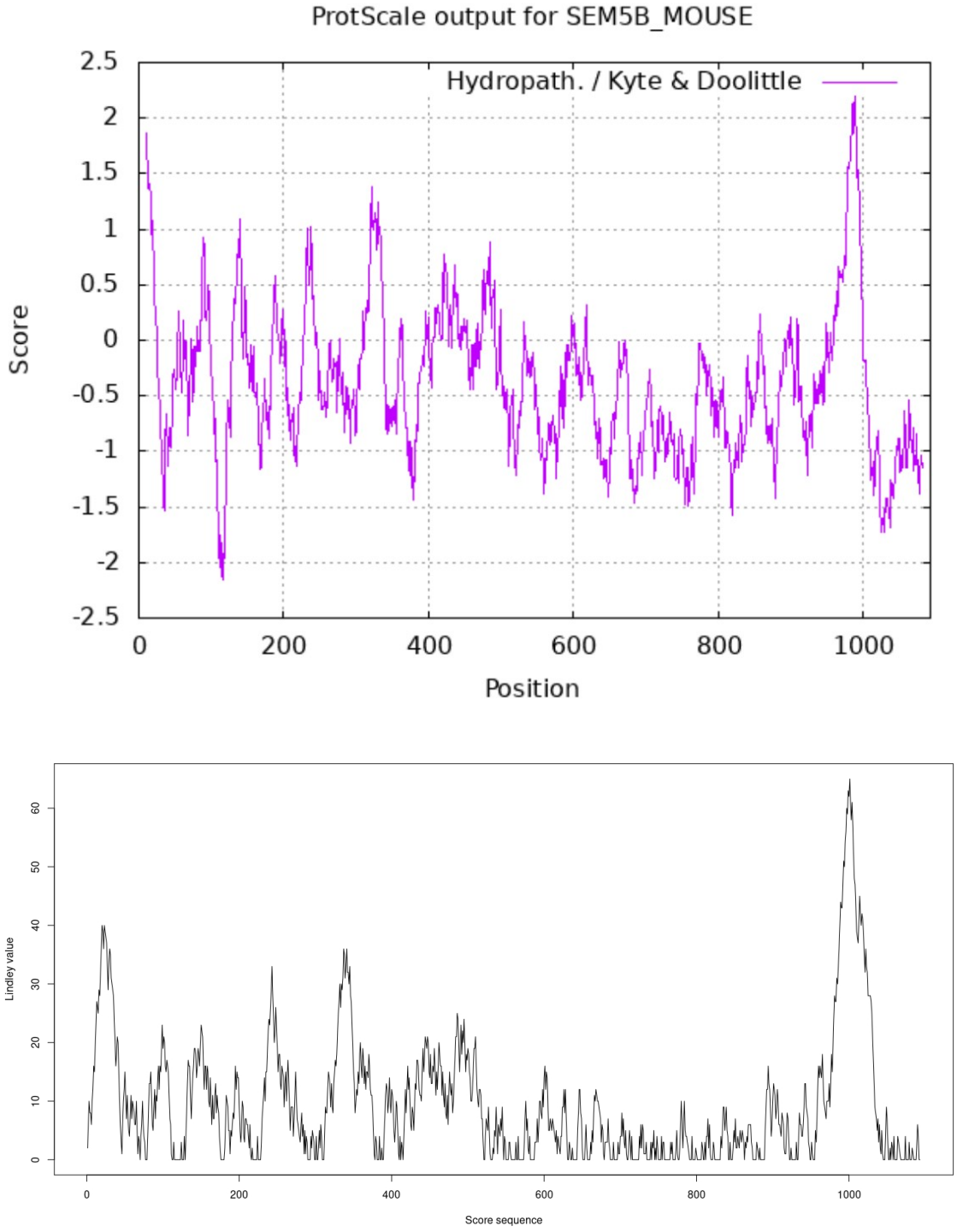

Figure 1 – Top: Graphical output of the results provided by the Expasy ProtScale web tool for the corresponding sequence Q60519, the Kyte and Doolittle scale and a window size equal to 21. Bottom: Lindley process calculated with the `localScore` package.

A set of sequences. The data consists in a list of 285 character strings with their entry codes as names extracted from the Structural Classification Of Proteins database (SCOP)[2]. More precisely this data contain the 285 protein sequences of the data called "CF_scop2dom_20140205aa" with sequence length from 31 to 404.

[2]https://scop.mrc-lmb.cam.ac.uk/

This sequence is a part of the package and can be loaded and briefly explored with:

```
R> data(SeqListSCOPe)
R> SeqListSCOPe[1]
P50456
"ARDVIQVVIDHNVGAGVITDGHLLHAGSSSLVEIGHTQVDPYGKRCYCGNHGCLETIAS
VDSILELAQLRLNQSMSSMLHGQPLTVDSLCQAALRGDLLAKDIITGVGAHVGRILAIMV
NLFNPQKILIGSPLSKAADILFPVISDSIRQQALPAYSQHISVEST"
R> nchar(SeqListSCOPe[1])
P50456
   165
R> summary(sapply(SeqListSCOPe, nchar))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   31.0    78.0   102.0   121.7   141.0   404.0
```

The sequence lengths varie from 31 to 404.

The function `CharSequence2ScoreSequence` transforms the protein sequence into a score sequence using the `HydroScore` object. The score corresponding to each amino acid can be displayed as:

```
R> data(HydroScore)
R> unlist(HydroScore)
```

and the conversion is done as follows:

```
R> MySeqScoreList <- lapply(SeqListSCOPe,
+     FUN = CharSequence2ScoreSequence, HydroScore)
```

Then we use `automatic_analysis` function to perform the most appropriate method to compute the *p*-value of the local score of each sequence.

```
R> ResAutoAnalysis <- automatic_analysis(sequences = MySeqScoreList,
+     model='iid')
```

The results can then be investigated.

```
R> ResAutoAnalysis[[1]]
$`p-value`
[1] 0.06389172

$`method applied`
[1] "Exact Method Daudin et al"

$localScore
$localScore$localScore
value begin   end
   67     4   144

$localScore$suboptimalSegmentScores
      value begin end
[1,]     2     1   1
```

```
[2,]    67     4 144

$localScore$RecordTime
[1] 1 4
```

The first sequence of the list has a local score value equal to 62 and the segment that realizes this maximum begins at index 4 and finishes at index 144. Its p-value equals 6.39%.

We can easily extract the first 10 p-values, the 5 smallest p-values, the significant sequences and their local score values.

```
R> sapply(ResAutoAnalysis, function(x){x$`p-value`})[1:10]
P50456 P14859 P10037 Q13619 P22262 P20823 P07014 Q9X399 Q0SB06 Q9I641
 0.064  0.973  0.875  0.896  0.451  0.967  0.749  0.681  0.994  0.512
R> sort(sapply(ResAutoAnalysis, function(x){x$`p-value`}))[1:5]
      Q5SMG8       P0A334       Q2W6R1       O27564       P12282
9.485100e-07 3.442818e-04 4.406208e-04 4.548065e-04 6.167591e-02
R> which(sapply(ResAutoAnalysis, function(x){x$`p-value`}) < 0.05)
Q2W6R1 O27564 P0A334 Q5SMG8
    14     90    150    192
```

The local score of every sequence can be displayed as a boxplot or an histogram (Fig. 2):

```
R> SeqLocalScore <- sapply(ResAutoAnalysis,
+    function(x){x$localScore$localScore[1]})
R> boxplot(SeqLocalScore, horizontal = TRUE)
R> hist(SeqLocalScore)
```

The methods used to compute the p-values can be retrieved with

```
R> table(sapply(ResAutoAnalysis, function(x){x$`method`}))
Exact Method Daudin et al
                       206
```

The maximum sequence length equals 404 so it is here normal that the exact method is used for all the 606 sequences of the data base. The score distribution that has been used to compute the p-value for every local scores is the empirical one estimated on the whole data set. It can be exhibited using

```
R> scoreSequences2probabilityVector(MySeqScoreList)
   -5    -4    -3    -2    -1     0     1     2     3     4     5
0.055 0.264 0.022 0.041 0.148 0.072 0.000 0.105 0.052 0.175 0.067
```

