TAIL-GREEDY BOTTOM-UP DATA DECOMPOSITIONS AND FAST MULTIPLE CHANGE-POINT DETECTION

By Piotr Fryzlewicz*

London School of Economics

This article proposes a 'tail-greedy', bottom-up transform for one-dimensional data, which results in a nonlinear but conditionally orthonormal, multiscale decomposition of the data with respect to an adaptively chosen Unbalanced Haar wavelet basis. The 'tail-greediness' of the decomposition algorithm, whereby multiple greedy steps are taken in a single pass through the data, both enables fast computation and makes the algorithm applicable in the problem of consistent estimation of the number and locations of multiple change-points in data. The resulting agglomerative change-point detection method avoids the disadvantages of the classical divisive binary segmentation, and offers very good practical performance. It is implemented in the R package `breakfast`, available from CRAN.

1. Introduction. Multiple change-point detection in data observed over time or on a one-dimensional spatial domain is a problem of fundamental importance in applied statistics and has its uses in, for example, bioinformatics (recombination detection (Minin et al., 2005), prediction of transmembrane helix locations (Lio and Vanucci, 2000), segmentation of microarray data (Erdman and Emerson, 2008), detection of changes in the DNA copy number (Olshen et al., 2004; Venkatraman and Olshen, 2007)), medicine (estimation of phase transitions in pain symptoms (Desmond et al., 2002)), climate (analysis of tropical cyclone activity (Chu and Zhao, 2004)), security applications (monitoring for denial-of-service attacks (Wang, Zhang and Shin, 2004) and other intrusions in computer networks (Tartakovsky et al., 2006)), linguistics (text segmentation (Choi, 2000)), audio and video processing (audio segmentation (Lu, Zhang and Jiang, 2002), speech segmentation (Shriberga et al., 2000), temporal video segmentation (Koprinska and Carrato, 2001)), quality control (calibration for aircraft testing (Mahmoud et al., 2007)), or economics and finance (identifying and dating change-points in stock market volatility (Aggarwal, Inclan and Leal, 1999) and in the evolution of macroeconomic variables (Bai and Perron, 2003), adaptive trend estimation in mar-

*Work supported by the Engineering and Physical Sciences Research Council grant no. EP/L014246/1.

Keywords and phrases: tail-greediness, bottom-up methods, multiscale methods, segmentation, thresholding, sparsity

kets (Schroeder and Fryzlewicz, 2013)).

The above list of applications spans a range of data structures (time series, regression models, image-valued signals) and includes problems requiring a posteriori or online change-point detection. Even if the interest is in online detection, then a posteriori detection of multiple change-points, sometimes referred to as segmentation, can often serve as the useful first step in the exploratory analysis of data, and is also the focus of this work.

In this paper, we work in the canonical change-point sequence model

$$X_t = f_t + \varepsilon_t, \quad t = 1, \dots T, \tag{1}$$

where f_t is a deterministic, one-dimensional, piecewise-constant signal with change-points whose number N and locations $\eta_1, \dots, \eta_N$ are unknown. In this article, we assume that the ε_t's are iid $N(0,1)$; in the supplemental article Fryzlewicz (2017), we show how to extend our methodology to heavy-tailed, dependent noise. Our task is to estimate N and $\eta_1, \dots, \eta_N$ under certain assumptions on N, the magnitudes of the jumps and the minimum distance between the change-point locations, which will be specified later.

Most of the literature on a posteriori multiple change-point detection falls into one of two broad categories. In the first category, change-points are found by minimising a criterion function comprising a likelihood-type (or least-squares) term measuring the fit of the estimate to the data and a penalty term to prevent overfitting. In this category, Yao and Au (1989) consider least-squares estimation of f_t for a fixed N and iid noise. Braun, Braun and Mueller (2000) extend this work to noise for which the variance is a function of the mean. In the Gaussian case, the Schwarz criterion is used to estimate an unknown but bounded N in Yao (1988), and a more general criterion but also linear in the number of change-points appears in Lee (1995). For an unknown but bounded N, Lavielle and Moulines (2000) consider penalised least-squares estimation, with a penalty linear in the number of change-points, for dependent ε_t's; see also Lavielle (1999) and Lavielle (2005) for related work. For a fixed N, Pan and Chen (2006) propose a likelihood criterion with a penalty depending not only on the number, but also on the locations of change-points, favouring more uniformly-spread estimated change-points; related ideas appear in Zhang and Siegmund (2007). For an unknown N, Lebarbier (2005) proposes least-squares estimation with a penalty originating from the model selection approach of Birge and Massart (2001). Boysen et al. (2009) use the least-squares criterion with a linear penalty on the number of change-points. More general forms of Schwarz-like penalties are studied, for example, in Wu (2008), Ciuperca (2011) and Ciuperca (2014). The SMUCE estimator of Frick, Munk and Sieling (2014) can also be

cast in the penalised cost function framework. An empirical Bayes approach to change-point estimation in a marginal likelihood framework appears in Du, Kao and Kou (2016).

Until recently, penalised approaches could be criticised for being slow; the typical computational speed of $O(T^2)$ (see e.g. Auger and Lawrence (1989) and Jackson et al. (2005)) made them impractical for large datasets. Some recent efforts made it less of an issue: the PELT algorithm by Killick, Fearnhead and Eckley (2012) and the pruned dynamic programming by Rigaill (2015) both reduce the speed to linear in best-case scenarios (while retaining quadratic speed in worst-case ones), and offer fast implementations. Related ideas appear also in Maidstone et al. (2017). Our experience is that penalty-based methods in which the penalty has not been chosen adaptively from the data can struggle to offer uniformly good performance across a variety of signals "of all shapes and sizes"; we illustrate this behaviour in Section 4.2. Other attempts to reduce the computational complexity of the problem include Davis, Lee and Rodriguez-Yam (2006), who (in a time series setting) use a genetic algorithm to minimise a Minimum Description Length criterion, and Harchaoui and Lévy-Leduc (2010) who consider the least-squares criterion with a total variation penalty, which enables them to use the LARS algorithm of Efron et al. (2004) to compute the solution in $O(NT\log(T))$ time. However, the total variation penalty is not an optimal one for change-point detection (in the sense of balancing out type-I and type-II errors, as described in Brodsky and Darkhovsky (1993) and Cho and Fryzlewicz (2011)). The total variation penalty is also considered in the context of peak/trough detection by Davies and Kovac (2001), who propose the 'taut string' approach for fast computation. In the context of multiple change-point detection, it is considered by Rinaldo (2009) (with a subsequent correction published on the author's web page) and Rojas and Wahlberg (2014) as part of the fused lasso penalty, proposed by Tibshirani et al. (2005) and equivalent to taut string in model (1). Wang (1995) uses the fast discrete wavelet transform to detect change-points. Huskova and Slaby (2001) and Eichinger and Kirch (2018) propose the "moving sum" (MOSUM) technique, which requires the choice of an extra bandwidth parameter. An early review of some multiple change-point detection methods appears in Braun and Mueller (1998). Killick et al. (2012) is a repository of publications and software related to change-point detection.

Besides penalty-based approaches, the other class of methods widely used in multiple change-point detection are those based on Binary Segmentation (BinSeg) and its variants. BinSeg is a generic technique in which, initially, the entire dataset is searched for one change-point. If a change-point is

detected, the data are then split into two subsegments, defined by the detected change-point. A similar search is then performed on both subsegments, possibly resulting in further splits. The recursion on a given segment continues until a certain criterion is satisfied on it. BinSeg is a 'greedy' procedure in the sense that it is performed sequentially, with each stage depending on the previous ones. Each stage involves one-dimensional optimisation. Possibly the first work to propose BinSeg in a stochastic process setting is Vostrikova (1981), who shows consistency of BinSeg for the number and locations of change-points for a fixed N. Venkatraman (1992) offers a proof of the consistency of BinSeg for N and the change-point locations, even for N increasing with T, albeit with sub-optimal assumptions and rates for the locations. In a setting similar to Vostrikova (1981) (for a fixed N and with ε_t following a linear process), BinSeg is considered in Bai (1997). Chen, Cohen and Sackrowitz (2011) provide a proof of consistency of BinSeg for the number of change-points in the case of fixed N and iid normal ε_t. BinSeg is used for univariate time series segmentation in Fryzlewicz and Subba Rao (2014) and Cho and Fryzlewicz (2012), and for multivariate, possibly high-dimensional time series segmentation in Cho and Fryzlewicz (2015). The benefits of BinSeg include low computational complexity (typically of order $O(T\log(T))$), conceptual simplicity, and ease of implementation.

Despite these appealing features, BinSeg can perform poorly in the more challenging settings, see e.g. Fryzlewicz (2014). This is due to its "top-down" character, by which we mean that it initially considers the entire dataset, and then narrows its focus from longer to shorter segments of the data. Each stage of BinSeg involves search for a *single* change-point, which means that if a given segment contains *multiple* change-points in certain unfavourable configurations, BinSeg may fail to perform adequately on it, as it attempts to fit the "wrong" model. Fryzlewicz (2014) shows that relatively restrictive theoretical assumptions are needed for BinSeg to offer near-optimal performance in terms of the accuracy of estimation of the change-point locations. Circular BinSeg (Olshen et al., 2004; Venkatraman and Olshen, 2007), Wild BinSeg (Fryzlewicz, 2014) and the Narrowest-Over-Threshold method (Baranowski, Chen and Fryzlewicz, 2016) are designed to improve the performance of BinSeg, but at the cost of increased computational speed.

In this work, we take the view that multiple change-point detection is a natural 'multiscale' problem, i.e. one best solved by combining local and global information from the data in a hierarchical fashion. We propose a new approach to multiple change-point detection in model (1), although its principles extend to more complex settings. Recalling that the root reason for the

frequently weak performance of BinSeg is its top-down nature, the methodology we propose has a "bottom-up" character, in the sense that its core mechanism consists of *successively merging those neighbouring regions of the data which are most likely to correspond to locally constant underlying signal* f, rather than successively sub-dividing data most likely to be separated by change-points, as is done in BinSeg. In other words, our approach leads to an "agglomerative" algorithm, in contrast to the "divisive" character of BinSeg. Agglomerative ideas in the context of change-point detection appear rarely in the literature, possibly because such algorithms tend to be more difficult to analyse than divisive ones. Matteson and James (2014) present a heuristic agglomerative algorithm for multiple change-point detection as a computationally attractive alternative to their divisive one. Solutions to the fused lasso (Tibshirani et al., 2005) can be found in a bottom-up way. In the broader context of nonparametric regression, the "lifting one coefficient at a time" methodology of Jansen, Nason and Silverman (2009) also involves a bottom-up but linear procedure, unlike the non-linear and data-adaptive decomposition methodology proposed in this work.

An important attribute of our new bottom-up method is what we term "tail-greediness". In the purely greedy approach, in each pass through the data, we would only be merging *one* pair of neighbouring regions, the one that is the most likely to correspond to locally constant f, as was done in the heuristic procedure outlined in Fryzlewicz (2007). However, this leads to a slow algorithm (of computational order $O(T^2)$) and does not guarantee statistical consistency in change-point detection, or even when the error in estimating f is measured in the L_2 norm. In this work, we instead propose to merge *multiple* (we later specify precisely how many) pairs of neighbouring regions in each pass through the data, those corresponding to the most likely, the second most likely, etc., segment of constancy in f. Since this algorithm considers the entire lower tail of the distribution (of certain localised measures of variability in X_t), we refer to it as "tail-greedy". A "tail-greedy" algorithm is 'less greedy than a greedy one'. The use of tail-greediness buys several attractive properties at once: it leads to an algorithm of computational complexity $O(T \log^2(T))$ regardless of the complexity of the signal or the number of change-points, and it enables statistical consistency of our procedure both in the L_2 sense and (after some post-processing) in detecting the number and locations of the change-points in f. The key reason for this is that each element of X_t is processed at most $O(\log(T))$ number of times. A similar 'tail-greedy' device, but not referred to by this name, was used in Fryzlewicz and Timmermans (2016) in an image analysis context with the purpose of accelerating computation; however, unlike in the current work,

no mention was made of the theoretical performance of the resulting estimator or how tail-greediness affected it. As in Fryzlewicz (2007), our algorithm involves transformation of the data with respect to a particularly selected Unbalanced Haar basis; but as our basis is constructed via a tail-greedy process, it is inherently different, in important ways already described above, from that in Fryzlewicz (2007). Our proposed transformation will be referred to as the Tail-Greedy Unbalanced Haar (TGUH) transform of the data.

The TGUH decomposition algorithm naturally generates a multiscale (hierarchical) data-adaptive decomposition of $X = (X_1, \ldots, X_T)'$ into a set of difference-type coefficients that form an unbalanced unary-binary tree. The resulting transformation is non-linear, but conditioning on the order in which region merges are performed, it is a linear and orthonormal transform of X.

