An Adaptive Lifting Algorithm and Applications

Knight, Marina
University of Bristol, School of Mathematics
University Walk
Bristol BS8 1TW, UK
E-mail: Marina.Knight@bristol.ac.uk

Nunes, Matthew
University of Bristol, School of Mathematics
University Walk
Bristol BS8 1TW, UK
E-mail: Matt.Nunes@bristol.ac.uk

1. Function decomposition with classical wavelets

In classical Fourier analysis, functions can be decomposed as the limit of a linear superposition of sine and cosine waves. However, waves at many frequencies are needed to represent functions which display a non-smooth behaviour. Bases of wavelet functions also allow the decomposition of a function, but in a location-scale environment. It all starts with a *mother wavelet*, a function $\psi \in L^2(\mathbb{R})$ with oscillatory behaviour ($\int_{-\infty}^{\infty} \psi(x)\, dx = 0$). The wavelet ψ can be translated and rescaled, $\psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k)$. The index j refers to the scale of the wavelet function, and the index k to its location. For certain choices of ψ, the wavelet family $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ is a (orthonormal) basis for the space $L^2(\mathbb{R})$. Hence a function, f, can be represented as

$$(1) \qquad f(x) = \sum_k c_{j_0,k}\varphi_{j_0,k}(x) + \sum_{j\geq j_0}\sum_k d_{j,k}\psi_{j,k}(x).$$

In the above formulae, $\varphi_{j_0,k}$ are rescaled and translated versions of the scaling function φ (at scale 2^{j_0} and location $2^{-j_0}k$), the component $\sum_k c_{j_0,k}\varphi_{j_0,k}(x)$ represents a coarse ("zoomed out") approximation of the function f, while the term $\sum_{j\geq j_0}\sum_k d_{j,k}\psi_{j,k}(x)$ contributes the detail lost by the approximation.

The coefficients $c_{j_0,k} =< f, \varphi_{j_0,k} >$ (respectively $d_{j,k} =< f, \psi_{j,k} >$) are known as the *scaling* (respectively *wavelet*) coefficients. The function f can now be regarded as being converted into a (sparse) set of coefficients, useful for compression applications.

The computation of the scaling and wavelet coefficients is set up under the Discrete Wavelet Transform (DWT). For an equally spaced sequence $\mathbf{f} = \{f(i/n)\}_{i=1}^n$ with $n = 2^J$, the DWT computes discrete (periodic) wavelet coefficients $\mathbf{d} = \{d_{j,k}\}_{j=0,\ldots,J-1;k=0,\ldots,2^j-1}$. Note that we abuse notation by keeping the same symbols for both the discrete and continuous wavelet coefficients.

In most implementations the fast DWT of Mallat (1989) is used for computing the scaling and wavelet coefficients.

2. Second generation wavelets

Motivation. As seen above, data is assumed to be equally spaced and of length $n = 2^J$ for some $J \in \mathbb{N}$. Many approaches have been proposed in the literature in case one of these assumptions is broken, e.g. interpolation (Kovac and Silverman, 2000). However, these methods are based on strong assumptions and/or subjective choices. For instance, in interpolation methods, choices need to be made, such as location and spacing of the regular grid or interpolation method, which will influence performance. In obtaining an efficient function representation (few wavelet coefficients), an appropriate choice of the decomposing wavelet is crucial, therefore posing the common question "Which wavelet should I use?" Some work (e.g. Nason (2002)) suggests using cross-validation to choose the wavelet smoothness, which can help, but not for functions that possess varying degrees of smoothness in different locations. The methods we propose efficiently address these limitations.

In the mid-nineties, Sweldens (1996, 1997) proposed a new wavelet construction which enabled wavelet decompositions to be applied to very general data situations, in particular to data with irregular design, irrespective of its length. These wavelets are known in the literature as *second generation wavelets* and can be designed through the *lifting algorithm*.

