Efficient denoising of very large experimental datasets. Application to FT-ICR mass spectrometry. *preprint*

L. Chiron[1] M.A. van Agthoven[2] B. Kieffer[1], C. Rolando[2], M-A. Delsuc[1]

- nov 2013-

1) Institut de Génétique et de Biologie Moléculaire et Cellulaire, INSERM, U596; CNRS, UMR7104; Université de Strasbourg, 1 rue Laurent Fries, 67404 Illkirch-Graffenstaden, France,

2) Miniaturisation pour l'Analyse, la Synthese & la Protéomique (MSAP), USR CNRS 3290, and Protéomique, Modifications Post-traductionnelles et Glycobiologie, IFR 147, Université de Lille 1 Sciences et Technologies, 59655 Villeneuve d'Ascq Cedex, France

Modern scientific research produces datasets of increasing size and complexity that require dedicated numerical methods to be processed. In many cases, the analysis of spectroscopic data involves the denoising of raw data prior to any further processing. Current efficient denoising algorithms require the Singular Value Decomposition of a matrix with a size that scales up as the square of the data length, preventing their use on very large datasets.

Taking advantage of recent progress on random projection and probabilistic algorithms, we developed a simple and efficient method for the denoising of very large datasets. Based on the QR decomposition of a matrix randomly sampled from the data, this approach allows a gain of nearly three orders of magnitude in processing time compared to classical SVD denoising. This procedure, called urQRd, strongly reduces the computer memory foot-print and allows the denoising algorithm to be applied to virtually unlimited data size.

The efficiency of these new numerical tools is demonstrated on experimental data from high resolution broad-band FT-ICR mass spectrometry, which has with applications in proteomics and metabolomics. We show that robust denoising is achieved in 2D spectra whose interpretation is severely impaired by scintillation noise. These denoising procedures can be adapted to many

other data analysis domains where the size and/or the processing time are crucial.

keywords : denoising | FT-ICR MS | random projection | SVD | big data
abbreviations : FT, Fourier transform; ICR, Ion Cyclotron Resonance; MS, mass spectrometry; AR, Auto Regressive; SVD, Singular Value Decomposition; rQRd, random QR denoising; urQRd, uncoiled random QR denoising;

1 Introduction

Big data are becoming predominant in modern science, and found in many scientific domains: astrophysics [1], large-scale physics experiments [2], biology [3, 4], or even natural text analysis [5]. This introduces a new challenge for data handling and analysis, as it requires special processing approaches, which avoid accessing the whole data at once [6], and make use of adapted algorithms easily parallelized. Such constraints may be difficult to fulfill, even for simple procedures such as noise reduction.

Every measurement is corrupted by unwanted noise, which is the combination of the effect of random fluctuations in the sample and the apparatus, but can also originate from external events like environmental noise. Denoising methods focus mainly on removing or reducing as much as possible an additive gaussian WSS (Wide Sense Stationary) noise.

For stationary signals the optimal linear denoising filter in the mean-square error sense is the Wiener filter. However, it suffers from the requirement of a precise estimate of the signal and noise auto- and cross-correlation functions. Many advanced denoising methods have been developed using linear algebra, which usually require considerable processing power. One of the main alternative approach relies on a multi-resolution analysis which sets apart noise from signal components more efficiently than classical orthogonal basis methods. In this respect, wavelets associated with soft thresholding have been considered for denoising purposes [7]. These methods require *a priori* knowledge about signal and noise power and are well suited for transient signals but not as much for stationary processes.

Harmonic signals can be modeled as the sum of damped sinusoids. They are typically found in spectroscopies like NMR, FT-MS, FT-IR but also in seismology, astrophysics, genetics, or financial analysis. They are easily analyzed by Fourier transformation if regularly sampled. For such specific signals, one class of denoising methods is based on modeling a sum of a fixed number of exponentials as devised by Prony [8]. This was recently revisited and improved by Beylkin et al [9, 10].

There are also statistical methods related to the Karhunen-Loève transform, which use adaptive basis instead of *a priori* basis. Relying on the auto-regressive model (AR) [11, 12] a Hankel matrix is built and its rank is then reduced to the number of expected frequencies. Rank reduction by the Singular Value decomposition (SVD) [13] of this matrix is known as Cadzow's method [14] also known as SSA [15]. The advantage is that no assumption about the noise or signal power is required and the number of frequencies is the only parameter.