A particularly attractive feature of the bottom-up TGUH approach is that it offers good practical performance (see Section 4.2), is fast and its speed of execution does not depend on the number of change-points in data. In particular, it performs well for signals with a large number of change-points, a setting in which its competitors tend to struggle. Other potentially attractive aspects of the TGUH methodology are discussed in Section 5. The TGUH methodology is implemented in the R package `breakfast`.

The paper is organised as follows. Section 2 outlines the TGUH data decomposition algorithm and change-point detection methodology, and Section 3 describes its theoretical properties. Section 4 discusses the choice of the parameters of the procedure, compares its finite-sample performance with that of the state-of-the-art competitors, and illustrates it on a data example. Section 5 offers an additional discussion. Proofs of our main theoretical results are in Appendix A and in the supplemental article Fryzlewicz (2017).

2. Methodology. Our procedure for estimating the number N and the locations $\eta_1, \ldots, \eta_N$ of change-points in f_t proceeds in four steps, listed below and described in the following four subsections.

1. Tail-Greedy Unbalanced Haar (TGUH) decomposition of the input data $X = (X_1, \ldots, X_T)'$. The transformation is multiscale in the sense that it decomposes X into a set of adaptively constructed detail-type coefficients, which are arranged into a natural unary-binary tree (i.e. one in which "parent" nodes have one or two "children"). In this process, a particular data-adaptive Unbalanced Haar (UH) basis of R^T is constructed.
2. Thresholding stage, in which the detail coefficients whose magnitude is less than a user-specified threshold are set to zero, as long as this does not spoil the connectedness of the tree of non-zero coefficients.

3. The inverse transform to that in step 1. At this stage, a piecewise-constant pre-estimator $\tilde{f}$ is produced, which is consistent for f in the L_2 sense, but which possibly contains spurious estimated change-points.
4. Post-processing stage, in which the spurious estimated change-points in $\tilde{f}$ are removed with a high probability, leading to the final estimator $\hat{f}$ of N and $\eta_1, \ldots, \eta_N$.

2.1. *Tail-Greedy Unbalanced Haar transformation.* The main idea of the standard Binary Segmentation applied in model (1) is to explain as much variance in the data as possible in a greedy, top-down fashion by iteratively fitting to X those step functions with one change-point which lead to the smallest residual sums of squares. In our proposed alternative approach, we take the opposite view and start by explaining as *little* variance of the data as possible at "fine" level of resolution (i.e. initially considering variability between consecutive pairs of observations, and then progressively moving up to larger regions). Explaining the least possible variance amounts to iteratively merging those local regions of the domain $\{1, \ldots, T\}$ which are the most likely to lie within segments of constancy of f. Because the transform we define is conditionally orthonormal, it preserves the sample variance of the input data X. Therefore, the bulk of the variance of the input data must be captured in later stages of the transform, which induces a sparse representation of X and enables change-point detection. Another key new feature of our proposed transform is "tail-greediness", defined and explained below. We now outline the decomposition algorithm, which will be referred to as the Tail-Greedy Unbalanced Haar (TGUH) transformation. (Some readers may find it useful to refer first to the illustrative Example of the TGUH transform provided later in this section.)

1. Introduce notation and initiate variables.
   (a) Assign the initial "smooth" (local rescaled average) coefficients to be the data: $\mathbf{s} = (s_{1,1}, s_{2,2}, \ldots, s_{T,T}) := (X_1, X_2, \ldots, X_T)$. The two subscripts in $s_{p,r}$ denote the initial (p) and final (r) index of the region of the data used to compute $s_{p,r}$. Initially, each time point $\{t\}, t = 1, \ldots, T$, is a separate region; as the algorithm progresses, we always have $s_{p,r} = (r - p + 1)^{-1/2} \sum_{s=p}^{r} X_s$.
   (b) Set $j := 1$; the parameter j describes the "scale" of the transform. After each pass through the data (described below) the scale j will increase by 1. At the "finest" scale $j = 1$, merges will take place between some neighbouring regions that are all individual time points $\{t\}, t = 1, \ldots, T$. At the "coarsest" scale $j = J$, there will

be a single merge between regions $\{1,\dots,q\}$ and $\{q+1,\dots,T\}$ for a certain $q \in \{1,\dots,T-1\}$.

(c) Let $\rho \in (0,1)$ be a constant (usually close to zero), used to describe the number of pairwise region merges to take place at scale j, as a function of the number of regions remaining after $j-1$ scales of the transform. More precisely, let $\alpha_{j,T}$ denote the number of regions remaining after $j-1$ scales of the transform; we have $\alpha_{1,T} = T$. At scale j, the algorithm merges (up to) $\lceil \rho \alpha_{j,T} \rceil$ pairs of regions.

2. At each scale j, search the vector $\mathbf{s}$ for $\lceil \rho \alpha_{j,T} \rceil$ "detail" coefficients (each representing a suitably scaled difference between the mean values of the data over a pair of neighbouring regions) that are the lowest in magnitude. To be more precise, proceed as follows: for each pair of neighbours $(s_{p,q}, s_{q+1,r})$, construct a "detail" filter $(a_{p,q}, -b_{q+1,r})$, where $a_{p,q}, b_{q+1,r} > 0$, in the following way.

(a) In order for the transform to produce as sparse a representation of the input data as possible, the transform needs to produce zero details over regions of constancy of the data. Thus, one requirement on $(a_{p,q}, -b_{q+1,r})$ is that if $(X_p,\dots,X_r)$ is a constant vector, the detail coefficient, defined by

$$d_{p,q,r} := a_{p,q} s_{p,q} - b_{q+1,r} s_{q+1,r}, \tag{2}$$

should be zero.

(b) To preserve the orthonormality of the transform, another requirement on $(a_{p,q}, -b_{q+1,r})$ is $a_{p,q}^2 + b_{q+1,r}^2 = 1$.

The two requirements (a) and (b) above determine that

$$a_{p,q} = \left\{ \frac{r-q}{r-p+1} \right\}^{1/2}, \qquad b_{q+1,r} = \left\{ \frac{q-p+1}{r-p+1} \right\}^{1/2}. \tag{3}$$

3. Sort the sequence $|d_{p,q,r}|$ in non-decreasing order. No region must be merged more than once at each scale j; therefore, consider each element of the sorted sequence $|d_{p,q,r}|$ from the smallest to the largest. If the current element considered, indexed (p,q,r), is such that any elements preceding it (i.e., smaller in magnitude), have been constructed as a function of $s_{p,q}$ or $s_{q+1,r}$, remove the current element from the sorted sequence $|d_{p,q,r}|$, and proceed to consider the next smallest element of this sequence in a similar way.

4. Extract $\lceil \rho\alpha_{j,T} \rceil$ detail coefficients $d_{p,q,r}$ corresponding to the $\lceil \rho\alpha_{j,T} \rceil$ smallest elements of the thus-processed sorted sequence $|d_{p,q,r}|$. It may be the case that after the removal process described in step 3 above, there will be fewer than $\lceil \rho\alpha_{j,T} \rceil$ detail coefficients left, in which case extract all of them. Denote these (up to) $\lceil \rho\alpha_{j,T} \rceil$ extracted detail coefficients $d_{p,q,r}$ by $d^{(j,k)}_{p,q,r}$, where $k = 1, \ldots, K(j)$ indexes them according to increasing p. Store the detail coefficients $d^{(j,k)}_{p,q,r}$ in memory.

   Note: the term "tail-greedy" originates from the fact that we are simultaneously extracting (up to) $\lceil \rho\alpha_{j,T} \rceil$ detail coefficients that are the smallest in magnitude, and hence target the *entire lower tail* of the distribution of their magnitudes. The tail-greediness has important and far-reaching consequences for the computational complexity and the theoretical guarantees of the performance of the algorithm, and we discuss them later.
5. For each $d^{(j,k)}_{p,q,r}$, using the filter $(b_{q+1,r}, a_{p,q})$, which is orthogonal to $(a_{p,q}, -b_{q+1,r})$, produce the corresponding new "smooth" coefficient $s_{p,r} = b_{q+1,r}s_{p,q} + a_{p,q}s_{q+1,r} = s_{p,r} = (r-p+1)^{-1/2} \sum_{s=p}^{r} X_s$. Replace the pair of neighbours $(s_{p,q}, s_{q+1,r})$ with the new smooth coefficient $s_{p,r}$, i.e. "merge" the regions $\{p, \ldots, q\}$ and $\{q+1, \ldots, r\}$ into the single region $\{p, \ldots, r\}$.

   Note: The new (detail, smooth) pair $(d^{(j,k)}_{p,q,r}, s_{p,r})$ is the result of a rotation of the pair $(s_{p,q}, s_{q+1,r})$. To see this, note that

   (4)
   $$\begin{pmatrix} d^{(j,k)}_{p,q,r} \\ s_{p,r} \end{pmatrix} = \begin{bmatrix} a_{p,q} & -b_{q+1,r} \\ b_{q+1,r} & a_{p,q} \end{bmatrix} \begin{pmatrix} s_{p,q} \\ s_{q+1,r} \end{pmatrix} =: \Lambda_{p,q,r} \begin{pmatrix} s_{p,q} \\ s_{q+1,r} \end{pmatrix}.$$

   The rotation interpretation comes from the orthonormality of $\Lambda_{p,q,r}$.
6. Set $j := j + 1$ and go to 2., unless only one detail coefficient was extracted in step 4, for which, necessarily, $(p, r) = (1, T)$. At this point, the TGUH transform is completed.

Denote by J the largest value of the scale parameter j for which a merge of regions took place. The vector of detail coefficients $d^{(j,k)}_{p,q,r}$ produced as above, along with the single remaining smooth coefficient $s_{1,T}$, defines the Tail-Greedy Unbalanced Haar (TGUH) transform of the input vector $X = (X_1, \ldots X_T)'$: $\mathrm{TGUH}(X) := (d^{(j,k)}_{p,q,r}, \quad j = 1, \ldots, J; \quad k = 1, \ldots, K(j)) \,\|\, (s_{1,T})$, where the $\|$ symbol denotes vector concatenation.

The coefficients $d^{(j,k)}_{p,q,r}$ can be viewed as scalar products between X and a particular basis of Unbalanced Haar (UH) vectors (Fryzlewicz, 2007) cho-

sen from the data, which is however substantially different from that proposed in the above work, as it has been selected via a tail-greedy algorithm. More formally, let $\psi^{(j,k)}$ be vectors such that $d^{(j,k)}_{p,q,r} = \langle X, \psi^{(j,k)} \rangle$ and $s_{1,T} = \langle X, \psi^{(0,0)} \rangle$, then the set $\{\psi^{(j,k)}\}$ is an orthonormal Unbalanced Haar basis for R^T. Occasionally, we will write $d^{(j,k)}$ or $d_{p,q,r}$ for $d^{(j,k)}_{p,q,r}$, as for a particular TGUH transform, there is a unique correspondence between the values of the triples p, q, r and those of the pairs (j, k). Although TGUH(X) is a data-adaptive and non-linear transform of X (in the sense that $\{\psi^{(j,k)}\}$ is also a function of X, i.e. formally, $\psi^{(j,k)} = \psi^{(j,k)}(X)$), it is also linear and orthonormal conditioning on the order in which the regions have been merged. In particular, with $\bar{X} = T^{-1} \sum_{s=1}^{T} X_s$, this implies Parseval's identities $\sum_{s=1}^{T} X_s^2 = \sum_{j=1}^{J} \sum_{k=1}^{K(j)} (d^{(j,k)})^2 + s_{1,T}^2$ and $\frac{1}{T} \sum_{s=1}^{T} X_s^2 - (\bar{X})^2 = \frac{1}{T} \sum_{j=1}^{J} \sum_{k=1}^{K(j)} (d^{(j,k)})^2$. Since the coefficients $d^{(j,k)}$ arising at fine scales (i.e. those indexed by small values of j) are small in magnitude by construction, the above identities imply that the bulk of the variance of the input vector X_t is captured by the detail coefficients $d^{(j,k)}$ arising at coarser scales (i.e. those indexed by larger values of j). This typically implies a certain "sparsity of representation" by which the main features of the input vector X_t tend to be encoded in only a few, usually coarse-scale, detail coefficients $d^{(j,k)}$.

Example. We now provide a cartoon example of how the algorithm may proceed at the first two scales ($j = 1, 2$). Suppose the input vector is $(X_1, \ldots, X_9)$ and ρ is such that the algorithm merges two regions at both scale $j = 1$ and scale $j = 2$.