Lifting a discrete data vector $\mathbf{f}$ essentially consists of three steps, split, predict and update:

- *Split* the signal into two mutually exclusive subsets, say $\mathbf{f}^I$ and $\mathbf{f}^J$.
- *Predict* one subsample, $\mathbf{f}^J$, using the other one, $\mathbf{f}^I$, and encode the difference in $\mathbf{d}^J$. Of course, good prediction is only possible if the function f possesses some degree of local smoothness.
- *Update* $\mathbf{f}^I$ by using the information contained in the wavelet coefficients. The purpose of this stage is to preserve in the updated signal some scalar quantity that characterizes the initial signal.
- Repeat the previous steps on the updated subsample until the end of the decomposition.

The signal $\mathbf{f}$ is replaced by a set of scaling and detail coefficients, which is similar to the DWT. The signal can be easily reconstructed by inverting the previous steps (undo update, undo predict, then merge).

Behind the scenes, the lifting scheme induces a recursive construction of (primal and dual) second generation wavelet functions. However, these functions are no longer translations and dilations of a single mother wavelet. Also, the wavelet bases generated as such are no longer guaranteed to be Riesz, and therefore may exhibit stability problems as investigated by Simoens and Vandewalle (2003), Vanraes *et al.* (2002).

Variations on the split-predict-update stages exist in the literature. Sweldens (1996, 1997) proposes splitting the data into odd and even indices, while Jansen *et al.* (2001, 2004) propose generating just one wavelet coefficient at each step. Simple wavelet bases constructed via the lifting scheme with an odd/even split and their properties have been described by Delouille *et al.* (2001, 2004). The prediction step can also be done in various ways: predicting each odd by the following even or neighbouring evens, or, when removing one coefficient at a time, by regression over the neighbours. Claypoole *et al.* (1998, 2003), Piella and Heijmans (2002) use a split-update-predict algorithm in order to allow for adaptively built wavelets.

Extensions of the lifting scheme to more than one dimension are not straightforward using the even/odd split, but there are various alternatives. In our work we adopt the 'one coefficient at a time' split proposed by Jansen *et al.* (2001, 2004), which we will next describe and then explain how we embed adaptiveness in the algorithm.

3. Lifting one coefficient at a time

Suppose we have a function f, sampled at n irregularly-spaced points $\mathbf{x} = \{x_i\}_{i=1}^n$. Our aim is to transform the sampled function values by means of lifting into a set of detail and scaling coefficients.

Take the initial scaling functions to be the characteristic functions of the cells associated with each point. So $\varphi_{n,k}(x_i) = \delta_{i,k}$, for $k, i \in \{1, \dots, n\}$, and f can be expressed as $f(x) = \sum_{k=1}^n c_{n,k}\varphi_{n,k}(x)$, where $f(x_i) = c_{n,i}$ (hence the function values on the irregular grid are used as the initial scaling coefficients).

For the first lifting step (say, stage n) a point to be lifted must be chosen. Jansen *et al.* (2004) propose to choose the point to be lifted, j_n, such that $\int \varphi_{n,j_n}(x)\, dx = \min_{k\in\{1,\dots,n\}} \int \varphi_{n,k}(x)\, dx$. Denote $I_{n,k} = \int \varphi_{n,k}(x)\, dx$. The first coefficients to be obtained are the ones corresponding to the finest detail, with further steps eliciting progressively coarser detail.

After choosing the point to be removed, j_n, we identify its set of neighbours, I_n. We use the neighbours to predict the value of the function at j_n using simple regression techniques. The prediction phase yields an estimate of the form $\sum_{i\in I_n} a_i^n c_{n,i}$, where a^n are the weights resulting from the regression procedure over I_n.