But the benefits of denoising are balanced by several drawbacks. If the assumed number of frequencies is incorrect, the denoised signal is polluted with spurious artifacts that are indistinguishable from the real signal. Additionally, the SVD decomposition is slow and scales in $O(mn^2)$ operations. Alternative rapid SVD algorithms can be used, such as the Lanczos bi-diagonalization method [16,17], the truncated SVD [18] or random projections [19] as was recently applied in seismology [20]. However, these algorithms do not solve the artifacts issue.

Capitalizing on recent progress in algebra on random projection and probabilistic algorithms [21–24], we present here a novel efficient approach to denoising which can be easily applied to the large datasets found in FT-ICR experiments, and more generally, to any big data analysis. The main driving idea is to avoid explicit computation of data derived quantities, but rather estimate the needed values, based on a partial sampling of the data. Extending from previous ideas [19], the denoising algorithm is based on a subsampling of the data-associated matrix. Here, rather than truncating the rank by removing some of the components of the SVD decomposition, we compute a randomized low-rank approximation of the Hankel matrix [24] that retains preferentially more signal than noise.

We show that this leads to a substantial improvement of the processing in terms of speed, with little compromise on the quality, allowing gains of two to three orders of magnitude in processing time and in memory size. Applications of this approach are demonstrated on the large datasets obtained in FT-ICR Mass Spectrometry.

2 Theory

2.1 Denoising algorithm

We propose here a simplified algorithm based on a random sampling of the transfer function associated with the matrix viewed as an operator. For large enough samplings, the data consistency is ensured through a variant of the Johnson-Lindenstrauss lemma [21, 23].

2.2 The AR model and the H matrix

The auto-regressive (AR) model assumes a regularly sampled complex harmonic signal. For such a signal composed of a sum of P components, each data point X_l can be expressed as a linear combination of the P preceding data points [11, 12]. This implies that the Hankel matrix H, built from the data series by copying a shifted version of the data series on each line:

$$(H_{i,j}) = (X_{i+j-1}) \tag{1}$$

is rank-limited to P in the absence of noise. In noisy datasets, this matrix becomes full-rank because of the partial decorrelation of the data points induced by the noise.

These properties have been used by many signal-improving techniques. Cadzow [14] proposed to perform the Singular Value Decomposition (SVD) of the matrix H, and compute a matrix $\tilde{H}$ by truncating to the K largest singular values σ_k. $\tilde{H}$ is not strictly Hankel-structured anymore, but a denoised signal $\tilde{X}$ can be reconstructed by taking the average of all its antidiagonals. (see eq 1)

$$\tilde{X}_l = \underset{i+j=l+1}{mean}(\tilde{H}_{ij}) \tag{2}$$

Unlike linear filtering approaches such as the Wiener filter, the signal is denoised without making assumptions on the signal line-widths, let alone modifying the line-widths.

2.3 Sampling H on a random set of directions

The matrix H can also be considered as an operator $\mathcal{H}$ that concentrates its input vectors onto the main singular vectors that correspond to correlations in the matrix, and thus to harmonic components in the series X. Earlier studies [21, 25] have shown that we can sample $\mathcal{H}$ efficiently by observing its effect on a set of random vectors. For a large enough sampling, the effect of $\mathcal{H}$ is essentially captured with an emphasis on the correlations, therefore taking more from the signal than from the noise. This random sampling considerably reduces the size of the problem and has already been used in the analysis of very large datasets [23, 26, 27].

From the data vector X of length L, let form the $(M \times N)$ Hankel matrix H using eq 1. Then compute the matrix Y as the product of H by a set of K random unit vectors handled as a matrix Ω.

$$\underset{(M\times K)}{Y} = \underset{(M\times N)}{H}\underset{(N\times K)}{\Omega} \tag{3}$$

with $M + N - 1 = L$ and the following relations: $K \leq M$ and $M \leq N$. The matrix Y is thus much smaller than H. M is chosen at will and called the *order* of the analysis.