Medical data

In other domains, as Telecommunication Sciences or Quality Control to name only those, where the goal is to highlight a change or a break point in the signal sequence, the data are analyzed as soon as there are collected. In such application domains, score scales are not previously proposed or constructed as it is done in biological sequence analysis. When testing at each time i, the null hypothesis H_0: "The observations $(A_k)_{1\leqslant k\leqslant i}$ follow the distribution f_θ with parameter

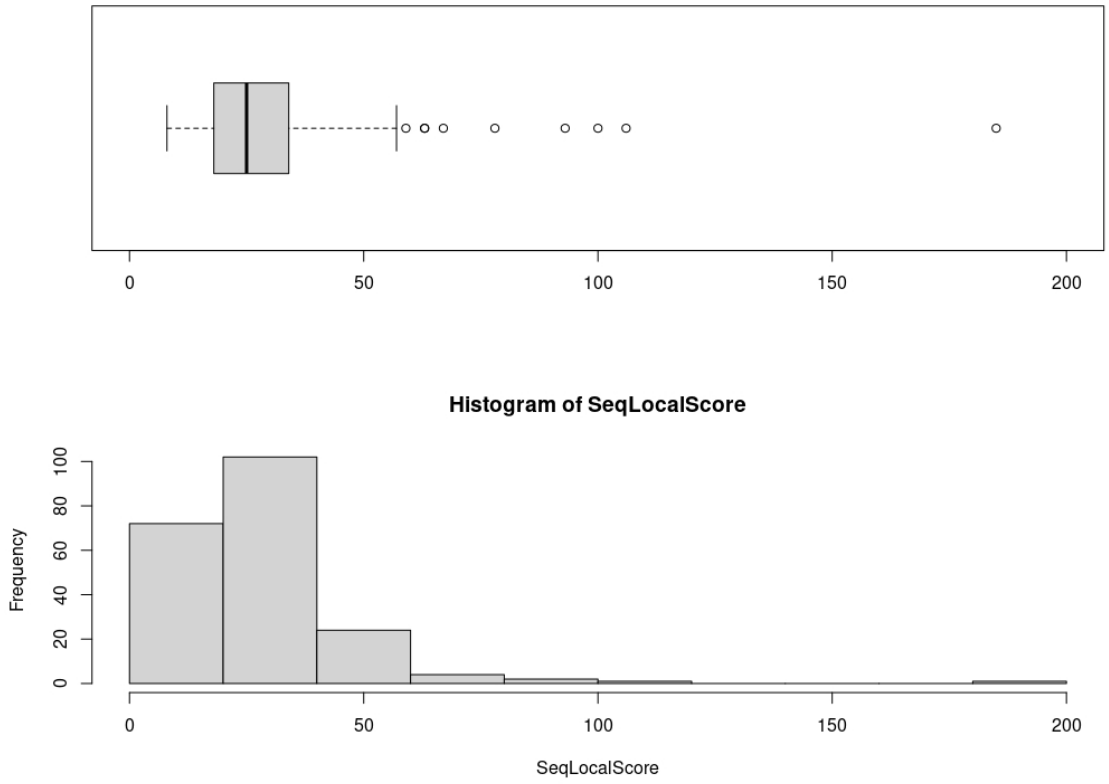

Figure 2 – Distribution of the local score of every sequence in the object `MySeqList`.

$\theta = \theta_0$" vs H_1: "The observations $(A_k)_{1 \leqslant k \leqslant i}$ follow f_{θ_1} with $\theta_1 \neq \theta_0$", it is usual to define the score of a given observation A_i at time i by the following Log Likelihood Ratio:

$$x_i = s(A_i) = \ln\left(\frac{f_{\theta_1}(A_i)}{f_{\theta_0}(A_i)}\right) .$$

Such a score function is used in this subsection and in the Subsection .

The local score can also be used to detect eventual shift in sequential observations. We propose here to analyze data on the apparition of the Stevens-Johnson syndrome, a serious dermatological disease due to a drug allergy.

```
R> data(SJSyndrome.data)
R> dim(SJSyndrome.data)
[1] 824 15
R> SJSyndrome.data[1:2,1:5]
Case.ID Initial.FDA.Received.Date days.since.last.fda Event.Date
1 4227848 10/01/1969 NA 02/16/1969
2 4227553 10/01/1969 0 07/10/1969
Latest.FDA.Received.Date
1 01-OCT-1969
2 01-OCT-1969
```

The third column `days.since.last.fda` corresponds to the number of days since the last event (the Time Between Event sequence). The data present 824 adverse event apparitions that lead to 823 Time Between two adverse Events (TBE) values in days.

```
R> DatesTBE <- SJSyndrome.data[-1,3] # the TBE sequence
R> n <- length(DatesTBE)
```

The TBE sequence can be modeled by a geometrical distribution. An estimation of its parameter is given by

```
R> p0Hat <- 1/(mean(DatesTBE[1:n]) - 1)
R> p0Hat
[1] 0.045349
```

with an estimated value equal to 0.045349 corresponding to the probability of observing an adverse event at a given day among the whole studied population. Let us denote $(T_i)_{1\leqslant i\leqslant n}$ the TBE observations. At each time i, we want to test the following hypotheses : H_0 "The observations $(T_k)_{1\leqslant k\leqslant i}$ follow a geometrical distribution with parameter p_0" vs H_1 "The observations $(T_k)_{1\leqslant k\leqslant i}$ follow a geometrical distribution with parameter $p_1 = 1.5 \cdot p_0$", with p_0 and p_1 in $]0,1[$.

Let us define:

$$LLR(T) = \ln \frac{f_1(T)}{f_0(T)}$$

with f_j the probability density function of a geometrical variable of parameter p_j for $j = 0, 1$. At each time i the local score of the sequence $(LRR(T_k))_{1\leqslant k\leqslant i}$ and its corresponding p-value are computed using the package. More precisely, we compute $LLR = \lfloor E \cdot \ln \frac{f_1(T)}{f_0(T)} \rfloor$ with E a tuning parameter which allows a larger range of possible non negative scores. The use of this tuning parameter does not change the segment that realizes the local score and neither its p-value (see Fariello et al., 2017 Supplementary materials, for more details), but allows to highlight suboptimal segments that could be interesting. We have here at least 3 non negative scores for $E = 8$.

```
R> p0 <- round(p0Hat,4)
R> p1 <- 1.5*p0
R> E <- 8
```

Let us compute the score sequence and the local score for each sequence up to index i for a sequential analysis.

```
R> ScoreSeq <- floor((log(dgeom(DatesTBE, p1) /
+    dgeom(DatesTBE, p0)))*E)
R> head(ScoreSeq)
[1] 3 -3 3 -15 -3 3
R> VectLS <- vector(length = n)
R> for (i in 1:n) {
+    VectLS[i] <- localScoreC(ScoreSeq[1:i])$localScore[1]}
R> head(VectLS)
R> tail(VectLS)
[1] 3 3 3 3 3 3
[1] 189 189 189 189 189 189
```

An alarm can be defined when the p-value of an observed local score value is less than a given nominal level, usually 5% or 1%. In order to establish the p-value, the distribution of the scores

under the H_0 hypothesis is needed. It is possible to established theoretically this distribution, but in order to have a lighter presentation here, we empirically estimate the score distribution on the data.

```
R> PkCal <- table(sort(ScoreSeq)) / length(ScoreSeq)
R> head(PkCal)
R> tail(PkCal)
[1] -109 -75 -61 -56 -54 -46
0.0012151 0.0012151 0.0012151 0.0036452 0.0012151 0.0012151
[1] -2 -1 0 1 2 3
0.036452 0.093560 0.117861 0.134872 0.227217 0.160389
```

We can notice that not all the possible values, between the minimum and the maximum score, are present. The vector of the score distribution must be fulfilled.

```
R> minXk <- min(ScoreSeq)
R> maxXk <- max(ScoreSeq)
R> score <- minXk:maxXk
R> PkEmp <- rep(0,length(score))
R> names(PkEmp) <- minXk:maxXk
R> for (i in 1:length(PkCal)) {
+    PkEmp[which(names(PkEmp) == names(PkCal)[i])] <- PkCal[i]}
R> head(PkEmp)
R> ProbaTh <- PkEmp
[1] -109 -108 -107 -106 -105 -104
0.0012151 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
R> mean(ScoreSeq)
[1] -1.633
```

The average score under H_0 is non positive so any method, exact as approximated ones, can be used to compute the statistical significance of the local score. We use the exact method with `daudin` function as the sequence lengths allow it.

```
R> VectPval <- vector("numeric", length = n)
R> for (i in 1:n) {
+    LS <- localScoreC(ScoreSeq[1:i])$localScore[1]
+    VectPval[i] <- daudin(localScore = LS,
+       sequence_length = i,
+       score_probabilities = ProbaTh,
+       sequence_min = minXk,
+       sequence_max = maxXk) }
R> head(VectPval)
[1] 0.16039 0.40797 0.58157 0.70423 0.79093 0.85222
R> min(which(VectPval < 0.05))
[1] 229
```

Figure 3 illustrates the example on the first 300 observations where a first alarm, using a nominal level $\alpha = 5\%$, appears at index 229.