If j_n has only one neighbour, i, then the prediction is $f(x_i)$. The detail coefficient will be obtained from

$$(2) \qquad d_{j_n} := c_{n,j_n} - \sum_{i \in I_n} a_i^n c_{n,i},$$

or in the one neighbour case,

$$(3) \qquad d_{j_n} := c_{n,j_n} - c_{n,i}.$$

The update phase only affects the scaling coefficients associated with the neighbouring points:

$$(4) \qquad c_{n-1,i} := c_{n,i} + b_i^n d_{j_n}, \; \forall i \in I_n, i \neq j_n.$$

The aim of the update stage is to keep $\sum_{i \in I_n} c_{n,i} \, I_{n,i}$ constant across the scales, and the b^n are obtained using this condition. Further, the integral associated with the removed point gets redistributed to its neighbours, see Jansen *et al.* (2001, 2004) for further details on these issues.

At this point, Jansen *et al.* (2004) state that the signal can be represented as

$$(5) \qquad f(x) = d_{j_n} \psi_{j_n}(x) + \sum_{i \in \{1,\ldots,n\} \backslash \{j_n\}} c_{n-1,i} \varphi_{n-1,i}(x),$$

where ψ_{j_n} and $(\varphi_{n-1,i})_i$ are the analogues of the usual wavelet and scaling functions.

To summarize, a point, j_n, is identified, the scaling function ϕ_{n,j_n} is destroyed and a wavelet ψ_{j_n} is created with new coefficient d_{j_n}. All neighbouring scaling function coefficients of point j_n get updated. All this results in representation (5), and the process is then repeated.

However, unlike the usual discrete wavelet case there are no neat analytical formulae for the scaling and wavelet functions. These functions are recursively constructed as the algorithm proceeds and depend on the locations of the input points $\mathbf{x}$. More details on the precise interpretation and construction can be found in Jansen *et al.* (2001, 2004). Orthogonality of the wavelet and scaling functions is a desirable feature since it would ensure the stability of the transform, but it does not hold in this context.

4. Adaptive lifting

Different adaptive lifting constructions appear in the literature, mostly in a 2D context (Claypoole *et al.*, 1998, 2003; Boulgouris *et al.*, 2001) although some 1D studies exist (Piella and Heijmans, 2002; Trappe and Liu, 2000). Only small studies on the performance of the proposed methods are provided, with little comparison to alternative competitors. In all this work the philosophy is to choose the 'wavelet functions' locally to represent the signal in the most efficient way. The above adaptive lifting techniques use the usual odd/even splitting, whereas we augment with adaptiveness the 'one coefficient at a time' methodology of Jansen *et al.* (2001). Further, we shall address the statistically important case of multiple f-values for each given x.

Possible sources of adaptiveness are:

1. At each step, different *configurations of the neighbours* can be chosen: the closest neighbours to the removed point might be a choice, or its symmetrical neighbours.

2. The *number of neighbours* to use at each step is also subject to choice.

3. The *prediction method* can be linear, quadratic or cubic regression over the specified neighbourhood. In wavelet language this corresponds to locally using more vanishing moments, which is of great utility when the function is locally smooth, but not when there are discontinuities present.

We embed this flexibility in the following two adaptive lifting algorithms:

AdaptPred. At each step, the algorithm chooses the type of regression (linear, quadratic or cubic, with or without an intercept) which generates the smallest detail in absolute value.

This method surpasses the choice of regression order, and the wavelet bases adapt themselves to the signal smoothness.

AdaptNeigh. We introduce even more flexibility by allowing the neighbourhood size and configuration to change at each step. Essentially, *AdaptNeigh* performs the *AdaptPred* procedure for different neighbourhood choices, and minimizes the detail coefficient from all implementations of *AdaptPred*.

This second construction completely frees the user from making any choice except for the neighbourhood size. The considered neighbourhood configurations are symmetric neighbours up to and including a prespecified number each side, and of closest neighbours up to twice the specified number.

At this point one might like to refer to Figure 1 which shows the *HeaviSine* test signal decomposed with an *AdaptNeigh* algorithm. The plots show where linear, quadratic and cubic basis functions get placed.