2.4 QR step, approximation

A QR factorization of Y is performed $Y = QR$ with the matrix Q, as a reduced rank orthonormalized basis of H. From this decomposition the matrix $\tilde{H}$ is built:

$$\tilde{H} = QQ^*H \tag{4}$$

$\tilde{H}$ is the projection of H on the reduced rank orthonormalized basis Q and is a rank K approximation. Unlike the SVD method which gives the low-rank approximate closest to H in sense of the Frobenius norm [28], the random projection gives less tight bounds on H recovery. It has been shown [24] that, for a signal containing exactly P components, this approximation is bound (in term of spectral norm) with a probability larger than $1 - 3p^{-p}$ with $p = K - P$, to:

$$\left\| H - \tilde{H} \right\| \leq \left[1 + 9\sqrt{K}\sqrt{M} \right] \sigma_{P+1} \tag{5}$$

and σ_{P+1} the $P + 1$ largest singular value of H.

$\tilde{X}$ is finally rebuilt from $\tilde{H}$ in a step similar to the one performed in the SVD approach. We propose to call this approach rQRd, standing for ***r**andom **QR** **d**enoising*.

The approach described here relies heavily on the Hankel structure of the underlying matrix, built from the AR model. A slightly different expression of this model leads to Toeplitz matrices which present very similar properties [12].

2.5 Fast Hankel matrix product

The Hankel structure of H implies that applying this matrix to a vector is equivalent to computing the convolution of the data series with this vector. Thanks to the properties of the Fast digital Fourier Transform (FFT), fast Hankel matrix product algorithms can be designed that perform this operation much more rapidly and with a much smaller memory footprint than direct multiplication [13, 29]. This approach presents a processing cost proportional to $O(L\log(L))$ rather than $O(MN)$. In the same manner, using fast Hankel matrix-vector multiplications, the total cost of the product of H with Ω can be reduced to $O(KL\log(L))$ rather than $O(KMN)$ (recall that $K \leq M \leq N \leq L = M + N - 1$).

By combining equations (2) and (4) the denoised signal can also be computed from Q and Q^*H using fast Hankel matrix-vector multiplications. The result of the randomized algorithm for rank-K approximation is a $M \times K$ matrix Q and a $K \times N$ matrix U that approximate H from (4) via

$$H_{i,j} \approx \tilde{H}_{i,j} = \sum_{k=1}^{K} Q_{i,k} U_{k,j} \tag{6}$$

where Q is obtained from the QR decomposition and $U = Q^*H$ (computed again using the fast Hankel matrix product). It follows from (6) that the sum over the i^{th} antidiagonal of $\tilde{H}$ expressed for j ranging from $j_1 = \max(i - M + 1, 1)$ to $j_m = \min(i, N)$ is:

$$\begin{aligned}
\sum_{j=j_1}^{j_m} \tilde{H}_{i-j+1,j} &= \sum_{k=1}^{K} \sum_{j=j_1}^{j_m} Q_{i-j+1,k} U_{k,j} && (7) \\
&= \sum_{k=1}^{K} \sum_{j=j_1}^{j_m} Q^{(k)}_{i,j} U^{(k)}_j && (8) \\
&= \sum_{k=1}^{K} (Q^{(k)} \cdot U^{(k)})_i && (9)
\end{aligned}$$

Here, $Q^{(k)}$ is the $L \times N$ Toeplitz matrix formed from the $L + N - 1$ long vector $[0, ...0, Q_{k,1}, ..Q_{k,M}, 0, ..., 0]$ with $(N - 1)$ zeros added on each extremities, $U^{(k)}$ is the $N \times 1$ vector whose entries are $U^{(k)}_j = U_{k,j}$ and $(Q^{(k)} \cdot U^{(k)})$ denotes the matrix-vector product, which is computed again using a fast algorithm.

Evaluating the right-hand side of (9) requires K fast Toeplitz matrix-vector multiplications for each $i = 1, ..L$, for a total cost proportional to $KL\log(L)$. It is never necessary to explicitely express or calculate the matrix $\tilde{H}$, but just to sum over its antidiagonals.

We gave to this implementation of the rQRd algorithm, the name urQRd standing for ***u**ncoiled **r**andom **QR** **d**enoising*. Both algorithms implement the same analytical procedure but differ only in the implementation details. Only one parameter determines the computation and should be provided by the user: the rank K of the reduced Hankel matrix. The order M of the H matrix (see eq 3) can also be adapted, but plays a lesser role in the outcome of the analysis.