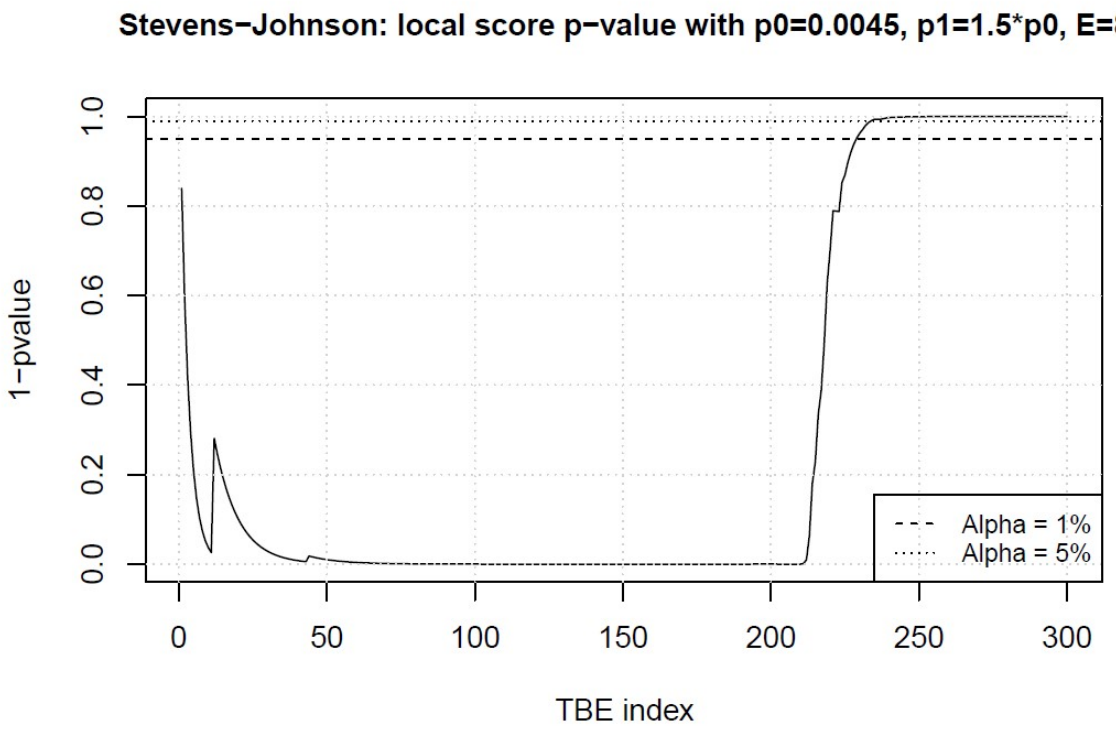

Figure 3 – Stevens Johnson syndrome: a unique alarm at index 229.

Different values for parameter p_1 has been tested, and each case leads to a similar result. Figure 3 representing the p-values at each index, can be seen as a control chart usually used to analyze on-line sequences in industrial data (see for example the first and the most famous control chart defined in 1930 and called the Shewhart chart, see W.A. Shewhart, 1931 mainly used for Gaussian distribution). We can see one unique clear alarm at index 229. In Mercier, 2020 for a Gaussian model, it is shown that using the local score avoids false alarm better than the usual control charts and allows to detect an existing change in the parameter in a competitive mean time.

Let us see the Shewhart g chart, adapted for geometrical distribution, and proposed in the package `qcc` in Figure 4.

```
R> library(qcc)
R> qcc(DatesTBE[1:300], type = "g")
R> Gchart <- qcc(DatesTBE[1:300], type = "g")
R> violating.runs(Gchart, run.length = qcc.options("run.length"))
```

Here the lower control limit (LCL) is equal to 0 and have no direct use. The twelve points upper than the upper control limit (UCL) in red are not “bad” alarm because they are corresponding to a longer run than expected between two adverse events and are then considered as an improved situation. We can observe several violating run in orange, corresponding to a particularly numerous successive points under the central control limit, that are considering as alarms. One is particularly long beginning at index 206 and including the index 229 of the alarm of the local score chart.

Regarding both local score and g charts we suggest that the violating runs before index 206 in the g chart could be considered as false alarms.

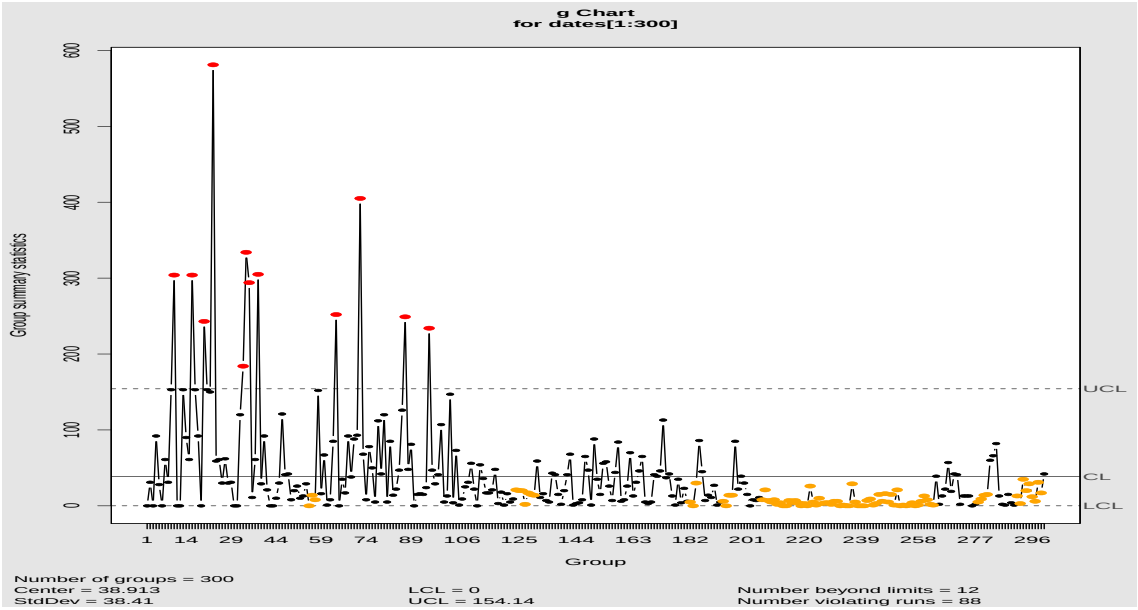

Figure 4 – Stevens Johnson syndrome - Shewhart g chart: A lot of alarms are pointed out. We can observe a violating run, corresponding to a particularly numerous successive points under the central control limit, beginning at index 206 and including the index 229 of the alarm of the local score chart.