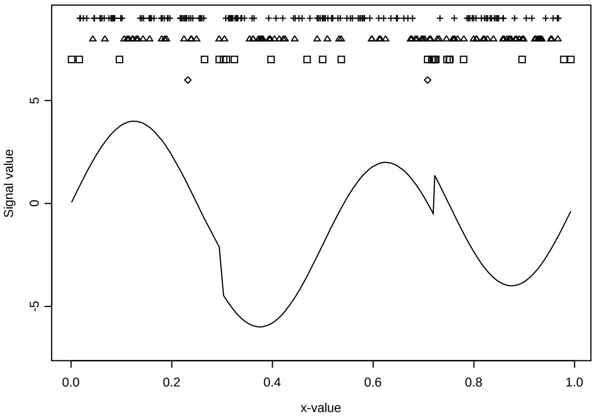

Figure 1: Plot showing choice of prediction scheme for the *HeaviSine* test signal decomposed with AN2 on an irregular grid. Horizontal placement of symbol indicates location of following kinds of prediction: linear (□); quadratic (△); cubic (+); scaling functions (◇).

We shall now discuss some properties of the proposed methods.

Choosing the neighbourhood. In both adaptive algorithms the neighbourhood size has to be specified by the user, and a choice can be made based on the prior knowledge of the signal. We do however advise against using large neighbourhoods, as this increases the chances of using points that do not belong to the same scale. In our algorithm, when at a particular stage the point to be removed happens to be on the boundary, rather than using the requested number of neighbours (which would then come only from one side), we only use its closest neighbour to prevent against using 'artificial' neighbours. Our investigations have shown that using asymmetrical neighbours does not seem to have an impact on our transforms, as long as we make use of the appropriate number of neighbours.

Regression order. Unstable transforms are generated using curves of higher order than the available number of neighbours would allow (i.e. curves of the highest possible order, forced through the origin). Hence we use each type of prediction with its appropriate number of neighbours. At certain steps of the transform we will be in the situation of not having enough neighbours, hence the order of prediction needs to be decreased, while for boundary points we will always predict using step functions. In consequence, we need to be aware that in fact we will obtain a mixture of regression orders (although we required a fixed order).

Nonlinearity of adaptive algorithms. Recall that for prediction schemes which use least squares regression, the prediction weights are computed using the grid values only, so the transform is linear. However, when we introduce adaptivity into the transform design, the type of prediction used adapts to the local signal structure. This in turn induces dependence of the prediction weights, function integrals, update weights and the next point chosen for removal on the signal f. The transform operator associated to the adaptive transforms is therefore not linear, and is a function of the signal.

Update weights. The update weights are also responsible for instability of the transform, and in our algorithm we used $b_i^r = I_{r,j_r} I_{r-1,i} / \sum_{k \in I_r} I_{r-1,k}^2$, as suggested by Jansen *et al.* (2001). This choice ensures that the obtained weights have minimum norm, which prevents the new wavelet and scaling functions at each stage (here r) from being too close to each other.

Multiple observations at x-values. Many real situations are encountered where we have several values f_i at the same location x_i. The prediction step uses a linear least squares approach to estimate the unknown parameters of the regression curve to be fit to the data, which can naturally cope with multiple observations. In the update step, all multiple neighbours get updated using the corresponding detail obtained in the prediction stage. So neighbour points that were multiple remain multiple after the update step. If the point to be removed is itself multiple, we have chosen to produce one detail coefficient by taking the mean of the distinct individual detail coefficients, although other quantities, such as the minimum, can be interesting alternatives. Finally, when the coarsest level for decomposition has been reached, if some of the scaling points are multiple, we replace them by their means, and then invert the lifting transform.

To conclude, adaptiveness in our algorithm brings some important advantages: wavelet functions are tuned to the local features of the signal, therefore the usually difficult question of which wavelet to use is waived; better wavelet coefficient sparsity and denoising capacity are achieved; the algorithm is computationally efficient $\mathcal{O}(n)$; it is simple to handle the situation of multiple f-values for a given x-value.