2.6 Implementation

The presented algorithms have been implemented in python, relying on standard mathematical libraries, and using the standard optimization performed in these libraries.

The dominant costs of the algorithm are the initial product with a random matrix (eq (3)) and the final summation over $\tilde{H}$ antidiagonals (eq (9)). Both steps can easily be distributed over a large number of processors, as all terms of the sums are independent and do not need any kind of communication, providing an additional gain in speed. This was not undertaken here, as we are solely using the standard libraries.

The detailed algorithms for fast Hankel matrix product, rQRd and urQRd are presented in Supp Info S1-S3. The code of the programs of the SVD, rQRd and urQRd algorithms ad well as the FT-ICR MS datasets are available at `http://urqrd.igbmc.fr`.

3 Material and Methods

3.1 Processing

All computations were performed on a Macintosh Mac Pro dual Xeon with a total of 12 cores with hyperthreading. The machine was equipped with 32 Gb of memory and was running MacOsX 10.6.8. Programs are implemented in python version 2.7, using the numpy/scipy libraries. The standard Enthought distribution EPD 7.3 (Enthought, Inc. Austin, TX) was used without any modifications. The standard routines available for SVD, QR and FFT are based on LAPACK and FFTPACK, some of the functions rely on the multithreaded MKL library (Intel, Inc. Santa Clara, CA) which insure a partial parallelization.

Simulations were run on a synthetic dataset generated as follows. In figure 1 a dataset of 2000 complex points was simulated consisting of 20 lines, of 1.1 Hz width, sampled over 1 second. A white gaussian noise was added for a SNR around 0 dB. In figure 2 a dataset of 1000 complex points was simulated, consisting of 50 lines in a pattern equivalent to 1. In figure 3 the simulated dataset consisted of 9 lines in a pattern equivalent to 1. The length of the dataset was varied from 1 000 to 4 096 000 complex points, with values alternating 2^n 1000 and 2^n 1400. The parameter M was set to $L/4$. rQRd, urQRd and SVD processing were performed with $K = 100$.

All SNR are expressed in dB as

$$\mathrm{SNR}_X = 10 \log_{10} \frac{\|X_o\|^2}{\|X - X_o\|^2} \tag{10}$$

where X_o is the original clean dataset and X is the noisy dataset. SNR gains were computed as the difference of SNR between before and after denoising; also as expressed as:

$$\text{SNR gain} = \text{SNR}_{\tilde{X}} - \text{SNR}_X \tag{11}$$

where $\tilde{X}$ is the cleaned dataset.

3.2 FT-ICR-MS experiments

The trypsin digest of Cytochrome C was purchased from LC-Packings and used as received. The FT-ICR-MS spectrum was acquired in direct injection at 80 fmol/μl using positive mode electrospray as an ion source. The spectrum was measured on a Bruker ApexQE mass spectrometer, operating at 9.45 T. Acquisition was performed on 524 288 points, sampled with a 1 MHz spectral width, with an *m/z* 144-1500 mass range. This mass spectrum was recorded in 1 scan. The Fourier transform was preceded by a Hamming apodisation of the dataset, and was computed on 2 Mb points. The modulus of the spectrum is displayed. urQRd processing took 25 minutes with K=1000 and M=245 760.

The triacylglycerols were extracted from a sample of human plasma (preparation to be published). The 2D FT-ICR mass spectrum was acquired in the same conditions as the FT-ICR mass spectrum of cytochrome C, with 2048 $\times$ 131 072 data points, with a 1 MHz spectral width in both dimensions and an *m/z* 144-1000 mass range, for a total acquisition time of 2 hours, and a final file size of 2 Gb. Fourier transformation was preceded by a Hamming apodisation, and the dataset was zero-filled once in each dimension. Spectra are presented in magnitude mode. Between the Fourier transformation in the horizontal dimension and the vertical dimension, a digital demodulation was applied, in order to remove the phase rotation introduced by the pulse generator carrier. rQRd processing was applied after the Fourier transform along axis F2 (horizontal) and before transform in F1 (vertical). It was applied on each F1 column, with $K = 50$. Complete rQRd processing time took 45 minutes.