Congenital oesophageal atresia data

The data consists of individual dates of birth over $n = 35$ cases of the birth defects oesophageal and tracheo-oesophagean fistula observed in a hospital in Birmingham, U.K., over 2191 days from 1950 through 1955, with Day one set as 1 January 1950 (see Knox, 1959). Glaz et al., 2009 present in Chapter 17 different works on these data based on the use of scan statistic. We first present in this section the results of the scan statistic analyses proposed in Glaz et al., 2009 and secondly two different approaches based on the local score. The discrete scan statistic $S_{n,k}$, with $k \leqslant n$ two positive integers, of a sequence $(S_i)_{1 \leqslant i \leqslant n}$ of n binary trials (1: success, 0: failure) has been defined as the maximum number of successes within any k consecutive trials. Let us consider a discrete sequence $(S_i)_{i=1\ldots n}$. We have $S_{n,k} = \max_{1 \leqslant i \leqslant n-k+1} \sum_{j=i}^{i+k-1} S_j$. The data are given in the following line code. We also derive the Time Between two Events (or success). The TBE sequence is modeled by a geometrical law of parameter p.

```
R> data(Aeso)
R> CasesIndex <- which(Aeso[,2] == 1)
R> tbe <- CasesIndex - c(0, CasesIndex[-length(CasesIndex)]) - 1
R> p <- sum(Aeso[,2]) / nrow(Aeso)
R> p
[1] 0.01597444
```

Scan statistic approach. Considering the sequence of the date (`Aeso[,2]` vector in the previous line code), Glaz et al., 2009 give the scan statistic values for different choices of the window length k; the corresponding statistical significance; and the position of the window that realizes the maximal value. They also present the method of Nagarwalla, 1996 using a scan statistic with a variable window size for which the statistical significance is established by Monte Carlo method. The results are presented in Table 2.

k	value	p-value	begin	end
100	7	0.08833	1233	1305
200	10	0.04993	1233	1390
300	15	0.00141	1233	1491
365	16	0.00271	1233	1583
Nagarwalla	15	0.00580	1233	1491

Table 2 – Results of the scan statistic approaches.

We can observe that the different statistical significances are very different depending on the window size choice: Using nominal level equal to 1%, we get not significant p-values for a window size $k = 100$ or $k = 200$ and significant ones for $k = 300, 365$ and for the Nagarwall method.

Log Likelihood Ratio test and local score approach. We propose here to consider an eventual drift in the parameter p. Let us consider H_0: $p = p_0$ and H_1: $p = p_0 \cdot (1 + \delta)$ for a given δ value. Let us consider first $\delta = 5\%$. We associate to the Time Between Events sequence (called `tbe` in the previous line code) the following score sequence computed using

$$X(tbe) = \lfloor E \cdot \ln\left(\frac{f_1(tbe)}{f_0(tbe)}\right) \rfloor$$

with f_i the probability for a random variable distributed as a geometrical law of parameter p_i, for $i = 0, 1$; E a tuning parameter we have previously presented in Subsection .

```
R> p0 <- p
R> delta <- 0.05
R> p1 <- p0 * (1 + delta)
R> # Choice for the value of E
R> # in order to allow at least 3 non negative scores
R> E <- 1
R> # Maximum of the scores in a geometrical model
R> maxXk <- floor(E * log(p1 / p0))
R> while (maxXk < 3) {
+   E <- E+1
+   maxXk <- floor(E * log(p1 / p0)) }
R> E
[1] 62
```

This leads to the following score sequence

```
R> ScoreSeq <- floor(E*log(dgeom(tbe, prob = p1) /
+    dgeom(tbe, prob = p0)))
R> ScoreSeq
R> minX <- min(ScoreSeq)
R> maxX <- max(ScoreSeq)
 [1]  -6  -5  -4   1 -21  -2  -2  -3   2   3   2   2   2   1  -1
[16]   2   2   0   2   2   2   1  -2  -3   2  -2  -4   0  -1   2
[31]   2   2   1   0   1
```

Score distribution. Let us compute the score distribution by two different ways: The first one based on the estimation on the score appearance on the observed sequence and a second way using theoretical work on geome-trical model that leads to a more accurate score distribution.

```
R> # Estimation on the data
R> ProbScoreInit <- table(ScoreSeq) / sum(table(ScoreSeq))
R> ProbScore <- rep(0, maxX - minX + 1)
R> names(ProbScore) <- minX:maxX
R> for (i in 1:length(ProbScoreInit)) {
+   w <- which(names(ProbScore) == names(ProbScoreInit)[i])
+   ProbScore[w] <- ProbScoreInit[i] }
R> fonctionProbaX <- function(x,E,p0,p1) {
+    calcul1 <- ((x/E)-log(p1/p0)) / log((1-p1)/(1-p0))
+    calcul2 <- (((x+1)/E) - log(p1/p0)) / log((1-p1)/(1-p0))
+    prob <- pgeom(prob=p0, floor(calcul1)) -
+            pgeom(prob=p0,floor(calcul2))
+    return(prob) }
R> ProbScoreTheo <- rep(0, maxX - minX + 1)
R> names(ProbScoreTheo) <- minX:maxX
R> for (i in 1:length(ProbScoreTheo)) {
+    ProbScoreTheo[i]<-fonctionProbaX(
+    x = as.numeric(names(ProbScoreTheo)[i]),
+    E, p0, p1 = (1 + delta) * p0) }
R> head(cbind(ProbScore, ProbScoreTheo))
```

```
     ProbScore   ProbScoreTheo
-21 0.02857143    0.0001725077
-20 0.00000000    0.0002380570
-19 0.00000000    0.0003285136
-18 0.00000000    0.0004533418
-17 0.00000000    0.0006256021
-16 0.00000000    0.0008132325
```

```
R> tail(cbind(ProbScore, ProbScoreTheo))
```

```
    ProbScore   ProbScoreTheo
-2 0.11428571      0.07592718
-1 0.05714286      0.09869924
0  0.08571429      0.14228152
1  0.14285714      0.19634549
2  0.37142857      0.27095262
3  0.02857143      0.01597444
```

Let us then compute, for the given shift δ, the local score value and its p-value with the two different score distributions. As the length sequence is very short, $n = 35$, we use the exact method to establish the p-value with the function `daudin`. The begin and end index are also given.

```
R> localScoreC(ScoreSeq)
```

```
$localScore
value begin   end
   22     9    22
$suboptimalSegmentScores
      value begin end
[1,]     1     4   4
[2,]    22     9  22
$RecordTime
[1] 4 9
R> LS <- localScoreC(ScoreSeq)$localScore[1]
R> BeginTbe <- localScoreC(ScoreSeq)$localScore[2]
R> EndTbe <- localScoreC(ScoreSeq)$localScore[3]
R> BeginDate <- CasesIndex[BeginTbe - 1]
R> EndDate <- CasesIndex[EndTbe]
R> pvalue <- daudin(
+   localScore = LS,
+   sequence_length = length(tbe),
+   score_probabilities = ProbScoreTheo,
+   sequence_min = minX,
+   sequence_max = maxX )
R> pvalue
[1] 0.02647577
R> BeginDate
[1] 1233
R> EndDate
[1] 1491
```

The segment that realizes the local score value begins at the date index 1233 and ends at the date index 1491 which corresponds to the segment highlighted by the scan statistic approach with a window size $k = 300$. Its statistical significance of the observed local score is around 0.026.

One could say that the choice of the window length in the scan statistic method does not have to be done in the local score one. But the choice is change in choosing a δ value to construct the score function based on p_0 and p_1. When no previous knowledge exists on the length of the segment we want to highlight, it is easier to choose the smallest drift we would like to detect. Let us have a look for a set of different δ values from 1% to 5%.

We can observed in Table 3 that the local score value does not change and neither the segment that realizes the local score: See $b.tbe$ (*respectively* $b.date$) the begin index in the tbe (*resp.* date sequence and see $e.tbe$ (*respectively* $e.date$) the end index in the tbe (*resp.* date) sequence. Moreover, its statistical significance is quite constant and around 3%.

Direct analysis on the 0-1 sequence. We propose below to analysis the initial sequence of occurrences (0-1) without constructing the TBE sequence. The model is then based on a Bernoulli distribution with still parameter $p_0 = 0.01597444$. Let us consider a drift $p_1 = 1.05 \cdot p_0$. We have

δ	E	Local score	$p-$value	b.tbe	e.tbe	b.date	e.date
0.01	302	22	0.02962169	9	22	1233	1491
0.02	152	22	0.02835914	9	22	1233	1491
0.03	102	22	0.02757781	9	22	1233	1491
0.04	77	22	0.02735163	9	22	1233	1491
0.05	62	22	0.02647577	9	22	1233	1491

Table 3 – Local score value, its statistical significance, the begin and end indices position in the *tbe* sequence and in the *date* sequence, for different value of δ. We also give in the second column the tuning parameter E used to get at least three non negative scores.

two different score values: $(\ln(p_1(1-p_0)/(p_0(1-p_1))) + \log((1-p_1)/(1-p_0)))$ corresponding to 1 and $\ln((1-p_1)/(1-p_0))$ to 0.

```
R> occur <- Aeso[,2]
R> p1 <- 1.05 * p0
R> p1
[1] 0.01677316
R> (log(p1*(1-p0) / (p0*(1-p1))) + log((1-p1) / (1-p0)))
[1] 0.04879016
R> log((1-p1) / (1-p0))
[1] -0.0008120179
```

These first score values lead us to choose a tuning parameter E equal to 1000 in order to keep the proportion between those two values and to get integer values which allow to use the exact method. We recall that this change in the score function has only consequences on the local score value, but no ones on the segment that realized the local score and neither the statistical significance.

```
R> E <- 1000
R> ScoreSeq2 <- floor(E*(occur*log(p1*(1-p0) / (p0*(1-p1)))
+    + log((1-p1) / (1-p0))))
R> table(ScoreSeq2)
R> localScoreC(ScoreSeq2)$localScore
score.seq2
  -1   48
2156   35
$localScore
value begin   end
  476  1233  1491
```

The two possible scores are then: -1 and 48. We observe that the segment that realizes the local score is still the same than with the geometrical model with a begin index 1233 and an end index 1491.