Scale $j = 1$. From formula (3), we have that at scale $j = 1$, the coefficients $(a_{p,q}, -b_{q+1,r}) = (1/\sqrt{2}, -1/\sqrt{2})$, since $p = q$ and $r = q + 1$, or in other words, all regions considered contain only individual data points. Suppose the two smallest details in absolute value are, in this order, $d_{3,3,4} = (X_3 - X_4)/\sqrt{2}$ and $d_{4,4,5} = (X_4 - X_5)/\sqrt{2}$. As described in Step 3 of the TGUH algorithm given earlier in this section, no region must be merged more than once at each scale, and therefore we must remove $d_{4,4,5}$ from consideration as X_4 is already used in the (smaller) coefficient $d_{3,3,4}$. Suppose the next smallest detail coefficient is $d_{6,6,7} = (X_6 - X_7)/\sqrt{2}$. As per Step 4 of the algorithm, we record the details $d^{(1,1)}_{3,3,4} = (X_3 - X_4)/\sqrt{2}$ and $d^{(1,2)}_{6,6,7} = (X_6 - X_7)/\sqrt{2}$. As described in Step 5, we replace the pairs of neighbours $(s_{3,3}, s_{4,4}) = (X_3, X_4)$ and $(s_{6,6}, s_{7,7}) = (X_6, X_7)$ with, respectively, $s_{3,4} = (X_3 + X_4)/\sqrt{2}$ and $s_{6,7} = (X_6 + X_7)/\sqrt{2}$. At the end of the above pass through the data at scale $j = 1$, the input vector will therefore be reduced

to $(s_{1,1}, s_{2,2}, s_{3,4}, s_{5,5}, s_{6,7}, s_{8,8}, s_{9,9}) = \left(X_1, X_2, \frac{X_3+X_4}{\sqrt{2}}, X_5, \frac{X_6+X_7}{\sqrt{2}}, X_8, X_9\right)$. At scale $j = 1$, we have merged the pairs of regions $(\{3\}, \{4\})$ and $(\{6\}, \{7\})$.

Scale $j = 2$. At scale $j = 2$, not all pairs of coefficients $(a_{p,q}, -b_{q+1,r})$ will be equal; as is apparent from formula (3), those used to contrast regions of size 1 with regions of size 2 will be different from those used to contrast pairs of regions of size 1. More specifically, for example, we will have $(a_{1,1}, -b_{2,2}) = (1/\sqrt{2}, -1/\sqrt{2})$, $(a_{2,2}, -b_{3,4}) = (\sqrt{2}/\sqrt{3}, -1/\sqrt{3})$ and $(a_{3,4}, -b_{5,5}) = (1/\sqrt{3}, -\sqrt{2}/\sqrt{3})$. Suppose the two smallest details in absolute value are $d_{3,4,5} = \frac{1}{\sqrt{3}}\left(\frac{X_3+X_4}{\sqrt{2}}\right) - \frac{\sqrt{2}}{\sqrt{3}}X_5$ and $d_{8,8,9} = \frac{X_8-X_9}{\sqrt{2}}$. We record the details $d^{(2,1)}_{3,4,5} = d_{3,4,5}$, $d^{(2,2)}_{8,8,9} = d_{8,8,9}$, and replace the pairs of neighbours $(s_{3,4}, s_{5,5})$ and $(s_{8,8}, s_{9,9})$ with, respectively, $s_{3,5} = \frac{\sqrt{2}}{\sqrt{3}}\left(\frac{X_3+X_4}{\sqrt{2}}\right) + \frac{1}{\sqrt{3}}X_5 = \frac{X_3+X_4+X_5}{\sqrt{3}}$ and $s_{8,9} = \frac{X_8+X_9}{\sqrt{2}}$. Therefore, at the end of the pass through the data at scale $j = 2$, the input vector will be reduced to $(s_{1,1}, s_{2,2}, s_{3,5}, s_{6,7}, s_{8,9}) = \left(X_1, X_2, \frac{X_3+X_4+X_5}{\sqrt{3}}, \frac{X_6+X_7}{\sqrt{2}}, \frac{X_8+X_9}{\sqrt{2}}\right)$.

We do not show subsequent scales j, but the transform continues until the input vector has been reduced to the single coefficient $s_{1,9} = \frac{1}{3}\sum_{i=1}^{9} X_i$. The coefficients $d^{(1,2)}_{6,6,7}$ and $d^{(2,2)}_{8,8,9}$ are both of the form $X_i/\sqrt{2} - X_{i+1}/\sqrt{2}$ (for $i = 6$ and $i = 8$, respectively), even though they "live" on different scales: $j = 1$ and $j = 2$, respectively. In this way, TGUH is different from the classical (non-adaptive) wavelet transform, where the scale is in one-to-one correspondence with the form of the filter used at that scale. The coefficients $d^{(j,k)}$ can be visualised as forming a unary-binary tree: for example, $d^{(2,1)}_{3,4,5}$ can be viewed as a "parent" coefficient of $d^{(1,1)}_{3,3,4}$ because $d^{(2,1)}_{3,4,5}$ is a function of $\{X_3, X_4, X_5\}$, and $d^{(1,1)}_{3,3,4}$ is a function of $\{X_3, X_4\}$, a subset of $\{X_3, X_4, X_5\}$. In this sense, the coefficient $d^{(2,1)}_{3,4,5}$ has no more "children" at scale $j = 1$. In general, a detail coefficient at scale $j+1$ may have 0, 1 or 2 children at scale j. This completes the example.

We now comment on the computational complexity of the TGUH transform. The number of regions remaining after j scales is at most $(1-\rho)^j T$, and, solving for the smallest j such that $(1-\rho)^j T \leq 1$, we have that the transform needs at most a logarithmic number of scales, $\lceil \log(T)/\log\{(1-\rho)^{-1}\} \rceil$, to terminate by reaching the scale, denoted by J, at which there is only one region remaining. The costliest step at each scale is the sorting in Step 3 of the TGUH algorithm, which takes up to $O(T \log(T))$ operations. Therefore, the computational complexity of the entire TGUH transform is $O(T \log^2(T))$. This is in contrast to the bottom-up transform introduced in

Fryzlewicz (2007), in which only one region merge takes place at each scale j, and therefore its computational complexity is $O(T^2)$. It is important to bear in mind that $O(T\log^2(T))$ is an upper bound; Section 4.1 offers numerical evidence that the actual average computation times of the (complete) TGUH algorithm may be faster.

2.2. *Connected thresholding of detail coefficients.* As per the discussion in the preceding section regarding the "sparsity of representation", the tail-greediness and the conditional orthonormality of the TGUH transform imply that the bulk of the variability of the input signal X will be concentrated in hopefully only a few large, coarse-scale detail coefficients $d^{(j,k)}$, while many fine-scale coefficients $d^{(j,k)}$ will be small and possibly carry mostly noise. Therefore, thresholding of the detail coefficients $d^{(j,k)}$ (i.e. setting the small ones to zero but retaining the large ones) appears to be a sensible strategy for removing noise from X_t and therefore estimating f_t and its change-points.

Let $f = (f_1, \ldots, f_T)'$ be the true unknown signal in (1) and let $\mu^{(j,k)} = \langle f, \psi^{(j,k)} \rangle$ be the UH coefficients of f, computed with respect to the basis $\{\psi^{(j,k)}\}$ chosen (from the data X) by the TGUH transform. We wish to estimate f by estimating each $\mu^{(j,k)}$ and then inverting the TGUH transform. For each (j,k), $j = 1, \ldots, J$, $k = 1, \ldots, K(j)$, let $\mathcal{C}_{j,k}$ denote the set of all those indices (j',k'), $j' = 1, \ldots, j$, $k' = 1, \ldots, K(j')$ for which $d^{(j',k')}_{p',q',r'}$ is a "child" of $d^{(j,k)}_{p,q,r}$, i.e. has been computed using a portion of the data X that is a subset of that used by $d^{(j,k)}_{p,q,r}$. More formally, $\mathcal{C}_{j,k} = \{(j',k'), j' = 1, \ldots, j, k' = 1, \ldots, K(j') \ : \ d^{(j',k')}_{p',q',r'}$ is such that $[p',r'] \subseteq [p,r]\}$. For $(j,k) \neq (0,0)$, we define our estimator $\hat{\mu}^{(j,k)}$ of $\mu^{(j,k)}$ by

$$\hat{\mu}^{(j,k)} = d^{(j,k)}_{p,q,r}\, \mathbb{I}\left\{\exists (j',k') \in \mathcal{C}_{j,k} \ \left|d^{(j',k')}_{p',q',r'}\right| > \lambda(q'-p'+1, r'-q')\right\}, \tag{5}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function and $\lambda(\cdot,\cdot)$ is a certain threshold, whose value will be specified later. In other words, we estimate $\mu^{(j,k)}$ by zero if and only if both $d^{(j,k)}_{p,q,r}$ and all its children coefficient fall below their respective thresholds; otherwise, $\mu^{(j,k)}$ is estimated by $d^{(j,k)}_{p,q,r}$. In this way, the unary-binary tree formed by the non-zero estimates $\hat{\mu}^{(j,k)}$ is connected. The process of producing this tree as described in formula (5) can be viewed as "pruning" those branches of the tree of the detail coefficients $d^{(j,k)}$ whose all components fall under their respective thresholds. This process promotes the survival of coarse-scale detail coefficients, and the killing off of fine-scale ones. We will refer to this procedure as "connected thresholding".

The main advantage of using connected thresholding is that the number of estimated change-points in $\tilde{f}$ (the piecewise-constant estimate f produced

by the inverse TGUH transformation of $\hat{\mu}^{(j,k)}$ described in Section 2.3) is equal to the number of coefficients $d^{(j,k)}$ that survive the thresholding. In diagonal thresholding (i.e. thresholding in which decisions as to whether to keep or kill a detail coefficients are made on the basis of the value of this coefficient alone), the number of estimated change-points is at least as large as the number of detail coefficients that survive the thresholding.

2.3. *Inverse TGUH transformation.* The inverse TGUH transformation, denoted by TGUH^{-1}, is performed in linear computational time by undoing the rotations specified in formula (4), in reverse order to that in which they were originally performed. Each rotation, being an orthonormal transform, is undone by using the same filters as those forming the rows of its corresponding matrix $\Lambda_{p,q,r}$. The initial estimate $\tilde{f}$ of f is obtained as

$$\tilde{f} = \mathrm{TGUH}^{-1}\left\{(\hat{\mu}^{(j,k)}, \quad j = 1, \ldots, J; \quad k = 1, \ldots, K(j)) \,\|\, (s_{1,T})\right\}. \tag{6}$$

We now illustrate the mechanics of the inverse TGUH transformation by continuing the Example of Section 2.1. One of the steps at scale $j = 2$ in the Example was the creation of the coefficient pair $(d_{3,4,5}, s_{3,5})$ from $(s_{3,4}, s_{5,5})$. This happened as the result of the rotation (see formula (4)):

$$\begin{pmatrix} d_{3,4,5} \\ s_{3,5} \end{pmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & -\frac{\sqrt{2}}{\sqrt{3}} \\ \frac{\sqrt{2}}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{pmatrix} s_{3,4} \\ s_{5,5} \end{pmatrix} =: \Lambda_{3,4,5} \begin{pmatrix} s_{3,4} \\ s_{5,5} \end{pmatrix}.$$

As $\Lambda_{3,4,5}$ is an orthonormal matrix, this operation is easily undone as follows, to produce the corresponding step of the inverse transform (i.e. to create $(s_{3,4}, s_{5,5})$ from $(d_{3,4,5}, s_{3,5})$):

$$\begin{pmatrix} s_{3,4} \\ s_{5,5} \end{pmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} \\ -\frac{\sqrt{2}}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{pmatrix} d_{3,4,5} \\ s_{3,5} \end{pmatrix}.$$

2.4. *Post-processing for consistent change-point detection.* By Theorem 3.1, the piecewise-constant estimator $\tilde{f}$ contains $\tilde{N} \leq C\,N\log(T)$ change-points with a high probability (where C is a constant), and therefore potentially overestimates the true number N of change-points. The post-processing described in this section proceeds in the following two stages.

Stage 1. Transformation of $\tilde{f}$ into an intermediate estimator $\tilde{\tilde{f}}$, which, like $\tilde{f}$, is consistent for f in the L_2 sense, but also potentially overestimates N. However, the number $\tilde{\tilde{N}}$ of change-points in $\tilde{\tilde{f}}$ is such that $\tilde{\tilde{N}} \leq 2(N+1)$. Unlike $\tilde{N}$, the bound on $\tilde{\tilde{N}}$ does not contain the $\log(T)$ term.