4 Results

4.1 Effect of rank

Robustness with respect to the rank K used for denoising is an important parameter. In the classical SVD approach, the rank is set to the number of expected components present in the signal, as the denoising is optimum at this point. However, this parameter is usually quite difficult to determine *a priori* in biophysical experiments, in particular for FT-ICR MS (see below). It plays a different role in rQRd as exemplified in figures 1 and 2. The algorithm was tested here on a harmonic synthetic dataset presenting a set of sharp frequencies. While generic enough to present the feature of the method as general, this dataset presents some analogy with a FT-ICR MS experiment.

Figure 1 presents the effect of the rank on the aspect of the filtered spectrum. As expected, SVD truncation to rank K of the Hankel matrix produces spectra in which nearly exactly K lines can be observed. This leads inevitably to additional spurious lines when $K > P$ and to missing lines when $K < P$. In the present case, where the smallest signals are vanishingly small and remain buried in the noise, spurious peaks appears even for $K = P$. In contrast, rQRd does not constrain the rank of the Hankel matrix as strongly. As a result, the remaining noise is spread evenly across the spectral width, leading to spectra which appear less distorted.

The quality of denoising is measured as the cartesian distance of the denoised dataset to the ideal noise-free data used for the simulation. With this measure, expressed as SNR gains, it can be observed that SVD presents high SNR for small K, whereas rQRd SNR keeps improving with larger K.

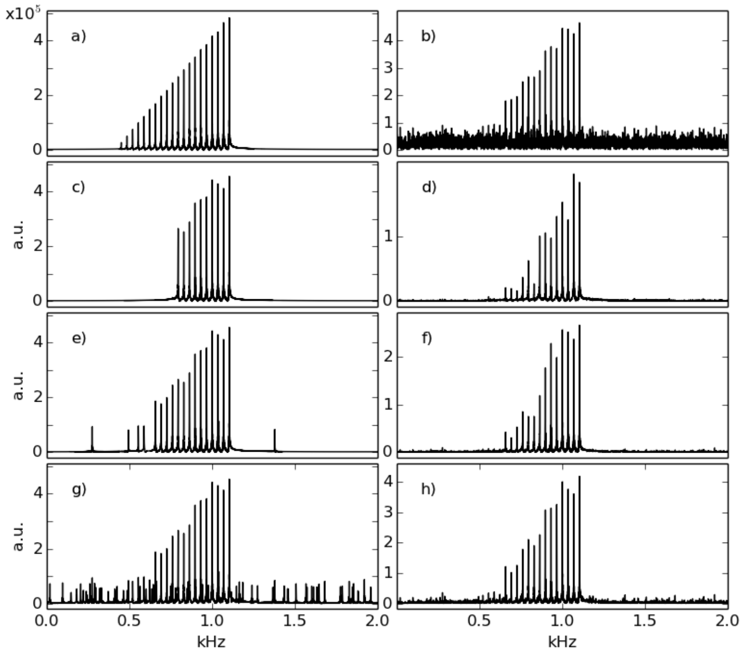

Figure 1: Comparison of the the SNR gain afforded by the denoising methods as a function of the rank. The computations are performed here on a synthetic complex 2000 points dataset containing 20 frequencies. a) Fourier Transform (FT) of the initial synthetic dataset composed of 20 lines of varying intensity. b) FT of the test dataset, with an added Gaussian white noise. SNR of the time-domain dataset is -0.14 dB. c-e-g) FT of the SVD processed of the synthetic dataset with with varying K. d-f-h) FT of the rQRd processed of the synthetic dataset with with varying K. c-d) rQRd and SVD processed of the synthetic dataset with $K = 10$ SNR gains: SVD 8.23 dB rQRd 2.91 dB, e-f) idem with $K = 20$ SVD 12.00 dB rQRd 5.13 dB. g-h) idem with $K = 80$ SVD 6.91 dB rQRd 9.95 dB.

4.2 Quantitativity

Inspection of the spectra shows that while the peak frequencies are perfectly conserved, the weak intensities are distorted in a systematic manner, in particular for small values of K. In many cases, the precision on the signal intensity is as important as on its position. However, due to the way the rQRd denoising algorithm weights the various components of the signal in order to separate the signal and the noise, the relative intensity of each frequency cannot be insured, in particular for signals intensities close to the noise level. While this effect is important for $K \approx P$ it tends to weaken for larger K.