Let us have a look on the statistical significance. But first let us give the score distribution. The probabilities under H_0 associated to the scores are equal to $1-p_0$ for -1 and p_0 for 48.

```
R> ProbScores <- rep(0,50)
R> names(ProbScores) <- -1:48
```

```
R> ProbScores[1] <- 1-p0
R> ProbScores[length(ProbScores)] <- p0
R> # expected score
R> -1*(1-p0) + 48*p0
[1] -0.2172524
R> LS <- localScoreC(ScoreSeq2)$localScore[1]
R> daudin(
+    localScore = LS,
+    sequence_length = length(ScoreSeq2),
+    score_probabilities = ProbScores,
+    sequence_min = -1,
+    sequence_max = 48 )
[1] 0.02497491
```

The p-value is of the same size as the previous study with the local score approach and with the geometrical model on the TBE sequence.

The two studies based on the local score highlight the same segment than the scan statistic with a window size choice of 300 and than the one highlighted by the method of Nagarwalla with a variable window. This segment is statistically significant in each method. The local score both avoids the window length choice and allows the statistical significance to be theoretically established.

Genomic regions associated with phenotypic differentiation of European local pig breeds

The original dataset is based on European local pig breeds genetically characterized using DNA-pool sequencing data and phenotypically characterized using breed level phenotypes related to stature, fatness, growth and reproductive performance traits. It is composed of 19 populations of European local pig breeds and 7 populations of industrial breeds. The genetic diversity is assessed through a SNP (Single Nucleotide Polymorphism) of medium density array leading to 16,403,270 SNPs covering 18 chromosomes of the pig genomes after filtering out SNPs with missing data. The second part of the original dataset consists of phenotype characterizations of each breed, combined into four distinct groups summarizing stature, fatness, growth, and reproductive performance. The purpose of the study published in Poklukar et al., 2023 is to detect genomic regions with selection signatures linked to phenotypic traits in order to uncover potential candidate genes that may be under adaptation to specific environments. The methododology in Poklukar et al., 2023 uses the same approach of Coop et al., 2010 leading to elaborate a Bayes Factor measuring the link between phenotypic and genotypic variations for each SNP. Statistical significance is then assessed on a SNP by SNP base correcting the multitest problem with a False Discovery Rate (FDR) approach from Benjamini and Hochberg, 1995. They finaly revealed 234 regions associated with stature, fatness, growth or reproduction traits.

Here we propose to use a local score approach to analyze the final dataset containing the Bayes Factors associated to stature traits kindly provided by the authors of Poklukar et al., 2023. For each of the about 16 millions SNPs covering 18 pig chromosomes, we have the SNP positions and the associated Bayes Factor statistics. Table 4 shows the number of points for each chromosome. Note that a Bayes Factor is a real number, and p-values associated to the local

score can not be directly assessed by the function `karlin`, `mcc` nor `daudin` as there associated methodologies request integer scores. A proposed solution is to discretize the scores. In the second part of this illustration, we also assess the effect of this discretisation on the results, comparing three schemes : 1. real scores 2. scores multiplied by 10 and rounded to closest unit 3. scores rounded to the closest unit. Due to the length of the sequence, we proceed to the evaluation of *p*-value through `karlinMonteCarlo` and `karlinMonteCarlo_double` functions see Formula (4).

Table 4 – Number of SNPs by pig chromosome in the dataset.

Chromosome	SNPs count
1	1427539
2	1072176
3	946332
4	923731
5	792366
6	1206701
7	899034
8	1110422
9	1020531
10	695091
11	664610
12	563400
13	1225446
14	1029162
15	899320
16	694637
17	570599
18	417965

Data analysis. In order to analyze this big data file, we proceed chromosome by chromosome and use the `R` library `sqldf` to load the data. Below is the `R` code to create a sqlite database file 'morpho_new_all_Dim1.sqlite' with the data file 'morpho_new_all_Dim1.flkadapt', then to load the data of chromosome 1 (refseq id : 'NC_010443.5') into a dataframe call 'NC_010443.5'.

```
R> library(sqldf)
R> read.csv.sql(file="./morpho_new_all_Dim1.flkadapt",
+    sql = c("attach 'morpho_new_all_Dim1.sqlite' as new",
+    "create table new.morpho as select * from file"), sep=" ")
R> NC_010443.5 <- sqldf(paste(
+    "select * from morpho where chr='NC_010443.5'",
+    "AND [converge.null]='1' AND converge='1' ;",
+    sep=" "),
+    dbname = "morpho_new_all_Dim1.sqlite")
```

Without lost of generality, we present here the detailed analysis and results on the chromosome 1.

```
R> summary(NC_010443.5$bf)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.9372 -1.4153 -1.1766 -0.8759 -0.6375 12.5007
```

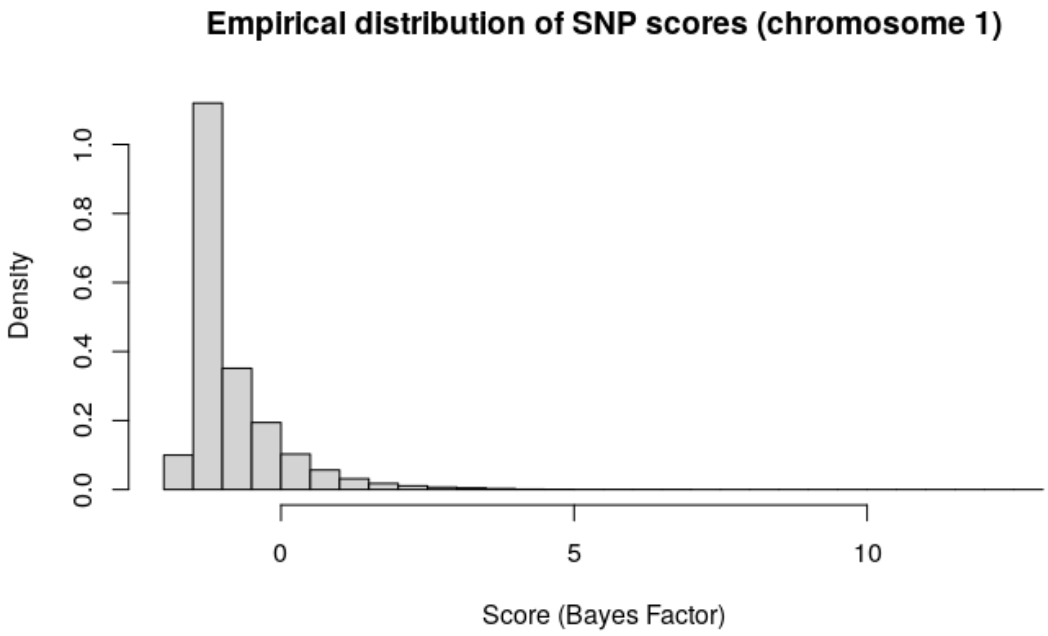

Figure 5 – Empirical SNP score distribution observed on the chromosome 1.

Note that the empirical expectation of the score is strictly negative and there is strictly positive scores as expected for a meaningful local score analysis. Figure 5 shows the empirical score distribution obtain on chromosome 1.