5. Performance and applications in nonparametric regression

A thorough simulation study was performed in order to evaluate the denoising performance of our proposed methods. The problem we addressed with the study is the classical nonparametric regression model, i.e. model the observations as $f_i = g(x_i) + \varepsilon_i$, where the vector $\mathbf{f}$ denotes noise-contaminated observations at irregularly spaced points $\mathbf{x}$, g is the true unknown function to be estimated, and ε is Gaussian noise. The approach we take is to apply wavelet shrinkage (Donoho and Johnstone, 1994, 1995) with adaptive lifting instead of the DWT, and an adapted version of the empirical Bayes thresholding (Johnstone and Silverman, 2004a,b; Nunes *et al.*, 2006).

The study was performed on the five test signals *Doppler*, *Bumps*, *Blocks*, *HeaviSine* (Donoho and Johnstone, 1994), and *Ppoly* (Nason and Silverman, 1994), over irregular grids with varying levels of irregularity and different additive noise levels.

Several different variants of our linear and adaptive lifting schemes were compared against wavelet and non-wavelet function estimators— the local polynomial fitting estimator *Locfit* Loader (1997, 1999); the S-Plus function `smooth.spline()` with cross-validation, which fits smoothing splines to the data (SSCV); the Comte and Rozenholc (2004) (CR) method, a denoiser based on adaptive polynomial basis selection; and the irregular interpolation wavelet algorithm by Kovac and Silverman (2000) (KS) .

The overall accuracy of the estimates is measured by the *average mean square error*, defined by AMSE $= (nK)^{-1} \sum_{k=1}^{K} \sum_{i=1}^{n} (g_i^k - \hat{g}_i^k)^2$, where $K = 100$ is the number of simulations and $\hat{g}^k$ is the estimate of the true signal g^k on the kth repetition.

Overall, our methods are very competitive (see Table 1), with *AdaptNeigh* with neighbourhoods of size up to two being significantly better on the signals with discontinuities over all grid irregularities and noise levels. For smoother signals, *AdaptPred* with two neighbours worked best, outperforming the other smoothers over all degrees of grid jitters and noise level, apart from one of the signals, where the Kovac-Silverman algorithm was best.

Table 1: AMSE ($\times 10^3$) simulation results for test signals of length $n = 256$ with SNR=3 on grids with three levels of irregularity, d_ℓ, for various denoising methods described in the text: LP (*LinearPred*), AP (*AdaptPred*), AN (*AdaptNeigh*). S denotes symmetrical neighbours, while N denotes closest neighbours.

	Blocks			*Bumps*			*HeaviSine*			*Doppler*			*Ppoly*		
Method	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3	d_1	d_2	d_3
LP1S	72	71	68	81	80	73	20	20	21	54	53	52	16	16	18
AP2N	69	70	59	78	75	64	21	21	22	53	52	48	15	16	17
AN1	55	54	52	66	67	61	36	39	37	61	61	59	38	33	32
Locfit	73	72	64	110	108	101	11	11	11	58	58	54	21	20	19
SSCV	74	74	67	307	315	250	12	11	12	61	60	53	20	20	19
KS	79	78	87	179	181	259	13	12	15	51	52	57	18	17	18
CR	119	119	133	332	313	284	25	25	25	155	155	148	13	13	13

6. Real data application

We shall now give a real data example which addresses the problem of predicting hydrophobic segments along the sequence of a transmembrane protein, when no information on the protein is available, other than its primary structure. For details of the method see Knight and Nason (2006).

For a protein, its hydropathy profile can be constructed by using a measure of hydrophobicity for each amino acid in its chain. We obtain a signal which on the horizontal axis has the residues in their order of appearance in the protein primary structure, and on the vertical axis their corresponding values from the hydrophobicity index (we use the Kyte and Doolittle (1982) hydrophobicity scale). The 3D information contained in proteins similar to the protein of interest will be used to estimate the residue locations. Figure 2 gives an example of such a signal, corresponding to the protein undecaprenyl-phosphate galactosephosphotransferase (UniProt entry 'rfbp-salty').