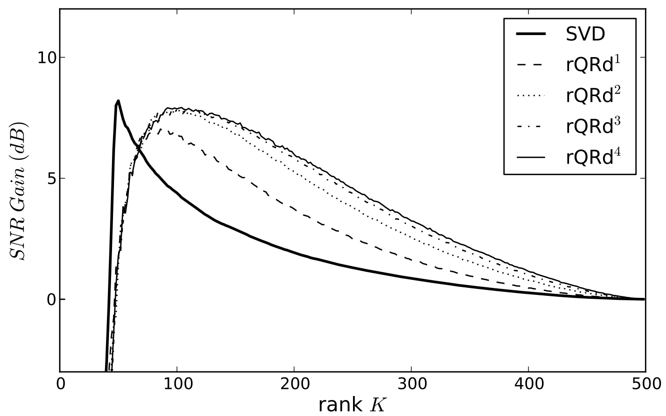

Figure 2: Comparison of the the SNR gain afforded by the denoising methods as a function of the rank. The computations are performed here on a synthetic dataset of 1000 complex points containing 50 components on increasing intensity in a pattern similar to figure 1. rQRd^n indicates the result obtained when iterating the rQRd method n times.

Figure 2 presents the evolution of the SNR gain with respect to the rank. While SVD presents the highest gain for a rank K equal to the number of lines P, rQRd presents a broad region of high SNR, for K in the range 1.5 P to 4 P.

4.3 Iterations

It is a usual approach to apply a denoising procedure in an iterative manner in order to improve noise rejection. This can be performed by applying the whole procedure several times, or by computing several successive multiplications of Ω by H [14, 24]. The first option prove to be more effective thanks to the combined effect of the two successive steps: the low rank projection of H (eq 4) and the antidiagonal averaging (eq 2). This averaging is nearly isometric but brings back the system projected onto a subspace of dimension K to the original space of dimension M. Thus alternating the independent steps of antidiagonal averaging and rank reduction allows a strong noise reduction. It is remarkable that iterating rQRd provides higher SNR in an even broader range of ranks.

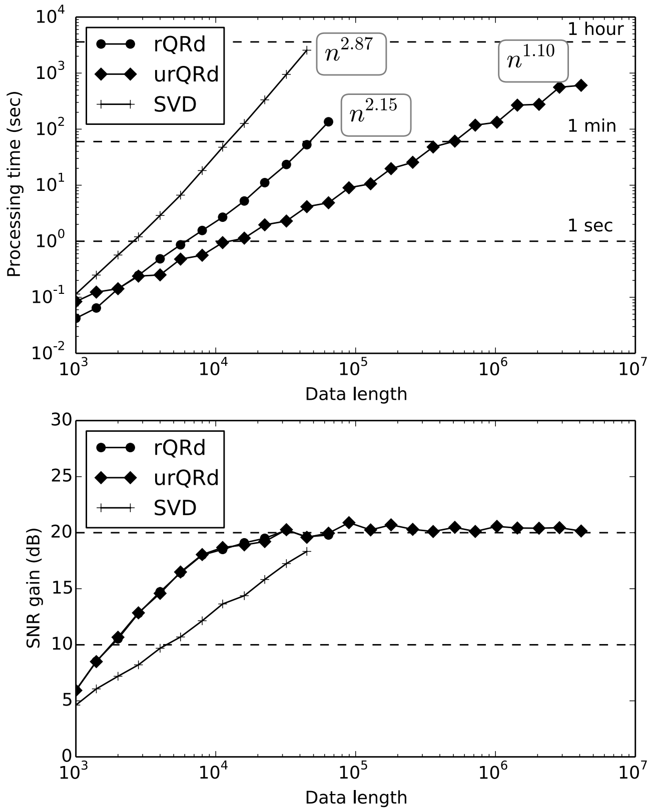

Figure 3: Comparison of the rQRd (bullets) and urQRd (diamonds) denoising approaches, as well as the SVD approach (crosses), performed here on a synthetic complex dataset containing 9 frequencies, and processed with $K = 100$. *Top:* processing times for each method. Asymptotic behavior fitted on the graph are SVD $\sim n^{2.87}$ rQRd $\sim n^{2.15}$ and urQRd $\sim n^{1.10}$. On our machine, the cross-over between rQRd and urQRd is around 2 000-3 000 complex points. *Bottom:* denoising efficiency, expressed as the SNR gain afforded by the denoising method; rQRd and urQRd are nearly indistinguishable on this graph.