Figure 6 shows the SNPs score observed along the chromosome 1 and the associated Lindley process calculated as follows.

```
R> ScoresLindley <- lindley(NC_010443.5$bf)
```

The local score, its position, and all the suboptimal scores are calculated by the `localScoreC` function.

```
# Calculate localscore and suboptimal localscores.
R> scores <- localScoreC_double(NC_010443.5$bf)
R> print(scores$localScore)
      value       begin         end
   702.7715 642973.0000 645759.0000
# Position of the segment realizing the local score
R> print(NC_010443.5$pos[scores$localScore[c("begin","end")]])
[1] 86184149 86566846
# Length (in base-pairs) of this segment
R> diff(NC_010443.5$pos[scores$localScore[c("begin","end")]])
[1] 382697
# Calculate p-value of the local score and estimation
# of the Karlin parameters K* and Lambda
R> ResKarlinMC <- karlinMonteCarlo_double(
+    local_score = scores$localScore["value"],
+    sequence_length = length(NC_010443.5$bf),
```

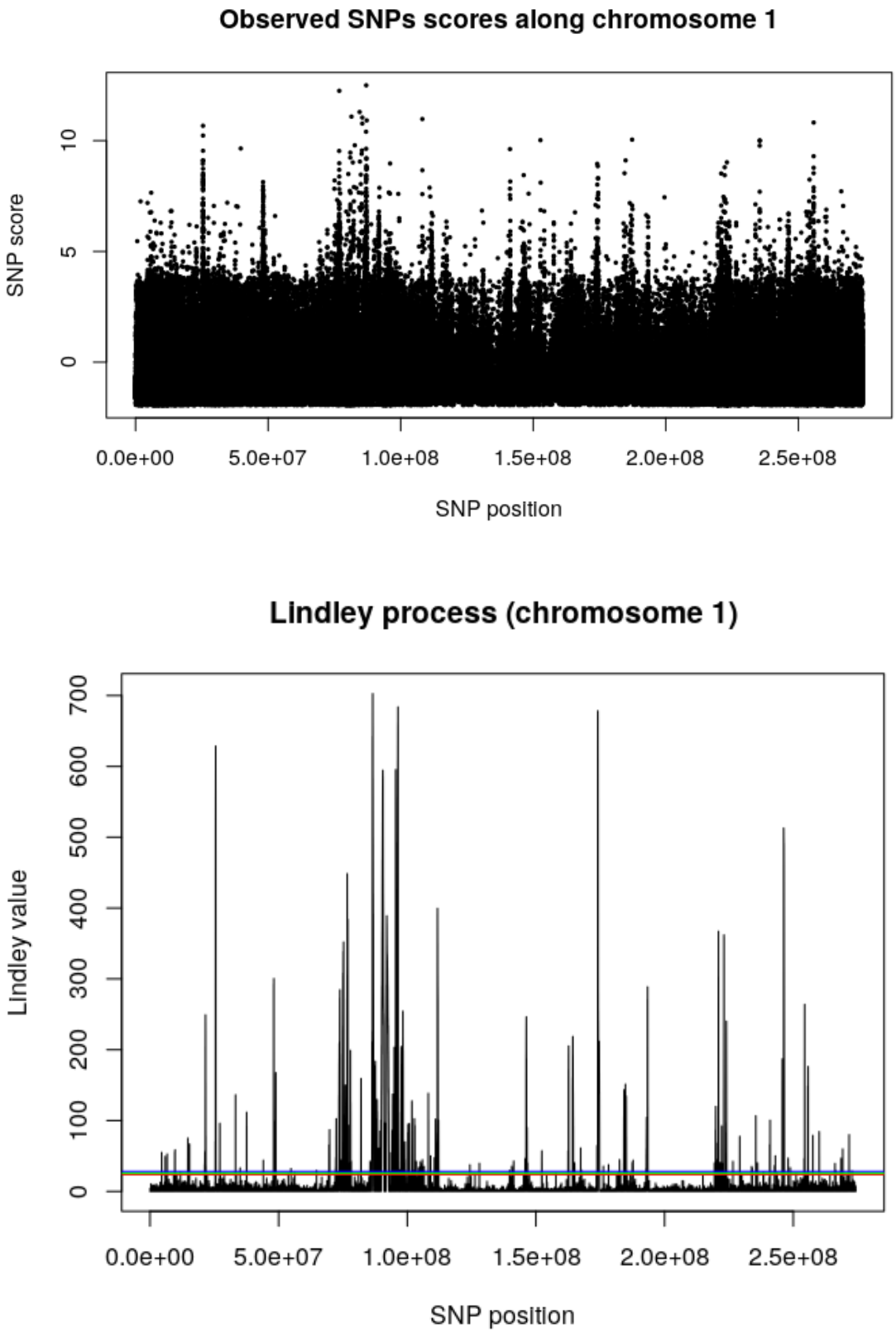

Figure 6 – SNP scores observed along the chromosome 1: Top) SNP Score values; Bottom) Associated Lindley process, with horizontal lines representing the thresholds associated to the statistical significance of the local score at 5% (red), 1% (green) and 1‰(blue) levels.

```
+    FUN = function(x, simulated_sequence_length)
+          {return(sample(x = x,
+           size = simulated_sequence_length,
+           replace = TRUE))},
+    x = NC_010443.5$bf,
+    simulated_sequence_length = 10000,
+    numSim = 1000, plot = FALSE)
R> kStar <- ResKarlinMC$`K*`
R> lambda <- ResKarlinMC$lambda
R> print(ResKarlinMC)
$`p-value`
value
    0

$`K*`
[1] 11.18694

$lambda
[1] 0.8285248
```

The local score on the chromosome 1 is 702.7715 and is realized by the segment situated in position (86184149, 86566846) with a p-value $< 10^{-16}$.

In the same way, we assess the statistical significance of the sub-optimal segments scores. As mentioned in Fariello et al., 2017, the local score threshold given for a first order risk α also ensures a first order risk α for at least one false positive among all excursions above this threshold. In other word all excursions above this threshold can be considered as significant sub-optimal segments scores. On the chromosome 1, we found a total of 67535 segments with positive cumulative score, from which 225 segments appear to be significant at 5%-level, 210 segments at 1%-level, and 183 segments at 1‰-level. See code below to compute these results.

```
R> SegmentScores <- as.data.frame(scores$suboptimalSegmentScores)
R> print(dim(SegmentScores)[1])
[1] 67535
R> # Sorting sub-optimal segments scores in decreasing
R> # order of scores
R> DecScoreOrder <- order(SegmentScores$value, decreasing = TRUE)
R> SegmentScores = SegmentScores[DecScoreOrder,]
R> # Calculating "p-values" majorant of each suboptimal
R> # segment scores until a threshold of 5\% is reached.
R> k_star <- res.karlinMC$`K*`
R> lambda <- res.karlinMC$lambda
R> alphaMax <- 0.05
R> pv <- 0.0
R> SegmentScores$pvkarlinMC <- NA
R> i <- 1
R> while ((pv <= alphaMax) && (i <= dim(SegmentScores)[1])) {
```

```
+   # Karlin: for n great,
+   #P( ln(n)/lambda+x>= M) = exp(-K_star*exp(-lambda*x))
+   # thus we set ln(n)/lambda+x = local_score and obtain
+   # x = local_score - ln(n)/lambda
+   x <- SegmentScores$value[i] -
+           log(length(NC_010443.5$bf)) / lambda
+   # now we calculate p-value with our approximate
+   # K star and lambda
+   pv <- 1 - exp(-k_star * exp(-lambda * x))
+   if (pv <= alphaMax) {
+     SegmentScores$pvkarlinMC[i] <- pv
+   }
+   i <- i + 1
+ }
R> SegmentScoresSignif <-
+    SegmentScores[!is.na(SegmentScores$pvkarlinMC),]
R> # Number of significative sub-optimal segments scores at
R> # thhreshold 0.05, 0.01 and 0.001
R> alpha <- c(0.05, 0.01, 0.001)
R> print(sapply(alpha,
+         function(x){return(sum(
+               SegmentScoresSignif$pvkarlinMC <= x))}))
[1] 225 210 183
```

The chromosome 1 contains a total of 1421525 SNPs with individual scores. The proportion of SNPs presents in significative segments compared to this total, approximates 4% (see Table 5). In the paper Poklukar et al., 2023, the whole genome analysis based on the Bayes Factor retains 2 segments significative at 5% corrected for multiple test, compared here to 225 segments significatively detected by the local score approach for the stature trait.

Table 5 – Numbers and proportions of SNPs present in a significative segment according to the test threshold.