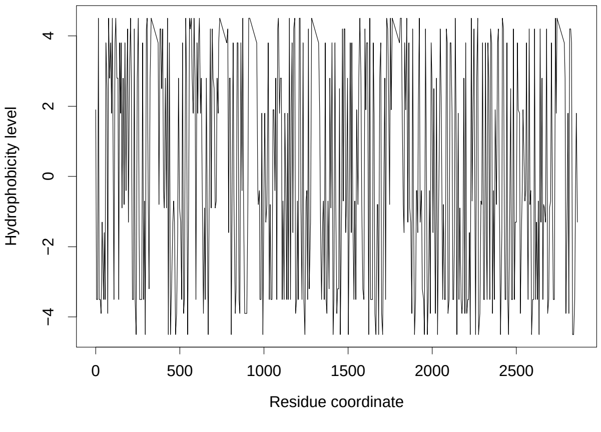

Figure 2: Example of a hydropathy profile.

Since transmembrane helices are sequences of predominantly hydrophobic residues, our purpose is to detect the points at which sharp changes occur in the hydrophobicity signal. This amounts to modelling the profile as noise-contaminated, and estimating the underlying signal.

We shall address the statistical problem of denoising the above hydropathy signal by using our adaptive wavelet methodology, specifically *AdaptNeigh* with at most two neighbours and empirical Bayes shrinkage using the posterior mean (*AN1 mean*).

We will now examine the results for 'rfbp-salty'. Starting from its primary structure of 476 residues, one protein with known 3D structure has been aligned to it and used in the computation of inter-residue distances.

Experimentally determined data is available for the transmembrane segments, which are thought to be 15-35, 52-72, 93-113, 115-135, 283-303. After employing our method (*AN1 mean*), we obtain the following predicted segments 15-40, 60-72, 94-107, 116-135, 284-301, while through the usage of classical wavelets Daubechies S8 with empirical Bayes shrinkage using the posterior mean (*Daub mean*) we obtain 12-40, 55-72, 89-110, 116-139, 170-186, 240-260, 279-304, 459-475, corresponding to Figure 3.

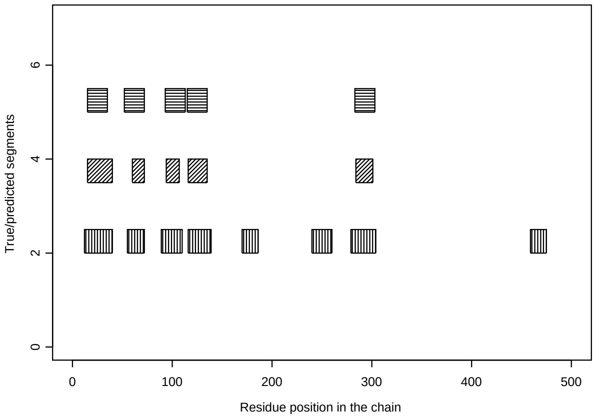

Figure 3: True and predicted segments for 'rfbp-salty': horizontally filled rectangles=True, diagonally filled rectangles=*AN1 mean*, vertically filled rectangles=*Daub mean*.

Our method does not predict extra segments, and it correctly identifies the true segments. The classical method falsely predicts as transmembranar three segments which would be difficult to eliminate through a further (chemical) filtering step.

REFERENCES

Boulgouris, N.V., Tzovaras, D. and Strintzis, M.G. (2001) Lossless image compression based on optimal prediction, adaptive lifting and conditional arithmetic coding. *IEEE Trans. Im. Proc.*, **10**, 1–14.

Claypoole, R.L., Baraniuk, R.G. and Nowak, R.D. (1998) Adaptive wavelet transforms via lifting. In Transactions of the International Conference on Acoustics, Speech and Signal Processing. *IEEE Trans. Im. Proc.*, **12**, 1513–1516.

Claypoole, R.L., Davis, G.M., Sweldens, W. and Baraniuk, R.G. (2003) Nonlinear wavelet transforms for image coding via lifting. *IEEE Trans. Im. Proc.*, **12**, 1449–1459.