The estimator $\tilde{\tilde{f}}$ is constructed as follows. First, represent the pre-estimate $\tilde{f}$ as a smooth vector $\mathbf{s} = (s_{1,\tilde{\eta}_1-1}, s_{\tilde{\eta}_1,\tilde{\eta}_2-1}, \ldots, s_{\tilde{\eta}_{\tilde{N}},T})$, where $\tilde{\eta}_i$, $i = 1, \ldots, \tilde{N}$ are the time points, arranged in increasing order, such that $\tilde{f}_{\tilde{\eta}_i} \neq \tilde{f}_{\tilde{\eta}_i-1}$. With vector $\mathbf{s}$ as input, execute the TGUH algorithm as described in Section 2.1, under the following constraint: at each consecutive scale j, only produce one coefficient $d^{(j,1)}$, and only if $d^{(j,1)} = d^{(j,1)}_{p,q,r} \leq \lambda(q-p+1, r-q)$, where $\lambda(\cdot,\cdot)$ is the same as in Section 2.2. If the above inequality is not satisfied, stop and denoting the transformed smooth vector at that stage of the transform by $\mathbf{s}' = (s_{1,\tilde{\tilde{\eta}}_1-1}, s_{\tilde{\tilde{\eta}}_1,\tilde{\tilde{\eta}}_2-1}, \ldots, s_{\tilde{\tilde{\eta}}_{\tilde{\tilde{N}}},T})$, construct the intermediate estimator $\tilde{\tilde{f}}$ as

$$\tilde{\tilde{f}}_t = \frac{1}{\tilde{\tilde{\eta}}_i - \tilde{\tilde{\eta}}_{i-1}} \sum_{l=\tilde{\tilde{\eta}}_{i-1}}^{\tilde{\tilde{\eta}}_i-1} X_l \qquad \text{for} \qquad t \in [\tilde{\tilde{\eta}}_{i-1}, \tilde{\tilde{\eta}}_i - 1], \qquad i = 1, \ldots, \tilde{\tilde{N}}, \tag{7}$$

where $\tilde{\tilde{\eta}}_0 = 1$ and $\tilde{\tilde{\eta}}_{\tilde{\tilde{N}}+1} = T+1$. Since $\tilde{N} \leq C\,N\,\log(T)$, the reduction of the vector $\mathbf{s}$ to $\mathbf{s}'$ is particularly fast and takes $O(\log(T))$ operations.

Stage 2. Transformation of $\tilde{\tilde{f}}$ into the final estimator $\hat{f}$, which estimates the number N and the locations $\eta_1, \ldots, \eta_N$ of change-points in f consistently with a high probability.

The estimator $\hat{f}$ is constructed by pruning the estimated change-points $\tilde{\tilde{\eta}}_i$ as follows. (We use the convention $\tilde{\tilde{\eta}}_0 = 1$ and $\tilde{\tilde{\eta}}_{\tilde{\tilde{N}}+1} = T+1$.)

1. For each $i = 1, \ldots, \tilde{\tilde{N}}$, use formula (2) to compute the coefficient d_{p_i,q_i,r_i} where $p_i = \left\lfloor \frac{\tilde{\tilde{\eta}}_{i-1}+\tilde{\tilde{\eta}}_i}{2} \right\rfloor$, $q_i = \tilde{\tilde{\eta}}_i - 1$ and $r_i = \left\lceil \frac{\tilde{\tilde{\eta}}_{i+1}+\tilde{\tilde{\eta}}_i}{2} \right\rceil - 1$.
2. With $i_0 = \arg\min_i |d_{p_i,q_i,r_i}|$, if $|d_{p_{i_0},q_{i_0},r_{i_0}}| < \lambda(q_{i_0} - p_{i_0} + 1, r_{i_0} - q_{i_0})$, then remove $\tilde{\tilde{\eta}}_i$ from the set of estimated change-points, reduce $\tilde{\tilde{N}}$ by 1, relabel the remaining change-points (in increasing order) as $\tilde{\tilde{\eta}}_i$ for $i = 0, \ldots, \tilde{\tilde{N}} + 1$, and go to step 1.
3. Otherwise, stop, set $\hat{N} = \tilde{\tilde{N}}$ and relabel the remaining change-points (in increasing order) as $\hat{\eta}_i$ for $i = 0, \ldots, \hat{N} + 1$, where $\hat{\eta}_0 = 1$ and $\hat{\eta}_{\hat{N}+1} = T+1$.

The key aspect of Stage 2 of our post-processing procedure is that each coefficient d_{p_i,q_i,r_i} is computed in such a way that the left and right ends of its support, p_i and r_i respectively, are located mid-way between the current change-point $\tilde{\tilde{\eta}}_i$ being examined and its left-hand and right-hand neighbours (respectively). This ensures removal of the spurious change-points in $\tilde{\tilde{f}}$ while preserving the non-spurious ones, with high probability. The resulting esti-

mator $\hat{f}$ is constructed as

$$(8)\quad \hat{f}_t = \frac{1}{\hat{\eta}_i - \hat{\eta}_{i-1}} \sum_{l=\hat{\eta}_{i-1}}^{\hat{\eta}_i - 1} X_l \qquad \text{for} \qquad t \in [\hat{\eta}_{i-1}, \hat{\eta}_i - 1], \qquad i = 1, \ldots, \hat{N}.$$

$\hat{f}_t$ is our final estimator of f. We take $\hat{N}$ to be our estimator of N, the true number of change-points in f, and $\hat{\eta}_i$ to be our estimators of η_i, the true locations of change-points in f.

Two other possible refinements to our post-processing methodology are mentioned in the supplemental article Fryzlewicz (2017), Section 2.

3. Theoretical behaviour. Here and throughout the paper, for any estimator $\tilde{f}$ of f, we denote its squared empirical L_2 risk as $\|\tilde{f} - f\|_T^2 = T^{-1} \sum_{i=1}^T (\tilde{f}_i - f_i)^2$. This is a commonly used error measure whose expectation was also used as a risk measure e.g. in Donoho and Johnstone (1994). Whenever we refer to the estimators $\tilde{f}$ or $\tilde{\tilde{f}}$ as "L_2-consistent" or "consistent" below, we mean it in the sense that $\|\tilde{f} - f\|_T^2 \to 0$ or $\|\tilde{\tilde{f}} - f\|_T^2 \to 0$, respectively, on a set of probability approaching one with T. We first demonstrate the L_2 behaviour of the initial estimator $\tilde{f}$.

THEOREM 3.1. *Let X_t follow model (1) and let the errors ε_t be independent and standard normal. Let $\tilde{f}$ be as in formula (6). Fix any $\delta > 0$ and let the thresholding function $\lambda(\cdot, \cdot)$ of formula (5) take one of the two forms: $\lambda_1(u, v) = 2\{(1 + \delta) \log(T)\}^{1/2}$ or $\lambda_2(u, v) = \{2(1 + \delta) \log(T)\}^{1/2} (u^{1/2} + v^{1/2})/(u + v)^{1/2}$. On the set $\mathcal{A}_{\delta,T}$, defined by*

$$\mathcal{A}_{\delta,T} = \left\{ \forall\, 1 \le l \le m \le T \quad (m - l + 1)^{-1/2} \left| \sum_{i=l}^{m} \varepsilon_i \right| \le \{2(1 + \delta) \log(T)\}^{1/2} \right\},$$

which satisfies $P(\mathcal{A}_{\delta,T}) \to 1$ as $T \to \infty$ for all $\delta > 0$, we have

$$\|\tilde{f} - f\|_T^2 \le 2(1+\delta)\, T^{-1} \log(T) \left\{ 1 + (3 + 2\sqrt{2}) N \lceil \log(T) / \log\{(1 - \rho)^{-1}\} \rceil \right\},$$

and the piecewise-constant estimator $\tilde{f}$ contains $\tilde{N} \le C\, N \log(T)$ change-points, where C is a constant.

Therefore, $\tilde{f}$ is L_2-consistent if $N \log^2(T)/T = o(1)$. The key driver behind the L_2 consistency result of Theorem 3.1 is the property of our tail-greedy algorithm by which multiple region merges take place at each scale j of the transform. L_2 consistency cannot be guaranteed if only one region merge

takes place at each scale, as is the case in the bottom-up method introduced in Fryzlewicz (2007).

In the idealized case in which f_i were constant for all i and the analyst knew this, they would use the sample mean $\bar{X} = T^{-1}\sum_{t=1}^{T} X_t$ to estimate $f = f_i$, leading to $\|\bar{X} - f\|_T^2 = O_P(T^{-1})$. In the setting of Theorem 3.1, If the number N of change-points did not increase with T, the rate for $\|\tilde{f} - f\|_T^2$, $O(N\log^2(T)/T)$, would be within a square-logarithmic factor of this idealized rate. We now turn to the intermediate estimator $\tilde{\tilde{f}}$.

THEOREM 3.2. *Let X_t follow model (1) and let the errors ε_t be independent and standard normal. Let $\tilde{\tilde{f}}$ be as in formula (7). Fix any $\delta > 0$ and let the thresholding function $\lambda(\cdot,\cdot)$ be as in Theorem 3.1. On the set $\mathcal{A}_{\delta,T}$ (defined in Theorem 3.1), which satisfies $P(\mathcal{A}_{\delta,T}) \to 1$ as $T \to \infty$ for all $\delta > 0$, we have $\|\tilde{\tilde{f}} - f\|_T^2 = O(NT^{-1}\log^2(T))$. Further, using the notation $\eta_0 = 1$, $\eta_{N+1} = T + 1$, the piecewise-constant estimator $\tilde{\tilde{f}}$ contains at most two change-points between each pair (η_i, η_{i+1}) of change-points in f, for $i = 0, \ldots, N$. Therefore the number $\tilde{\tilde{N}}$ of change-points in $\tilde{\tilde{f}}$ satisfies $\tilde{\tilde{N}} \le 2(N+1)$.*

To conclude, we describe the behaviour of the final estimator $\hat{f}$.

THEOREM 3.3. *Let X_t follow model (1) and let the errors ε_t be independent and standard normal. Let $\hat{f}$ be as in formula (8). Denote the number of change-points in $\hat{f}$ by $\hat{N}$ and their locations, in increasing order, by $\hat{\eta}_1, \ldots, \hat{\eta}_{\hat{N}}$. Fix any $\delta > 0$ and let the thresholding function $\lambda(\cdot,\cdot)$ be as in Theorems 3.1 and 3.2. Let the number N of change-points in f be finite. Denoting their locations, in increasing order, by $\eta_1, \ldots, \eta_N$, assume that $\min_{i=1,\ldots,N} |f_{\eta_i} - f_{\eta_i - 1}| \ge \underline{f} > 0$, and $\min_{i=1,\ldots,N+1} |\eta_i - \eta_{i-1}| \ge b_T$ where b_T is such that $\log^2(T) = o(b_T)$. Then, on the set $\mathcal{A}_{\delta,T}$ (defined in Theorem 3.1), which satisfies $P(\mathcal{A}_{\delta,T}) \to 1$ as $T \to \infty$ for all $\delta > 0$, and for T large enough, we have $\hat{N} = N$ and $|\hat{\eta}_i - \eta_i| \le C\log^2(T)$ for all $i = 1, \ldots, N$, where C is a constant.*

The error rate for the location estimators $\hat{\eta}_i$ can be improved by applying the refinement described in item (B) of Section 2 in the supplemental article Fryzlewicz (2017).

The importance of the iid standard normal assumption on the ε_t's in this section is that it enables the result that $P(\mathcal{A}_{\delta,T}) \to 1$ as $T \to \infty$ for all $\delta > 0$, thanks to the result of Lemma 1 in Yao (1988). We relax both the

Gaussianity and the independence of the ε_t's in the supplemental article Fryzlewicz (2017).

4. Practicalities, numerical study and data analysis.

4.1. *Parameter choice and other practicalities.* The discussion of Sections 4.1 and 4.2 applies to ε_t being iid Gaussian. As is commonly done in wavelet function estimation in Gaussian noise (either explicitly, or implicitly by applying suitably scaled thresholds), we divide all input data to the TGUH algorithm by the Median Absolute Deviation estimate of $\mathrm{Var}^{1/2}(\varepsilon_t)$.

To optimise the finite-sample performance of the TGUH estimator $\hat{f}$ (and the implied estimators $\hat{N}$ and $\hat{\eta}_1, \ldots, \hat{\eta}_{\hat{N}}$), we conducted a large-scale simulation study involving random signals drawn in the same way as in Section 4.1 of Fryzlewicz (2014). Our recommendations for the default parameter choices for $\hat{f}$, described below, are based on the outcome of this study.

Post-processing, Stage 1. We found that Stage 1 of the post-processing procedure from Section 2.4 rarely made a difference, in the sense that only in a very small proportion of cases did it lead to the pruning of change-points in $\tilde{f}$, if used with the same threshold $\lambda(\cdot, \cdot)$ as that used in the construction of $\tilde{f}$. When it did result in pruning, this tended to lead to small improvements in the estimation of N and $\eta_1, \ldots, \eta_N$. Overall, we do not issue a strong recommendation as to whether or not Stage 1 should be used, but we disable it in the remainder of the paper, to minimise computation times.