4.4 Processing efficiency

Figure 3 presents a comparison of SVD, rQRd and urQRd processing times and efficiencies. SVD and rQRd are limited by the amount of computer memory (here 32 Gb), and stop for datasets larger than $\approx 32\,000$ and $64\,000$ points respectively, a limit which is removed by the urQRd algorithm which does not require large H and $\tilde{H}$ matrices to be stored in memory. It can be seen that the rQRd algorithm affords important SNR gains, with values over 20 dB in the case of large datasets. This simulation has been run with a rank on the order of 10 times the number of expected signals, a conditions which is favorable to rQRd.

Usually many sources of noise are present in a measurement, and most of them are actually non-additive (multiplicative or jitter noise due to fluctuations in the apparatus; scintillation noise due to fluctuation in the object under scrutiny; missing or corrupted points; etc...). Different types of noise were tested on a synthetic dataset, (Supp Info figure S5) and it was found that rQRd is efficient in most of these situations.

4.5 Speed

A large processing time improvement for rQRd over SVD is observed in figure 3, with a speed-up of approximatively $\times 40$ for the largest datasets. This difference can be explained in two ways. First, the QR factorization step provides a large speed improvement when compared to the burdensome SVD decomposition. Moreover, the Y matrix on which it is applied is quite smaller than H in the typical case of a large data measurement in which the number of lines is much smaller than the number of acquired points ($K \ll L$).

Because of the FFT based implementation of the matrix products, urQRd presents an additional speed improvement, displaying a factor $\times 25$ over rQRd for the 64 000 points datasets. Moreover, memory requirements are much weaker and figure 3 presents results for interferograms with up to 4 096 000 complex points.

The observed processing time asymptotic behavior displays the expected trend, with $N^{2.1}$ for rQRd and $N^{1.1}$ for urQRd, to be compared with a dependence in $N^{2.9}$ for SVD. urQRd is slower for small datasets because the additional complexity dominates at lower sizes. Finally, it should be noted that because the FFT algorithm time is not regular on the vector length, the urQRd processing time reflects this irregularity in figure 3 where the processed lengths alternates between ($2^{n+3}\,5^3$) and ($2^{n+3}\,5^2\,7$) (multiples of 1000 and 1400).

4.6 Application to FT-ICR MS

FT-MS measures the frequencies of ions orbiting in an electric (Orbitrap [30]) or magnetic field (ICR). This is the MS technique with the highest resolution today, with $m/\Delta m$ over 1 000 000. FT-MS therefore knows a growing interest, in particular for proteomics, metabolomics and petroleomics [31, 32]. In these "*-omics*" studies, a large number of samples have to be processed rapidly, and the throughput of the processing techniques is thus a paramount parameter.