Comte, F. and Rozenholc, Y. (2004) A new algorithm for fixed design regression and denoising. *Ann. Inst. Statist. Math.*, **56**, 449–473.

Delouille, V., Franke, J. and von Sachs, R. (2004) Nonparametric stochastic regression with design-adapted wavelets. *Sankhya*, A, **63**, 328–366.

Delouille, V., Simoens, J. and von Sachs, R. (2004) Smooth design-adapted wavelets for nonparametric stochastic regression. *J. Am. Statist. Soc.*, **99**, 643–658.

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

Donoho, D.L. and Johnstone, I.M. (1995) Adapting to unknown smoothness via wavelet shrinkage. *J. Am. Statist. Soc.*, **90**, 1200–1224.

Jansen, M., Nason, G.P. and Silverman, B.W. (2001) Scattered data smoothing by empirical Bayesian shrinkage of second generation wavelet coefficients. In Unser, M. and Aldroubi, A. (eds) *Wavelet applications in signal and image processing*, Proceedings of SPIE, **4478**, 87–97.

Jansen, M., Nason, G.P. and Silverman, B.W. (2004) Multivariate nonparametric regression using lifting. *Technical Report* 04:17, Statistics Group, Department of Mathematics, University of Bristol, UK.

Johnstone, I.M. and Silverman, B.W. (2004a) Needles and hay in haystacks: Empirical Bayes estimates of possibly sparse sequences. *Ann. Statist.*, **32**, 1594–1649.

Johnstone, I.M. and Silverman, B.W. (2004b) EbayesThresh: R programs for Empirical Bayes thresholding. *J. Statist. Soft.*, **12**, 1–38.

Knight, M.I. and Nason, G.P. (2006) Improving prediction of hydrophobic segments along a transmembrane protein sequence using adaptive multiscale lifting. *Multisc. Model. & Sim.***5**, 115–129.

Kovac, A. and Silverman, B.W. (2000) Extending the scope of wavelet regression methods by coefficient-dependent thresholding. *J. Am. Statist. Ass.*, **95**, 172–183.

Kyte, J. and Doolittle, R.F. (1982) A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, **157**, 105–132.

Loader, C. (1997) Locfit: an introduction. *Stat. Comput. Graph. News.*, **8**, 11-17.

Loader, C. (1999) *Local regression and likelihood.* Springer: New York.

Mallat, S.G. (1989) A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattn. Anal. Mach. Intell.*, **11**, 674–693.

Nason, G.P. and Silverman, B.W. (1994)The discrete wavelet transform in S. *J. Comp. Graph. Statist.*, **3**, 163–191.

Nason, G.P. (2002) Choice of wavelet smoothness, primary resolution and threshold in wavelet shrinkage. *Stat. Comput.*, **12**, 219–227.

Nunes, M.A., Knight, M.I. and Nason, G.P. (2006) Adaptive lifting for nonparametric regression. *Stat. Comput.*, **16**, 143–159.

Piella, G. and Heijmans, H.J.A.M. (2002) Adaptive lifting schemes with perfect reconstruction. *IEEE Trans. Sig. Proc.*, **50**, 1620–1630.

Simoens, J. and Vandewalle, S. (2003) A stabilized construction of wavelets on irregular meshes on the interval. *SIAM J. Scientific Computing*, **24**, 1356–1378.

Sweldens, W. (1996) Wavelets and the lifting scheme: A 5 minute tour. *Z. Angew. Math. Mech.*, **76**, 41–44.

Sweldens, W. (1997) The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, **29**, 511–546.

Trappe, W. and Liu, K.J.R. (2000) Denoising via adaptive lifting schemes. In *Proceedings of SPIE, Wavelet applications in signal and image processing VIII*, Aldroubi, A., Laine, M.A. and Unser, M.A. (eds), **4119**, 302–312.

Vanraes, E., Jansen, M. and Bultheel, A. (2002) Stabilised wavelet transforms for non-equispaced data smoothing. *Sig. Proc.*, **82**, 1979–1990.