Post-processing, Stage 2. Stage 2 of the post-processing procedure tended to over-prune change-points, in the sense that it frequently removed at least some of those change-points in $\tilde{\tilde{f}}$ which appeared to be the correct and unique estimates of η_i. This happened when the Stage 2 post-processing was applied with the same threshold $\lambda(\cdot, \cdot)$ as that used in the construction of $\tilde{f}$ and $\tilde{\tilde{f}}$. The use of a lower threshold in Stage 2 tended to make the Stage 2 post-processing less aggressive, but this came at the cost of having to choose two different thresholds (one in the construction of and $\tilde{f}$ and $\tilde{\tilde{f}}$, and the other in the Stage 2 post-processing), which added an extra layer of complexity. For this reason, we recommend disabling the Stage 2 post-processing as a default; this is done in the remainder of the paper.

Threshold $\lambda(\cdot, \cdot)$. The results of Theorems 3.1 – 3.3 state consistency when thresholds $\lambda_1(u, v) = 2\{(1+\delta)\log(T)\}^{1/2}$ or $\lambda_2(u, v) = (u^{1/2} + v^{1/2})/(u + v)^{1/2}\{2(1+\delta)\log(T)\}^{1/2}$ are used (observe that $\lambda_2 \leq \lambda_1$). In practice, we found that thresholds even lower than λ_2 worked better (Theorems 3.1 – 3.3 do *not* claim that λ_2 is the lowest threshold that guarantees consistency). In

T	10^3	10^4	10^5	10^6	10^7
Y_T	0.098	0.59	6.12	75.97	1616.93

TABLE 1

Execution times Y_T (in seconds) of the TGUH algorithm coded in pure R, executed on a standard iMac (late 2015) with white noise vectors of length T on input.

the simulation study, we only considered thresholds constant in u, v. With δ fixed at $\delta = 0.01$, and the parameterisation $\lambda = C\{2(1+\delta)\log(T)\}^{1/2}$, we tested C varying from 0.8 to 1.2. The performance appeared to be the best for values of C close to 1, and we therefore recommend $C = 1$ as a suitable default and use it in the remainder of the paper.

The ρ parameter. ρ is a parameter that describes the upper bound on the number of region merges that take place at each scale, as a proportion of the number of distinct regions remaining. Small values of ρ slow down the procedure; large values decrease its adaptivity: $\rho = 1/2$ reduces the TGUH transform to a decomposition similar to the standard non-adaptive Haar wavelet transform. We did not optimise for ρ but used $\rho = 0.01$ in the simulation study. This is also the value used in the remainder of the paper.

The β parameter. The β parameter controls the "balancedness" of the estimated change-points $\hat{\eta}_i$ and is described in refinement (A) in Section 2 of the supplemental article Fryzlewicz (2017). We tested $\beta = 0$ (no control) and $\beta = 0.05$ and found that $\beta = 0.05$ led to improvement in the accuracy of the $\hat{N}$ estimator. We use $\beta = 0.05$ in the remainder and recommend it as a default value of this parameter. Smaller values of β may be more suitable for signals in which change-points are expected to occur e.g. very close to the left or right boundary of the data.

Computation times. With the parameters set as above, we conducted a simulation study in which we numerically investigated the execution times of the TGUH algorithm (on a standard iMac) on white noise vectors of sizes $10^3, 10^4, \ldots, 10^7$. These are listed in Table 1. Denoting the execution times in seconds by Y_T, we then fitted the best-fitting curve $T\log^a(T)$ by regressing $\log(Y_T/T)$ linearly on $\log(\log(T))$ and estimating a by least squares. The resulting value of a was 0.45, therefore our belief is that the "average" computation times for the TGUH algorithm may be closer to $O(T\log^{1/2}(T))$ than to $O(T\log^2(T))$, at least for data within this range of sample sizes.

4.2. *Comparative simulation study.* To test the finite-sample performance of the TGUH algorithm, we compare it against what we believe is the state of the art in multiple change-point detection for iid Gaussian noise. We consider the following competing methods: PELT (R package `changepoint` version

2.2.2 published on 2016-10-04, see Killick, Fearnhead and Eckley (2012)), Segmentor3IsBack (S3IB; R package `Segmentor3IsBack` version 2.0 published on 2016-06-30, see Rigaill (2015)), SMUCE (R package `stepR` version 2.0-1 published on 2017-05-19, see Frick, Munk and Sieling (2014)), FDRSeg (R package `FDRSeg` version 1.0-1, available from `http://www.stochastik.math.uni-goettingen.de/fdrs`, accessed on 2017-08-24, see Li, Munk and Sieling (2016)), WBS with the sSIC penalty, WBS with threshold constant $C = 1$, BinSeg with threshold constant $C = 1$ (for all three procedures, see Fryzlewicz (2014) and the R package `wbs`) and TGUH. All the procedures are called with their default parameters; the exact calls for PELT, Segmentor3IsBack are as in Fryzlewicz (2014); the calls for SMUCE and FDRSeg use the `stepFit` and `fdrseg` routines, respectively. In those methods that require it, including TGUH, the standard deviation σ of the noise ε_t is estimated via the Mean Absolute Deviation estimator. PELT and S3IB are algorithms for fast minimisation of penalised minus log-likelihood criteria, and the SMUCE and FDRSeg estimators can also be cast in a penalised framework. BinSeg and WBS are multiscale methods.

Our test models are as follows (the signals in models (1)–(5b) are defined in Appendix B of Fryzlewicz (2014) and shown in the supplemental article Fryzlewicz (2017)): (1) the `blocks` model (length 2048, 11 change-points), (2a) the `fms` model (length 497, 6 change-points), (2b) as in (2a) but with $\sigma = 0.4$, (3) the `mix` model (length 560, 13 change-points), (4a) the `teeth10` model (length 140, 13 change-points), (4b) as in (4a) but with $\sigma = 0.5$, (5a) the `stairs10` model (length 150, 14 change-points), (5b) as in (5a) but with but with $\sigma = 0.4$. Models (1), (2a), (3), (4a) and (5a) were used as test beds in Fryzlewicz (2014). The rationale behind adding models (2b), (4b) and (5b) is that they contain higher noise levels and are therefore more testing than (2a), (4a), (5a). We further investigate the following three models: (6a) the `extreme.teeth.5` signal, defined as oscillating between 0 and 1 with change-points every five observations, with $\sigma = 0.2$ (length 1000, 199 change-points); (6b) the `extreme.teeth.10` signal, defined as oscillating between 0 and 1 with change-points every ten observations, with $\sigma = 0.35$ (length 1000, 99 change-points); (6c) the `extreme.teeth.20` signal, defined as oscillating between 0 and 1 with change-points every twenty observations, with $\sigma = 0.5$ (length 1000, 49 change-points). Models (6a)-(6c) are designed to test the performance of the competitors for signals with large numbers of frequently occurring change-points.

We set the random seed to 1 before running 100 simulations for each of the test signals and each of the competitors. The performance metrics are: some aspects of the distribution of $\hat{N} - N$, and the Mean-Square Error of $\hat{f}_t$

in estimating f_t (for each method, $\hat{f}_t$ is constructed as a piecewise-constant function whose value between each pair of consecutive estimated change-points is the mean of the data over the interval delimited by this pair of change-points). Tables 2 and 3 show the results. We now briefly describe the performance of each method in turn.

PELT appears to systematically underestimate the number of change-points, which is at least partly due to the default penalty used in version 2.2.2 of the `changepoint` package, which is MBIC. (However, in our experience it is rare for any penalized method to offer good performance for *any* particular penalty across a wide range of signals.)

S3IB offers very good or acceptable performance in models (1)–(4b), but not so in models (5a), (5b), in which it never estimates the right number of change-points (despite the number of change-points in these models not exceeding the upper bound considered by S3IB, which is 14). In models (6a)–(6c), S3IB consistently estimates 0, rather than 14 change-points.

SMUCE appears to systematically underestimate N.

FDRSeg is a very good performer throughout, even though as far as the estimation of the number of change-points is concerned, it appears to be significantly behind the best method in many of the models. In models (6a)–(6c), it is the only method besides TGUH that is able to detect the right number of change-points in at least some simulations. We also comment on the execution speed of FDRSeg. When called with its default parameters on a signal whose length exceeds the length of previously processed signals, the routine `fdrseg` runs a Monte Carlo simulation, which appears to be costly. For example, the first execution of `fdrseg` on a signal of length 10^4 took over 400 seconds on a standard iMac (late 2015). We attempted to input a signal of length 10^5 but interrupted the execution after 25 minutes. This has to be contrasted with the execution times for TGUH reported in Table 1.

WBS sSIC offers excellent performance for models (1)–(5b), which contain relatively small numbers of change-points with (mostly) large separation between them. It understandably performs poorly for models (6a)–(6c); this is due both to the sSIC penalty generally struggling for models with many change-points and to the default number of random intervals used being 5000, which is insufficient in these models.

WBS with threshold constant $C = 1$ has the tendency to overestimate the true number of change-points in most of the models (1)–(5b). However, it underestimates the true number of change-points in models (6a)–(6c). This is due to the default number of random intervals drawn ($M = 5000$) in the WBS algorithm being insufficient for these change-point-rich signals.

We also ran the WBS method with threshold constant $C = 1$ on models (6a)–(6c) but with the number of intervals increased to $M = 200000$. This resulted in the number of change-points being correctly estimated in 17, 44, 50 simulations out of 100, for models (6a), (6b), (6c), respectively; this came at the cost of increased computation time. However, the increase in M did not induce good performance in models (1)–(5b).

BinSeg with threshold constant $C = 1$ performs poorly for many (but not all) of the models.

TGUH offers excellent performance in all of the models and it is either the best of not far behind the best performer in each model in terms of the accurate estimation of N. In particular, in models (6a)–(6c), it is the only method besides FDRSeg that is able to detect the right number of change-points in at least some simulations. It is interesting to observe that e.g. for the `blocks` model, the MSE of TGUH appears to be closer to BinSeg than, say, to WBS. This is unsuprising given that TGUH and BinSeg have similarly low upper bounds on their computational complexities (which are guaranteed irrespective of the signal), which cannot be said of WBS. However, we also recall the remark in item (B) of Section 2 of the supplemental article Fryzlewicz (2017) regarding the possibility of improving (in any multiple change-point estimation technique including TGUH) the accuracy of estimates $\hat{\eta}_i$ by iterative re-estimation. Finally, the computational speed of the TGUH algorithm is independent of the input signal, and in particular of the number of its change-points. The same cannot be said of e.g. WBS, in which, as mentioned earlier, more intervals need to be drawn for the method to perform well on signals with numerous change-points.

To summarise, TGUH emerges as an overall winner, which combines a high degree of accuracy with low computational complexity independent of the input signal. Its performance is particularly impressive in signals with frequent change-points. In Section 3 of the supplemental article Fryzlewicz (2017), we further investigate the performance of the TGUH method in estimating the locations η_i of the true change-points in models (1)–(5b).