Test threshold	0.05	0.01	0.001
Number of SNPs present in significant segment	54129	52954	51357
Proportion of SNP present in significant segment	0.04	0.04	0.04

Score discretization assessment. Three scoring schemes are compared: a) real scores as given by the Bayes Factor in input b) one decimal scores times 10 then rounded c) rounded scores to closest unit. b) and c) give integer scores. Figure 7 shows the empirical distributions of each score scheme obtained from chromosome 18. Other chromosomes show very close distributions (not shown).

For each chromosome of the whole genome, Table 6 summaries the number of significant detected segments applying a level of 5%, 1% and 1‰. These numbers are also shown regarding the scoring scheme.

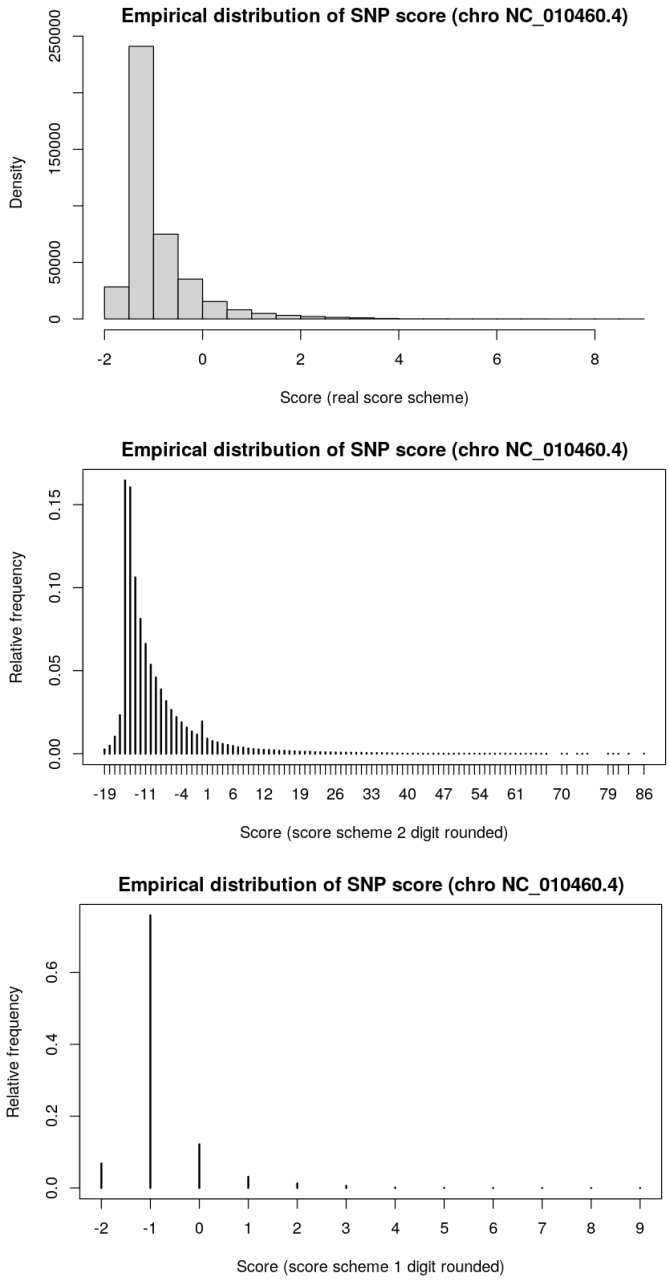

Figure 7 – Empirical distributions of SNP scores obtained on chromosome 18 for three scoring scheme: 1. Real score 2. Two-digits rounded score 3. One-digit rounded score.

Let's also look at the influence of the three scoring schemes on the length of segments that achieve the local score: Figure 8 shows comparable boxplots of the (log)-length of the detected segments below the threshold of 5% for the three scoring schemes.

Considering the segments obtained with the real scores as reference, Table 7 displays the numbers of false positive and false negative segments which occur with 1-digit scores and 2-digits scores. Care should be taken as a cut-off is applied to p-value less than 5% which can modify the significative segments list close to the cut-off. Regarding the big range and the skewness distribution of real scores, the 2-digits rounded scores just slightly change the results, missing only 9 (0.5%) segments over 1882 real segments on the whole scale and detecting falsely 45

Table 6 – Numbers of significant detected segments applying a level of 5%, 1% and ‰. These numbers regarding the scoring scheme are also shown.

chromosomes		Real scores			2-digits scores			Unit scores		
		5%	1%	5‰	5%	1%	5‰	5%	1%	5‰
1	NC_010443.5	226	212	185	225	213	185	218	201	179
2	NC_010444.4	70	60	49	70	59	48	67	55	44
3	NC_010445.4	56	54	46	56	53	45	56	53	45
4	NC_010446.5	90	78	70	87	78	67	89	80	71
5	NC_010447.5	94	85	67	94	85	68	105	92	72
6	NC_010448.4	93	83	74	102	91	78	109	96	88
7	NC_010449.5	94	83	71	96	85	72	96	84	72
8	NC_010450.4	131	118	108	131	118	107	127	121	112
9	NC_010451.4	132	120	105	134	123	107	145	136	120
10	NC_010452.4	108	97	86	111	98	86	123	104	91
11	NC_010453.5	67	58	50	66	57	50	60	51	44
12	NC_010454.4	52	46	40	55	47	41	56	46	41
13	NC_010455.5	161	153	130	162	156	135	160	150	133
14	NC_010456.5	135	122	105	143	126	112	140	127	107
15	NC_010457.5	142	124	110	157	138	120	152	133	113
16	NC_010458.4	74	69	64	75	71	65	81	76	69
17	NC_010459.5	70	62	57	68	61	57	68	61	57
18	NC_010460.4	87	71	61	87	73	61	86	73	63
	Total	1882	1695	1478	1919	1732	1504	1938	1739	1521

(2.3%) segments over 1919 detected segments. Note that the brutal unit rounded score performance essentially reflects the real segment detected with only 76 (4%) missing segments (false negative) over 1882 and detect falsely (false positive) 157 (8%) segments over 1938 detected segments.

Table 7 – Considering real scoring scheme as detection reference, the table shows the numbers of segments which differ from the reference for the 2-digit rounded score scheme, and the unit rounded scheme on the whole genome analysis. "False negative": number of segment which are present in the reference, but not in considered scoring scheme; "False positive": number of segments significantly detected but not present in the reference.

score scheme	Real	2-digits	1-digit
Total detected segments (<5%)	1882	1919	1938
False negative		9 (0.5%)	76 (4%)
False positive		45 (2.3%)	157 (8%)

Computational details

The results in this paper were obtained using `R` 4.3.1. `R` itself and all packages used are available from the Comprehensive `R` Archive Network (CRAN) at `https://CRAN.R-project.org/`.

Summary and discussion

When no a priori information is known about the length of the segments to be highlighted, the local score is a dedicated tool to exploit and supplement the methods of sliding-windows or