4.3. *Data example.* We analyse monthly percentage changes in the UK's Land Registry House Price Index (HPI), from January 1995 to December 2015, in three east London boroughs: Hackney, Newham and Tower Hamlets. The HPI provides an overall measure of completed house sale transactions, and the methodology used in its computation is available from the Land Registry website. The data, accessed in February 2016, are available from `http://landregistry.data.gov.uk/app/hpi`. Hackney and Tower Hamlets are located more centrally than Newham, and both border on the City

		$\hat{N}-N$					
Method	Model	<0	0	>0	$\hat{E}(\hat{N}-N)$	$\hat{\mathrm{Var}}(\hat{N}-N)$	MSE
PELT		88	12	0	-1.37	0.58	3.32
S3IB		52	45	3	-0.57	0.47	2.44
SMUCE		96	4	0	-1.17	0.24	3.15
FDRSeg	(1)	37	49	14	-0.18	0.65	2.51
WBS sSIC		56	42	2	-0.56	0.39	2.47
WBS $C=1$		30	33	37	0.38	1.89	2.59
BinSeg $C=1$		34	45	21	-0.11	0.66	3.14
TGUH		43	44	13	-0.32	0.70	3.21
PELT		29	71	0	-0.61	0.97	646×10^{-5}
S3IB		0	86	14	0.19	0.26	414×10^{-5}
SMUCE		28	67	5	-0.26	0.36	611×10^{-5}
FDRSeg	(2a)	3	80	17	0.24	0.53	446×10^{-5}
WBS sSIC		3	93	4	0.01	0.13	427×10^{-5}
WBS $C=1$		0	39	61	1.34	2.63	528×10^{-5}
BinSeg $C=1$		36	45	19	-0.16	0.62	792×10^{-5}
TGUH		2	84	14	0.21	0.47	507×10^{-5}
PELT		87	13	0	-2.07	1.03	160×10^{-4}
S3IB		26	64	10	-0.35	1.02	93×10^{-4}
SMUCE		82	17	1	-1.16	0.62	138×10^{-4}
FDRSeg	(2b)	40	47	13	-0.44	1.36	114×10^{-4}
WBS sSIC		36	63	1	-0.66	1.03	104×10^{-4}
WBS $C=1$		9	35	56	1.19	3.31	109×10^{-4}
BinSeg $C=1$		68	24	8	-0.88	0.93	137×10^{-4}
TGUH		26	61	13	-0.18	1.3	113×10^{-4}
PELT		99	1	0	-3.28	1.11	2.26
S3IB		88	12	0	-2.10	1.46	1.73
SMUCE		93	7	0	-1.54	0.61	1.78
FDRSeg	(3)	69	25	6	-1.00	1.41	1.60
WBS sSIC		69	28	3	-1.18	1.38	1.64
WBS $C=1$		27	33	40	0.36	1.75	1.70
BinSeg $C=1$		72	20	8	-1.12	1.3	2.21
TGUH		57	38	5	-1.01	1.65	1.86
PELT		90	10	0	-8.26	18.9	186×10^{-3}
S3IB		52	48	0	-4.10	28.68	114×10^{-3}
SMUCE		96	4	0	-5.39	7.09	189×10^{-3}
FDRSeg	(4a)	42	45	13	-1.56	10.19	87×10^{-3}
WBS sSIC		13	73	14	-0.28	3.37	59×10^{-3}
WBS $C=1$		12	67	21	0.12	0.55	55×10^{-3}
BinSeg $C=1$		84	9	7	-2.99	7.91	136×10^{-3}
TGUH		26	68	6	-0.43	1.54	68×10^{-3}
PELT		98	2	0	-10.73	7.98	0.23
S3IB		84	16	0	-8.22	29.14	0.2
SMUCE		100	0	0	-8.19	7.04	0.23
FDRSeg	(4b)	80	14	6	-5.25	22.19	0.17
WBS sSIC		67	23	10	-5.56	27.78	0.17
WBS $C=1$		47	33	20	-0.80	3.35	0.11
BinSeg $C=1$		97	1	2	-6.45	10.25	0.2
TGUH		69	26	5	-2.57	8.07	0.13

TABLE 2

Aspects of the distribution of $\hat{N}-N$ for the various methods and models, over 100 simulations. Also the average Mean-Square Error of the resulting estimate of f_t.

		$\hat{N}-N$					
Method	Model	<0	0	>0	$\hat{E}(\hat{N}-N)$	$\hat{\mathrm{Var}}(\hat{N}-N)$	MSE
PELT		12	87	1	-0.13	0.17	25×10^{-3}
S3IB		100	0	0	-5.72	1.64	210×10^{-3}
SMUCE		75	25	0	-1.90	1.99	98×10^{-3}
FDRSeg	(5a)	0	79	21	0.30	0.49	22×10^{-3}
WBS sSIC		0	57	43	0.59	0.67	25×10^{-3}
WBS $C=1$		0	62	38	0.52	0.64	26×10^{-3}
BinSeg $C=1$		1	73	26	0.28	0.28	27×10^{-3}
TGUH		0	92	8	0.10	0.15	23×10^{-3}
PELT		92	7	1	-2.61	2.26	124×10^{-3}
S3IB		100	0	0	-4.45	2.25	184×10^{-3}
SMUCE		100	0	0	-4.71	1.66	204×10^{-3}
FDRSeg	(5b)	47	38	15	-0.55	1.3	74×10^{-3}
WBS sSIC		14	56	30	0.25	0.82	60×10^{-3}
WBS $C=1$		21	49	30	0.16	1.02	63×10^{-3}
BinSeg $C=1$		33	56	11	-0.29	0.71	66×10^{-3}
TGUH		38	53	9	-0.42	1.15	70×10^{-3}
PELT		100	0	0	-178.46	712.65	228×10^{-3}
S3IB		100	0	0	-199.00	0	250×10^{-3}
SMUCE		100	0	0	-117.48	188.39	230×10^{-3}
FDRSeg	(6a)	3	42	55	0.95	1.42	11×10^{-3}
WBS sSIC		100	0	0	-198.00	0	250×10^{-3}
WBS $C=1$		100	0	0	-114.47	40.64	208×10^{-3}
BinSeg $C=1$		100	0	0	-52.86	231.33	124×10^{-3}
TGUH		31	68	1	-0.72	1.58	13×10^{-3}
PELT		100	0	0	-90.77	120.74	236×10^{-3}
S3IB		100	0	0	-99.00	0	250×10^{-3}
SMUCE		100	0	0	-54.10	50.01	225×10^{-3}
FDRSeg	(6b)	34	33	33	-0.12	2.01	38×10^{-3}
WBS sSIC		100	0	0	-97.88	0.1	250×10^{-3}
WBS $C=1$		100	0	0	-34.06	18.84	164×10^{-3}
BinSeg $C=1$		100	0	0	-33.71	72.37	162×10^{-3}
TGUH		68	31	1	-2.28	6.36	46×10^{-3}
PELT		100	0	0	-37.53	89.65	209×10^{-3}
S3IB		100	0	0	-49.00	0	250×10^{-3}
SMUCE		100	0	0	-17.24	13.76	195×10^{-3}
FDRSeg	(6c)	20	47	33	0.10	1.65	51×10^{-3}
WBS sSIC		100	0	0	-46.96	5.25	247×10^{-3}
WBS $C=1$		96	3	1	-3.55	4.09	77×10^{-3}
BinSeg $C=1$		100	0	0	-10.64	17.32	137×10^{-3}
TGUH		28	64	8	-0.59	2.18	58×10^{-3}

TABLE 3

Aspects of the distribution of $\hat{N}-N$ for the various methods and models, over 100 simulations. Also the average Mean-Square Error of the resulting estimate of f_t.

of London, a major business and financial centre. In addition, the second major financial centre, Canary Wharf, is located within the borough of Tower Hamlets. Newham is located to the east of Hackney and Tower Hamlets, and was the main host borough of the London 2012 Olympic Games. Our analysis comes in two parts.

Part I. In Part I, we treat the data as coming from a "piecewise constant mean + iid Gaussian noise" model and apply our TGUH method with default parameters to the raw data. Figure 1 shows the three estimates (from January 2008 to December 2015) superimposed.

It is interesting to observe that for extended periods of time, the HPI increases in the borough of Newham appeared to trail those in the other two boroughs, despite the large-scale investment in the borough of Newham related to the London 2012 Olympic Games. In particular, this is clearly seen in Figure 1 for the time period between June 2011 and August 2013, in which (except the strong positive spike in August 2012), the average HPI increase in Newham is negative. One commentary in the Guardian, a national newspaper, attributes this to what it sees as an over-supply of new properties in the borough (https://www.theguardian.com/money/2012/mar/13/olympics-house-prices-boom-fails).

Curiously, the situation is reversed in the more recent time period November 2014 to December 2015, in which Newham shows the strongest increases in the HPI among the three boroughs. Interestingly, some newspapers and online news sources speculate that this may have been due not only to the regeneration taking place in the borough of Newham, but also to some buyers having been priced out of the more central (and expensive) boroughs of Hackney and Tower Hamlets, as suggested e.g. in The Wharf, a Canary Wharf newspaper (http://www.wharf.co.uk/news/property/newham-shows-steepest-house-price-9160342) and in The Guardian (https://www.theguardian.com/money/2015/dec/28/newham-east-london-sees-uks-biggest-house-price-rise).

Part II. In Part II, we first "prewhiten" the data by fitting univariate time series models to each time series separately, and then apply the TGUH method with default parameters to the residual sequences from these fits. To be more precise, we obtain these residuals as follows: we first fit univariate AR models to the data with the AR coefficients estimated using Yule-Walker and the orders chosen via the AIC, up to the maximum order of 12; we then fit GARCH(1,1) models via Gaussian MLE to the residuals from the AR fits, and obtain residuals from the combined AR+GARCH fits.

The results are shown in Figure 2. The TGUH fit to the AR+GARCH

residuals reveals a completely different picture to the fit from Part I. In Figure 2, the period from the end of the financial crisis onwards (i.e., roughly speaking, to the right of the green line), is uneventful for the three estimates, with the exception of the possibly spurious feature towards the end of the data for the TGUH estimate for the borough of Newham. However, an interesting comparison of the three estimates can be made to the left of the green line (i.e. during the financial crisis). Once the time series effect has been removed, the TGUH estimator indicates that the residuals for Tower Hamlets and Hackney were strongly negative in the mean for much of this period, which can be interpreted as house prices in these two boroughs being judged to have dropped significantly during the financial crisis, and that this drop was not merely due to a "random" fluctuation of a stationary AR+GARCH process. By contrast, no such dip was observed for the borough of Newham. Figure 6 in the supplemental article Fryzlewicz (2017) shows how this conclusion may also be supported by the raw data; for example, the time series for Newham exbitis far more zero-crossings than the other two series in the period January 2008 – December 2009, and therefore it is more difficult to classify it as "firmly in the negative territory, followed by firmly in the positive territory" than the other two series.

It is, of course, difficult to make recommendations as to which of the two analyses, that in Part I or that in Part II, should be preferred, without a specific objective function in mind. We refer the reader to Robbins et al. (2011) for an interesting review of typical issues faced by the change-point analyst in a time series versus iid noise setting.

5. Discussion. The TGUH approach only fundamentally relies on each merged region having a neighbour, so it can easily be generalised, at least algorithmically, to other, more complex data structures that include the notion of a local neighbourhood. Examples include graphs and networks, images, videos, nonlinear manifolds in $\mathbb{R}^k$ and even unstructured data, provided that a neighbourhood structure can be constructed on them, via e.g. Minimum Spanning Trees. This makes TGUH potentially applicable to other "change-point-like" problems such as e.g. community detection in networks or image segmentation. A version of the TGUH method for images was proposed, without the associated theory, in Fryzlewicz and Timmermans (2016). This is in contrast to top-down change-point detection methods, which are typically hard or impossible to generalise to more complex data structures.

An exciting aspect of TGUH is that its extensions can be built that consider not pairs of neighbours to merge at each stage, but triplets or, more generally, k-tuples. Just as the TGUH algorithm presented in this paper can

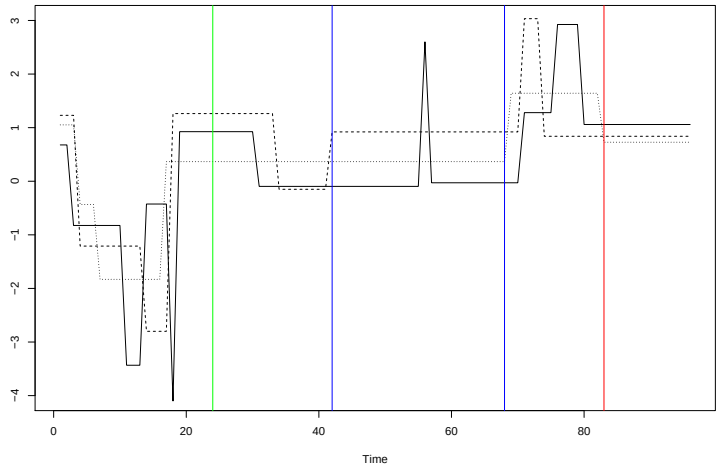

FIG 1. *TGUH estimates for Newham (solid), Hackney (dashed) and Tower Hamlets (dotted), from January 2008 to December 2015. Green line: December 2009; blue lines: June 2011 and August 2013; red line: November 2014.*

be used in detecting change-points in the constant mean (=polynomial of degree 0), our expectation is that a more general TGUH algorithm involving k-tuples could possibly be used in detecting change-points in a piecewise-polynomial model of degree $\leq k-2$. Fryzlewicz (2007) presents a heuristic algorithm akin to the Unbalanced Haar transform but involving triplets of neighbours – however, it comes without the tail-greedy feature and is slow.

APPENDIX A: PROOFS OF MAIN THEORETICAL RESULTS

LEMMA A.1. *Let* $\mathcal{S}_j^1 = \{1 \leq k \leq K(j) : d_{p,q,r}^{(j,k)} \text{ is such that } p < \eta_i - \frac{1}{2} < r \text{ for some } i = 1,\dots,N\}$, *and* $\mathcal{S}_j^0 = \{1,\dots,K(j)\} \setminus \mathcal{S}_j^1$. *On* $\mathcal{A}_{\delta,T}$, *for* $j = 1,\dots,J$, $k \in \mathcal{S}_j^0$, *we have* $|d^{(j,k)}| \leq \lambda_2$, *and hence also* $|d^{(j,k)}| \leq \lambda_1$, *where* λ_1, λ_2 *are as in the statement of Theorem 3.1.*

Proof. On $\mathcal{A}_{\delta,T}$, for $j = 1,\dots,J$, $k \in \mathcal{S}_j^0$, we have

$$\begin{aligned} |d_{p,q,r}^{(j,k)}| &= \left| \left\{ \frac{r-q}{r-p+1} \right\}^{1/2} \frac{\sum_{t=p}^{q} \varepsilon_t}{(q-p+1)^{1/2}} - \left\{ \frac{q-p+1}{r-p+1} \right\}^{1/2} \frac{\sum_{t=q+1}^{r} \varepsilon_t}{(r-q)^{1/2}} \right| \\ &\leq \{2(1+\delta)\log(T)\}^{1/2} \frac{(r-q)^{1/2} + (q-p+1)^{1/2}}{(r-p+1)^{1/2}}, \end{aligned}$$

which completes the proof.