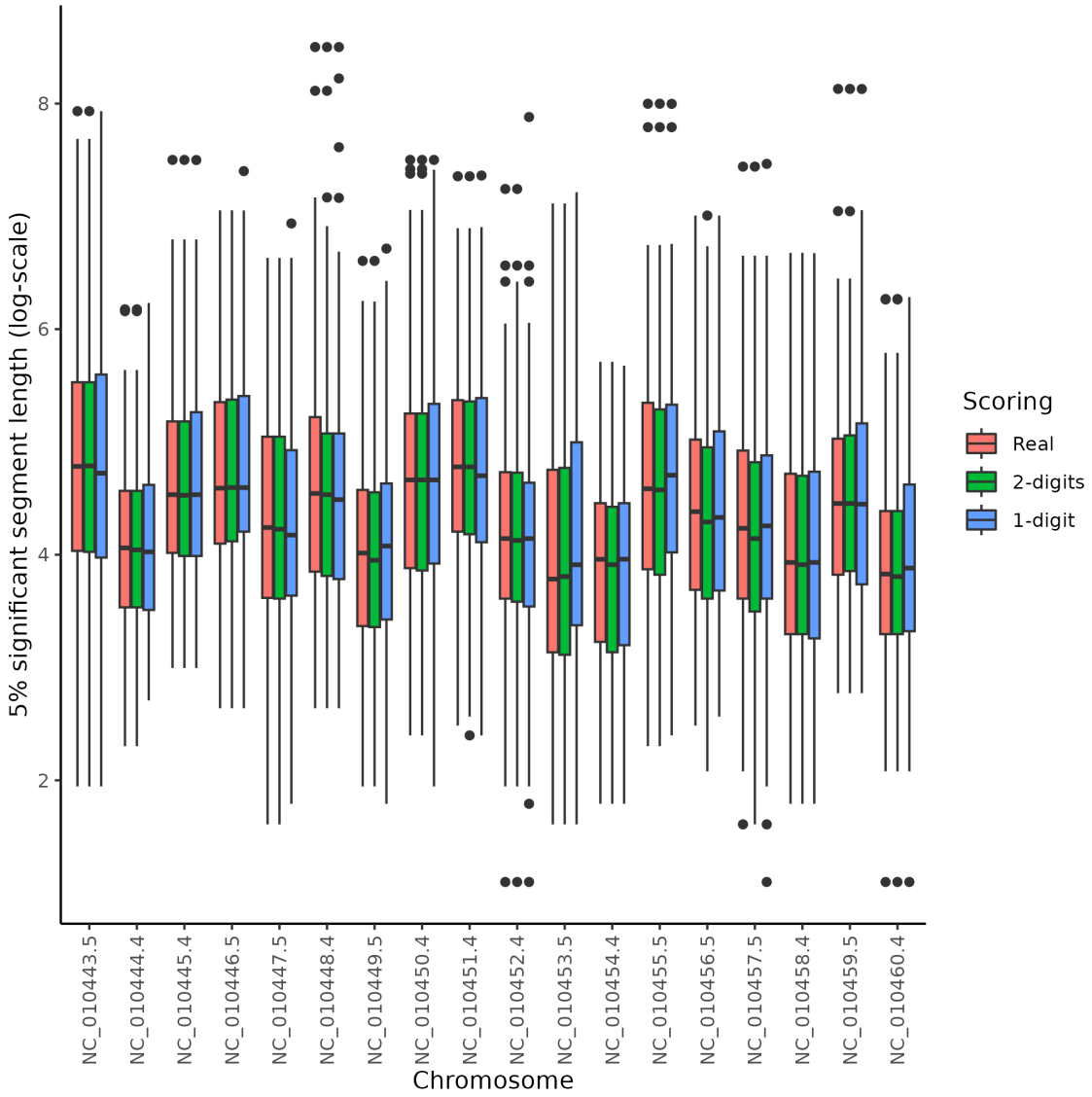

Figure 8 – Log-length of the detected segments below the threshold of 5% for the three scoring schemes by chromosomes.

scan statistics. In addition, the package allows to calculate the statistical significance and to distinguish the segments of atypical optimal scores from those appearing by chance. The package brings together various functions that notably allow to visualize and point-out the highlighted regions. Different ways of assessing the statistical significance are proposed. A function allows to perform this calculation by automatically selecting the method most suited to the context related to the length of the sequence, and the average score under a given model or learned. If initially the local score has been defined for the identification of atypical regions within biological sequences, it can also be useful in many fields of application as we wanted to illustrate in our examples. It can also be applied for online analyses including breakpoint detection. Further developments will be made for Markov models and the statistical significance of sub-optimal segments in a future version of the package.

Acknowledgements

The authors thank Sebastian Simon who began to build the package during his internship.

Fundings

This work has been supported by the project "Highlight" of the *Excellence Laboratory International Center of Mathematics and Computer Science in Toulouse* (Labex CIMI).

Conflict of interest disclosure

The authors declare that they comply with the PCI rule of having no financial conflicts of interest in relation to the content of the article.

Data, script, code, and supplementary information availability

The source code of the `localScore` package is available at: https://cran.r-project.org/web/packages/localScore/index.html. The package also contains the datasets analyzed in the article except the Genomic dataset that is not publicly available so far.

References

Benjamini Y, Hochberg Y (1995). *Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing*. *Journal of the Royal Statistical Society: Series B (Methodological)* **57**, 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.

Cellier D, Charlot F, Mercier S (2003). *An improved approximation for assessing the statistical significance of molecular sequence features*. *Jour. Appl. Prob.* **40**, 427–441. https://doi.org/10.1239/jap/1053003554.

Chabriac C, Lagnoux A, Mercier S, Vallois P (2014). *Elements related to the largest complete excursion of a reflected BM stopped at a fixed time. Application to local score*. *Stochastic Processes and their Applications* **124**, 4202–4223. https://doi.org/10.1016/j.spa.2014.07.003.

Coop G, Witonsky D, Di Rienzo A, Pritchard JK (2010). *Using environmental correlations to identify loci underlying local adaptation*. *Genetics* **185**, 1411–1423. https://doi.org/10.1534/genetics.110.114819.

Fariello M, Boitard S, Mercier S, Robelin D, Faraut T, Arnould C, Le Bihan-Duval E, Recoquillay J, Salin G, Dahais G, Pitel F, Leterrier G, Sancristobal M (2017). *Accounting for Linkage Disequilibrium in genome scans for selection without individual genotypes : the local score approach*. *Molecular Ecology* **26(14)**, 3700–3714. https://doi.org/10.1111/mec.14141.

Glaz J, Pozdnyakov V, Wallenstein S (2009). *Scan statistics - Methods and applications*. Birkhauser Bosten. Chap. 1. https://doi.org/10.1007/978-0-8176-4749-0.

Grusea S, Mercier S (2020). *Improvement on the distribution of maximal segmental score in a Markovian sequence*. *Journal of Applied Probability* **57.1**, 29–52. https://doi.org/10.1017/jpr.2019.75.

Hassenforder C, Mercier S (2007). *Exact Distribution of the Local Score for Markovian Sequences*. *AISM* **59**, 741–755. https://doi.org/10.1007/s10463-006-0064-6.

Karlin S, Altschul SF (1990). *Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes*. *PNAS* **87**, 2264–2268. https://doi.org/10.1073/pnas.87.6.2264.

Karlin S, Dembo A (1992). *Limit distributions of maximal segmental score among Markov-dependent partial sums*. *AdAP* **24**, 113–140. https://doi.org/doi.org/10.2307/1427732.

Knox G (1959). *Secular pattern of congenital oesophageal atresia*. *British Journal of Preventive Social Medicine* **13**, 222–226. https://doi.org/10.1136/jech.51.2.110.

Kyte J, Doolittle R (1982). *A simple method for displaying the hydropathic character of a protein*. *Journal of molecular biology* **157**, 105–132. https://doi.org/10.1016/0022-2836(82)90515-0.

Lagnoux A, Mercier S, Vallois P (2017). *Statistical significance based on length and position of the local score in a model of i.i.d. sequences*. *Bioinformatics* **33**, 654–660. https://doi.org/10.1093/bioinformatics/btw699.

Mercier S (2020). *Transferring biological sequence analysis tools to break-point detection for on-line monitoring: A control chart based on the Local Score*. *Qual. Reliab. Engng. Int.* **36**, 2379–2397. https://doi.org/doi.org/10.1002/qre.2703.

Mercier S, Daudin J (2001). *Exact Distribution for the Local Score of One i.i.d. Random Sequence*. *Jour. Comp. Biol* **8**, 373–380. https://doi.org/10.1089/106652701752236197.

Nagarwalla N (1996). *A scan statistic with a variable window*. *Statistics in Medecine* **15**, 845–850. https://doi.org/10.1002/(sici)1097-0258(19960415)15:7/9<845::aid-sim254>3.0.co;2-x.

Poklukar K, Mestre C, Škrlep M, Čandek-Potokar M, Ovilo C, Fontanesi L, Riquet J, Bovo S, Schiavo G, Ribani A, Muñoz M, Gallo M, Bozzi R, Charneca R, Quintanilla R, Kušec G, Mercat MJ, Zimmer C, Razmaite V, Araujo JP, et al. (2023). *A meta-analysis of genetic and phenotypic diversity of European local pig breeds reveals genomic regions associated with breed differentiation for production traits*. *Genetics Selection Evolution* **55**. https://doi.org/10.1186/s12711-023-00858-3.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. URL: https://www.R-project.org/.

Simon S, Robelin D, Mercier S, Dejean S (2023). *localScore: Package for Sequence Analysis by Local Score*. R package version 1.0.11.

W.A. Shewhart (1931). *Economic Control of Quality of Manufactured Product*. D. Van Nostrand Company, Inc., New York.