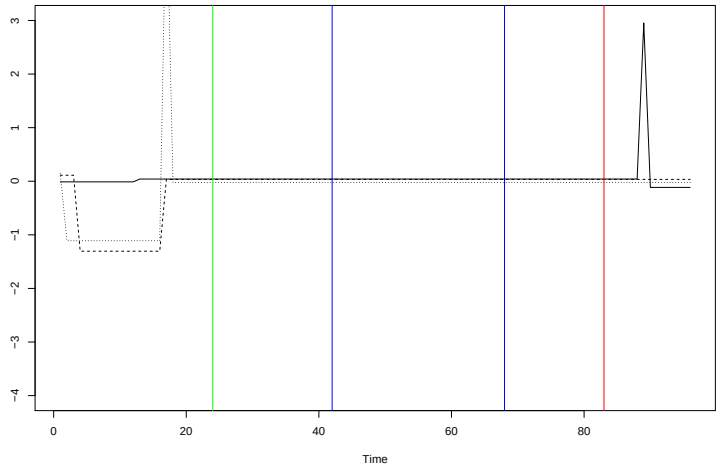

FIG 2. *TGUH estimates for Newham (solid), Hackney (dashed) and Tower Hamlets (dotted), with time series residuals as input, from January 2008 to December 2015. Green line: December 2009; blue lines: June 2011 and August 2013; red line: November 2014.*

Proof of Theorem 3.1. The fact that $P(\mathcal{A}_{\delta,T}) \to 1$ for all $\delta > 0$ is the statement of Lemma 1 in Yao (1988). Let $\lambda_\cdot$ denote either of λ_1 and λ_2. Let $\mathcal{S}_j^1 = \{1 \le k \le K(j) : d_{p,q,r}^{(j,k)}$ is such that $p < \eta_i - \frac{1}{2} < r$ for some $i = 1, \ldots, N\}$, and $\mathcal{S}_j^0 = \{1, \ldots, K(j)\} \setminus \mathcal{S}_j^1$. Due to the conditional orthonormality of the Unbalanced Haar transform, on set $\mathcal{A}_{\delta,T}$, we have

$$\begin{aligned}
\|\tilde{f} - f\|_T^2 &= T^{-1} \sum_{j=1}^{J} \sum_{k=1}^{K(j)} (d^{(j,k)}\, \mathbb{I}\{\exists (j',k') \in \mathcal{C}_{j,k}\ |d^{(j',k')}| > \lambda_\cdot\} - \mu^{(j,k)})^2 \\
&+ T^{-1}(s_{1,T} - \mu^{(0,0)})^2 \\
&\le T^{-1} \sum_{j=1}^{J} \left(\sum_{k \in \mathcal{S}_j^0} + \sum_{k \in \mathcal{S}_j^1} \right) (d^{(j,k)}\, \mathbb{I}\{\exists (j',k') \in \mathcal{C}_{j,k}\ |d^{(j',k')}| > \lambda_\cdot\} - \mu^{(j,k)})^2 \\
&+ 2(1+\delta)\, T^{-1} \log(T) =: I + II + 2(1+\delta)\, T^{-1} \log(T).
\end{aligned}$$

Turning first to I, recall that $\mu^{(j,k)} = 0$ for $k \in \mathcal{S}_j^0$ and, by Lemma A.1, $\mathbb{I}\{\exists (j',k') \in \mathcal{C}_{j,k}\ |d^{(j',k')}| > \lambda_\cdot\} = 0$ for $k \in \mathcal{S}_j^0$; therefore $I = 0$. In consider-

ing II, we denote $\mathcal{B} = \{\exists (j',k') \in \mathcal{C}_{j,k} \ \ |d^{(j',k')}| > \lambda_.\}$ and first compute

$$\begin{aligned}
(d^{(j,k)}\, \mathbb{I}\{\mathcal{B}\} - \mu^{(j,k)})^2 &= (d^{(j,k)}\, \mathbb{I}\{\mathcal{B}\} - d^{(j,k)} + d^{(j,k)} - \mu^{(j,k)})^2 \\
&\leq (d^{(j,k)})^2 \mathbb{I}(|d^{(j,k)}| \leq \lambda_.) + 2|d^{(j,k)}|\mathbb{I}(|d^{(j,k)}| \leq \lambda_.)|d^{(j,k)} - \mu^{(j,k)}| \\
&+ \{d^{(j,k)} - \mu^{(j,k)}\}^2 \\
&\leq \lambda_.^2 + 2\lambda_.\{2(1+\delta)\log(T)\}^{1/2} + 2(1+\delta)\log(T) \\
&\leq (6+4\sqrt{2})(1+\delta)\log(T). \qquad (9)
\end{aligned}$$

Combining this with the fact that $J \leq \lceil \log(T)/\log\{(1-\rho)^{-1}\}\rceil$, $|\mathcal{S}_j^1| \leq N$, we obtain $II \leq (6+4\sqrt{2})(1+\delta)N\, T^{-1}\lceil \log(T)/\log\{(1-\rho)^{-1}\}\rceil \log(T)$, and hence $\|\tilde{f} - f\|_T^2 \leq 2(1+\delta)\, T^{-1}\log(T)\left\{1 + (3+2\sqrt{2})N\lceil \log(T)/\log\{(1-\rho)^{-1}\}\rceil\right\}$. Also, estimated change-points are produced in $\tilde{f}$ only by those coefficients $d^{(j,k)}$ that are computed over those portions of the data that contain true change-points in f. Therefore, at each scale, we add up to N estimated change-points to $\tilde{f}$. With the number J of scales bounded by $C\,\log(T)$, we obtain that $\tilde{f}$ contains $\tilde{N} \leq C\,N\,\log(T)$ estimated change-points, which completes the proof of the theorem.

Proof of Theorem 3.2. We first establish that $\tilde{\tilde{f}}$ can be interpreted as the result of (a) decomposing the data X with respect to a particular UH basis (possibly different from that used in $\tilde{f}$ but sharing some of its basis vectors), (b) connected thresholding of this decomposition with threshold $\lambda(\cdot,\cdot)$, and (c) inverse UH transformation.

Let $\tilde{B}$ and $\tilde{\tilde{B}}$ (the latter to be constructed) denote the UH bases corresponding to $\tilde{f}$ and $\tilde{\tilde{f}}$, respectively. As the set of change-points in $\tilde{\tilde{f}}$ is a subset of that in $\tilde{f}$, we first include in $\tilde{\tilde{B}}$ all those vectors $\psi^{(j,k)} \in \tilde{B}$ for which the coefficients $d_{p,q,r}^{(j,k)} = \langle X, \psi^{(j,k)}\rangle$ are such that $|d_{p,q,r}^{(j,k)}| < \lambda(q-p+1, r-q)$, as these basis vectors $\psi^{(j,k)}$ are not going to introduce any change-points in $\tilde{\tilde{f}}$.

We then include in $\tilde{\tilde{B}}$ those basis functions $\psi^{(j,1)}$ for which $d_{p,q,r}^{(j,1)} = \langle X, \psi^{(j,1)}\rangle$ are the detail coefficients produced in Stage 1 of the post-processing algorithm of Section 2.4; we include all those for which $|d_{p,q,r}^{(j,1)}| < \lambda(q-p+1, r-q)$ as well as the *first* one for which $|d_{p,q,r}^{(j,1)}| \geq \lambda(q-p+1, r-q)$. Denote the scale j for this latter coefficient by J_0. Suppose the computation of the detail coefficients $d^{(j,1)}$ continues beyond scale J_0 according to the recipe outlined in Stage 1 of the post-processing algorithm of Section 2.4.

By its construction, $\tilde{\tilde{f}} = \sum_{j \geq J_0} d^{(j,1)}\psi^{(j,1)} + s_{1,T}\psi^{(0,0)}$. We now demonstrate that the coefficients $\{d^{(j,1)}\}_{j>J_0}$ must be retained by the connected thresholding. This will certainly happen, *since only one detail coefficient is*

produced at each scale j. Therefore, the detail coefficients $d^{(j,1)}$ for $j > J_0$ either (a) are parent coefficients of $d^{(J_0,1)}$ or (b) satisfy $|d^{(j,1)}| > |d^{(J_0,1)}|$, and hence will be retained by the connected thresholding in either case.

We now evaluate the total number of scales needed to carry out this particular UH transform to completion. The basis vectors $\psi^{(j,k)}$ added to $\tilde{\tilde{B}}$ at the beginning are inherited from $\tilde{B}$ and therefore the corresponding coefficients $d^{(j,k)}$ live on no more than J scales. Since $\tilde{N} \leq CN\log(T)$, and the remaining coefficients $d^{(j,1)}$ each correspond to a different change-point in $\tilde{f}$, there are at most $\tilde{N}$ coefficients $d^{(j,1)}$ (and hence at most $\tilde{N}$ additional scales). Since $J = O(\log(T))$ and $\tilde{N} = O(\log(T))$, the total number of scales J_1 required by the complete transform is $J_1 = O(\log(T))$.

We are now in a position to show the L_2 result, which is obtained in analogy to that in Theorem 3.1 since both $\tilde{f}$ and $\tilde{\tilde{f}}$ are estimators involving UH transforms with logarithmic number of scales and connected thresholding, which is all that is needed for L_2 consistency with the rate as in Theorem 3.1. Re-examining the proof of Theorem 3.1, the equivalent of quantity II in that proof for $\tilde{\tilde{f}}$ is bounded by $II \leq (6+4\sqrt{2})(1+\delta)N\,T^{-1}J_1\log(T)$, which implies $\|\tilde{\tilde{f}} - f\|_T^2 = O(NT^{-1}\log^2(T))$.

Finally, we show that there are at most two change-points in $\tilde{\tilde{f}}$ between each pair of true change-points (η_i, η_{i+1}) for $i = 0, \ldots, N$ (we use the convention $\eta_0 = 1$, $\eta_{N+1} = T+1$). Indeed, the post-processing algorithm described in Stage 1 of Section 2.4 would not be able to terminate with $\tilde{\tilde{f}}$ containing three change-points, say $\tilde{\tilde{\eta}}_l, \tilde{\tilde{\eta}}_{l+1}, \tilde{\tilde{\eta}}_{l+2}$, between η_i, η_{i+1}, as the next $d^{(j,1)}$ coefficient to be computed would be either $d^{(j,1)}_{\tilde{\tilde{\eta}}_l, \tilde{\tilde{\eta}}_{l+1}-1, \tilde{\tilde{\eta}}_{l+2}-1}$, or a coefficient smaller than $|d^{(j,1)}_{\tilde{\tilde{\eta}}_l, \tilde{\tilde{\eta}}_{l+1}-1, \tilde{\tilde{\eta}}_{l+2}-1}|$ in absolute value, which by Lemma A.1 would not exceed $\lambda(\tilde{\tilde{\eta}}_{l+1} - \tilde{\tilde{\eta}}_l, \tilde{\tilde{\eta}}_{l+2} - \tilde{\tilde{\eta}}_{l+1})$ and therefore the algorithm would continue. This completes the proof.

Proof of Theorem 3.3. From Theorem 3.2, there must exist a constant C such that for T large enough, there must be at least one estimated change-point $\tilde{\tilde{\eta}}_l$ within the distance of $C\log^2(T)$ from each true change-point η_i, $i = 1, \ldots, N$. Indeed, if it were not true, it would not be possible to achieve the rate for $\|\tilde{\tilde{f}} - f\|_T^2$ specified in Theorem 3.2. For each i_0 from Stage 2 of our post-processing algorithm of Section 2.4, either of two cases are possible:

1. $\tilde{\tilde{\eta}}_{i_0}$ is not the closest estimated change-point to either the nearest true change-point on its left-hand side, or the nearest true change-point on its right-hand side. Then, by the construction of d_{p_i,q_i,r_i}, Lemma A.1 guarantees that $|d_{p_{i_0},q_{i_0},r_{i_0}}| < \lambda(q_{i_0} - p_{i_0} + 1, r_{i_0} - q_{i_0})$ and $\tilde{\tilde{\eta}}_{i_0}$ gets

removed.

2. $\tilde{\tilde{\eta}}_{i_0}$ is the closest estimated change-point to a true change-point η_i, and is therefore within the distance of $C\log^2(T)$ from η_i. If it so happens that $|d_{p_{i_0},q_{i_0},r_{i_0}}| < \lambda(q_{i_0} - p_{i_0} + 1, r_{i_0} - q_{i_0})$ and therefore $\tilde{\tilde{\eta}}_{i_0}$ gets removed, it must be true that there is another $\tilde{\tilde{\eta}}_j$ within the distance of $C_{i_0}\log^2(T)$ from $\tilde{\tilde{\eta}}_{i_0}$ (and therefore also from η_i), where C_{i_0} is a constant. Indeed, if there were no such $\tilde{\tilde{\eta}}_j$ on either side of $\tilde{\tilde{\eta}}_{i_0}$, then by the construction of d_{p_i,q_i,r_i} (formula (2)), the order of magnitude of $d_{p_{i_0},q_{i_0},r_{i_0}}$ would be such that $|d_{p_{i_0},q_{i_0},r_{i_0}}| > \lambda(q_{i_0} - p_{i_0} + 1, r_{i_0} - q_{i_0})$ and $\tilde{\tilde{\eta}}_{i_0}$ would not get removed.

Since there are at most $\tilde{\tilde{N}}$ removals with $\tilde{\tilde{N}}$ being a finite number by Theorem 3.2, the constants C_i must be bounded from above by constant C'. Therefore, by case 2 above, after the algorithm has terminated, each true change-point η_i must have an associated estimated change-point $\hat{\eta}_i$ within the distance of $C'\log^2(T)$. It must also have only one such change-point as if it had two, the more remote of them would not be the closest estimated change-point to any true change-point, and case 1 would apply. This completes the proof.

SUPPLEMENTARY MATERIAL

Supplement A: Supplement to "Tail-greedy bottom-up data decompositions and fast multiple change-point detection" (doi: COMPLETED BY THE TYPESETTER; .pdf). Extension of the TGUH methodology to dependent non-Gaussian data; refinements to post-processing; study of the accuracy of TGUH in estimating change-point locations; additional figure for the analysis of Section 4.3.

REFERENCES

Aggarwal, R., Inclan, C. and Leal, R. (1999). Volatility in Emerging Stock Markets. *Journal of Financial and Quantitative Analysis* **34** 33–55.

Auger, I. and Lawrence, C. (1989). Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology* **51** 39-54.

Bai, J. (1997). Estimating multiple breaks one at a time. *Econometric Theory* **13** 315–352.

Bai, J. and Perron, P. (2003). Computation and Analysis of Multiple Structural Change Models. *Journal of Applied Econometrics* **18** 1–22.

Baranowski, R., Chen, Y. and Fryzlewicz, P. (2016). Narrowest-Over-Threshold detection of multiple change-points and change-point-like features. *Preprint*.

Birge, L. and Massart, P. (2001). Gaussian model selection. *J. European Math. Soc.* **3** 203-268.

Boysen, L., Kempe, A., Liebscher, V., Munk, A. and Wittich, O. (2009). Consistencies and rates of convergence of jump-penalized least squares estimators. *Annals of Statistics* **37** 157-183.

Braun, J., Braun, R. and Mueller, H. G. (2000). Multiple changepoint fitting via quasilikelihood, with application to DNA sequence segmentation. *Biometrika* **87** 301–314.

Braun, J. and Mueller, H. G. (1998). Statistical Methods for DNA Sequence Segmentation. *Statistical Science* **13** 142–162.

Brodsky, B. and Darkhovsky, B. (1993). *Nonparametric Methods in Change-Point Problems*. Kluwer Academic Publishers.

Chen, K. M., Cohen, A. and Sackrowitz, H. (2011). Consistent multiple testing for change points. *Journal of Multivariate Analysis* **102** 1339-1343.

Cho, H. and Fryzlewicz, P. (2011). Multiscale interpretation of taut string estimation and its connection to Unbalanced Haar wavelets. *Statistics and Computing* **21** 671–681.

Cho, H. and Fryzlewicz, P. (2012). Multiscale and multilevel technique for consistent segmentation of nonstationary time series. *Statistica Sinica* **22** 207–229.

Cho, H. and Fryzlewicz, P. (2015). Multiple change-point detection for high-dimensional time series via Sparsified Binary Segmentation. *Journal of the Royal Statistical Society Series B* **77** 475–507.

Choi, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In *NAACL 2000 Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference* 26–33.

Chu, P. S. and Zhao, X. (2004). Bayesian Change-Point Analysis of Tropical Cyclone Activity: The Central North Pacific Case. *Journal of Climate* **17** 4893–4901.

Ciuperca, G. (2011). A general criterion to determine the number of change-points. *Statistics & Probability Letters* **81** 1267-1275.

Ciuperca, G. (2014). Model selection by LASSO methods in a change-point model. *Statistical Papers* **55** 349–374.

Davies, P. L. and Kovac, A. (2001). Local extremes, runs, strings and multiresolution. *Ann. Statist.* **29** 1-48.

Davis, R., Lee, T. and Rodriguez-Yam, G. (2006). Structural Break Estimation for Nonstationary Time Series Models. *Journal of the American Statistical Association* **101** 223-239.

Desmond, R., Weiss, H., Arani, R., Soong, S. J., Wood, M., Fiddian, P., Gnann, J. and Whitley, R. (2002). Clinical Applications for Change-Point Analysis of Herpes Zoster Pain. *Journal of Pain and Symptom Management* **23** 510–516.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81** 425–455.

Du, C., Kao, C. L. and Kou, S. (2016). Stepwise Signal Extraction via Marginal Likelihood. *Journal of the American Statistical Association* **111** 314–330.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics* **32** 407–499.

Eichinger, B. and Kirch, C. (2018). A MOSUM procedure for the estimation of multiple random change points. *Bernoulli* **24** 526–564.

Erdman, C. and Emerson, J. (2008). A fast Bayesian change point analysis for the segmentation of microarray data. *Bioinformatics* **24** 2143–2148.

Frick, K., Munk, A. and Sieling, H. (2014). Multiscale change-point inference (with discussion). *Journal of the Royal Statistical Society Series B* **76** 495–580.

Fryzlewicz, P. (2007). Unbalanced Haar technique for nonparametric function estimation. *J. Amer. Stat. Assoc.* **102** 1318–1327.

Fryzlewicz, P. (2014). Wild Binary Segmentation for multiple change-point detection. *Annals of Statistics* **42** 2243–2281.

Fryzlewicz, P. (2017). Supplement to "Tail-greedy bottom-up data decompositions and

fast multiple change-point detection". *Preprint*.

Fryzlewicz, P. and Subba Rao, S. (2014). Multiple-change-point detection for auto-regressive conditional heteroscedastic processes. *Journal of the Royal Statistical Society Series B* **76** 903–924.

Fryzlewicz, P. and Timmermans, C. (2016). SHAH: SHape-Adaptive Haar wavelets for image processing. *Journal of Computational and Graphical Statistics* **25** 879898.

Harchaoui, Z. and Lévy-Leduc, C. (2010). Multiple Change-Point Estimation With a Total Variation Penalty. *Journal of the American Statistical Association* **105** 1480–1493.

Huskova, M. and Slaby, A. (2001). Permutation tests for multiple changes. *Kybernetika* **37** 605–622.

Jackson, B., Sargle, J., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L. and Tsai, T. T. (2005). An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters* **12** 105-108.

Jansen, M., Nason, G. and Silverman, B. (2009). Multiscale Methods for Data on Graphs and Irregular Multidimensional Situations. *Journal of the Royal Statistical Society Series B* **71** 97–125.

Killick, R., Fearnhead, P. and Eckley, I. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association* **107** 1590–1598.

Killick, R., Nam, C., Aston, J. and Eckley, I. (2012). changepoint.info: The changepoint repository.

Koprinska, I. and Carrato, S. (2001). Temporal video segmentation: A survey. *Signal Processing: Image Communication* **16** 477–500.

Lavielle, M. (1999). Detection of multiple changes in a sequence of dependent variables. *Stochastic Processes and their Applications* **83** 79–102.

Lavielle, M. (2005). Using penalized contrasts for the change-point problem. *Signal Processing* **85** 1501-1510.

Lavielle, M. and Moulines, E. (2000). Least-squares estimation of an unknown number of shifts in a time series. *J. Time Ser. Anal.* **21** 33–59.

Lebarbier, E. (2005). Detecting multiple change-points in the mean of Gaussian process by model selection. *Signal Processing* **85** 717-736.

Lee, C. B. (1995). Estimating the number of change points in a sequence of independent normal random variables. *Statistics and Probability Letters* **25** 241–248.

Li, H., Munk, A. and Sieling, H. (2016). FDR-control in multiscale change-point segmentation. *Electron. J. Statist.* **10** 918–959.

Lio, P. and Vanucci, M. (2000). Wavelet change-point prediction of transmembrane proteins. *Bioinformatics* **16** 376–382.

Lu, L., Zhang, H. J. and Jiang, H. (2002). Content Analysis for Audio Classification and Segmentation. *IEEE Transactions on Speech and Audio Processing* **10** 504–516.

Mahmoud, M., Parker, P., Woodall, W. and Hawkins, D. (2007). A Change Point Method for Linear Profile Data. *Qual. Reliab. Engng. Int.* **23** 247–268.

Maidstone, R., Hocking, T., Rigaill, G. and Fearnhead, P. (2017). On Optimal Multiple Changepoint Algorithms for Large Data. *Statistics and Computing* **27** 519–533.

Matteson, D. and James, N. (2014). A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data. *Journal of the American Statistical Association* **109** 334–345.

Minin, V., Dorman, K., Fang, F. and Suchard, M. (2005). Dual multiple change-point model leads to more accurate recombination detection. *Bioinformatics* **21** 3034–3042.

Olshen, A., Venkatraman, E. S., Lucito, R. and Wigler, M. (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics* **5** 557–572.

Pan, J. and Chen, J. (2006). Application of modified information criterion to multiple change point problems. *Journal of Multivariate Analysis* **97** 2221-2241.

Rigaill, G. (2015). A pruned dynamic programming algorithm to recover the best segmentations with 1 to $K_{\max}$ change-points. *Journal de la Societe Francaise de Statistique* **156** 180–205.

Rinaldo, A. (2009). Properties and refinements of the fused lasso. *Annals of Statistics* **37** 2922-2952.

Robbins, M., Gallagher, C., Lund, R. B. and Aue, A. (2011). Mean shift testing in correlated data. *Journal of Time Series Analysis* **32** 498511.

Rojas, C. and Wahlberg, B. (2014). On change point detection using the fused lasso method. *Preprint*.

Schroeder, A. L. and Fryzlewicz, P. (2013). Adaptive trend estimation in financial time series via multiscale change-point-induced basis recovery. *Statistics and Its Interface* **6** 449–461.

Shriberga, E., Stolckea, A., Hakkani-Türb, D. and Türb, G. (2000). Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication* **32** 127–154.

Tartakovsky, A., Rozovskii, B., Blazek, R. and Kim, H. (2006). A Novel Approach to Detection of Intrusions in Computer Networks via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods. *IEEE Transactions on Signal Processing* **54** 3372–3382.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc. B* **67** 91-108.

Venkatraman, E. S. (1992). Consistency results in multiple change-point problems. *Technical Report No. 24, Department of Statistics, Stanford University, available from* https://statistics.stanford.edu/resources/technical-reports.

Venkatraman, E. S. and Olshen, A. (2007). A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics* **23** 657–663.

Vostrikova, L. (1981). Detecting 'disorder' in multidimensional random processes. *Soviet Math. Dokl.* **24** 55–59.

Wang, Y. (1995). Jump and Sharp Cusp Detection by Wavelets. *Biometrika* **82** 385–397.

Wang, H., Zhang, D. and Shin, K. (2004). Change-Point Monitoring for the Detection of DoS Attacks. *IEEE Transactions on Dependable and Secure Computing* **1** 193–208.

Wu, Y. (2008). Simultaneous change point analysis and variable selection in a regression problem. *Journal of Multivariate Analysis* **99** 2154-2171.

Yao, Y. C. (1988). Estimating the number of change-points via Schwarz' criterion. *Stat. Prob. Lett.* **6** 181–189.

Yao, Y. C. and Au, S. T. (1989). Least-Squares Estimation of a Step Function. *Sankhya Series A* **51** 370–381.

Zhang, N. and Siegmund, D. (2007). A Modified Bayes Information Criterion with Applications to the Analysis of Comparative Genomic Hybridization Data. *Biometrics* **63** 22–32.

Department of Statistics
London School of Economics
Houghton Street
London WC2A 2AE
UK E-mail: p.fryzlewicz@lse.ac.uk