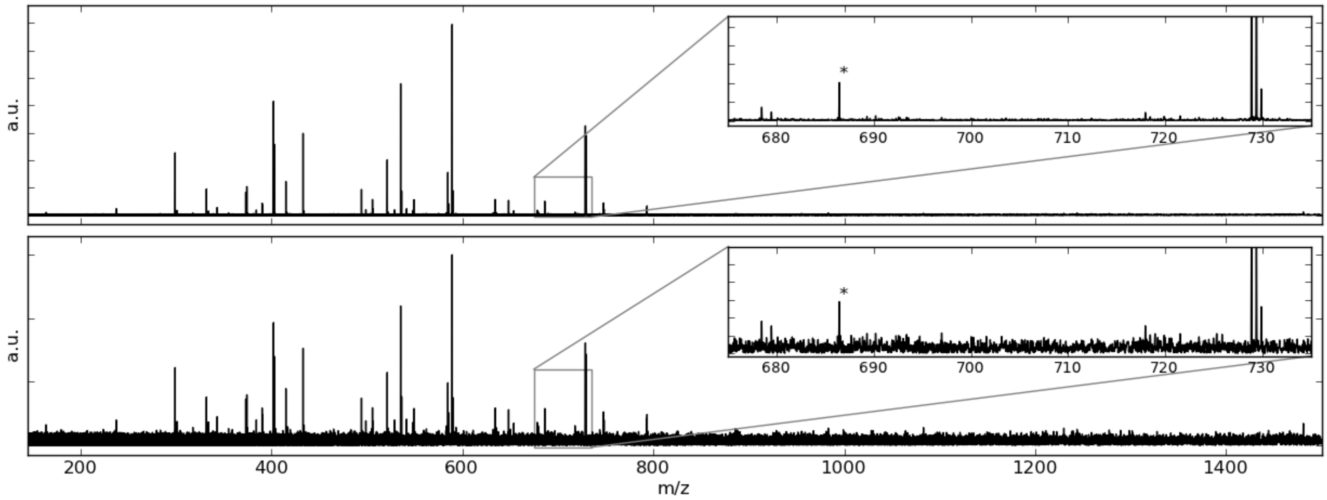

Figure 4: Processing of a single-scan FT-ICR mass spectrum of a trypsin digest of Cytochrome C. The experimental interferogram is 512k points. *Bottom* original spectrum, SNR measured on the *m/z* 728.8388 peak is 34.1 dB. *Top* same spectrum after urQRd processing ($K = 1000$), SNR measured on the *m/z* 728.8388 peak is 64.6 dB. *inset*) the *m/z* 728.8388 peak corresponds to the `TGQAPGFSTDANK`$^{2+}$ ion, *m/z* 678.3821 to `YIPGTK`+ and *m/z* 717.9012 to `GEREDLIAYLKK`$^{2+}$. The peak labeled with a star at *m/z*=686.390, lacking isotopic structure, is likely to be an experimental artifact.

Figure 4 shows a single-scan FT-ICR mass spectrum of a partial tryptic digest of cytochrome C, performed on a 9.4 T mass spectrometer. Because of the tryptic digestion, the sample contains many different peptides with a large range of masses and concentrations. Moreover, the dynamic range is further extended by the isotopic patterns which presents high intensity differences between the most and the least statistical probable isotopomers. As a consequence, the number of signals expected in the spectrum is unpredictable, and depends not only on the sample preparation, but also on the signal-to-noise ratio of the measure. A higher signal-to-noise ratio can be obtained with an increased number of scans. However, because of the coupling to a chromatographic system, it is of great interest to keep the total acquisition time to a minimum, thus improving the resolution along the chromatographic axis.

The experimental dataset consist in 512 k real points, regularly sampled at 1 MHz which are standard conditions for this spectrometry. The experiment was run here in one scan, corresponding to a total acquisition time of one second.

In addition to thermal electronic noise, high resolution FT-ICR experiments are characterized by ion cloud instabilities that generate frequency and phase instabilities [33], which produce a phase noise difficult to reduce with regular approaches. The principle of the AR model is to extract long-range correlations in the signal by the use of the matrix H. The order M of the model, which corresponds to the number of lines in H determines the longest correlations to be analyzed. The largest possible order value was chosen here so as to smooth out long range fluctuations. The size of datasets required for very high resolution precludes the use of standard noise reduction approaches, and urQRd seems to be unequaled here.

4.7 Application to 2D FT-ICR

FT-MS opens itself up to multidimensional techniques [34–36]. The 2D FT-ICR spectrometry was introduced as early as 1987 [34], but laboratory use of this promising technique has been hampered by the amount of data which needs to be stored and processed. Only now, with the increased power of computers, can this technology proves its use in particular in proteomics or metabolomics (see [37] for a recent review). 2D FT-ICR MS maps the fragmentation patterns of ions in complex samples without prior ion isolation and affords important structural information. However, fluctuations of the ion number in the ICR cell is an additional noise source which causes considerable scintillation noise and calls for denoising [38]. Figure 5 presents a 2D FT-ICR mass spectrum for metabolomics analysis. The Cadzow approach has shown satisfactory results [39], but is prohibitively costly in terms of computer capacity and computing times. The use of rQRd resulted in an equivalent denoising quality, albeit at a drastically reduced cost in processing times. Indeed the computation was carried out within a few hours on a desktop computer instead of one week on a departmental cluster for the SVD approach.

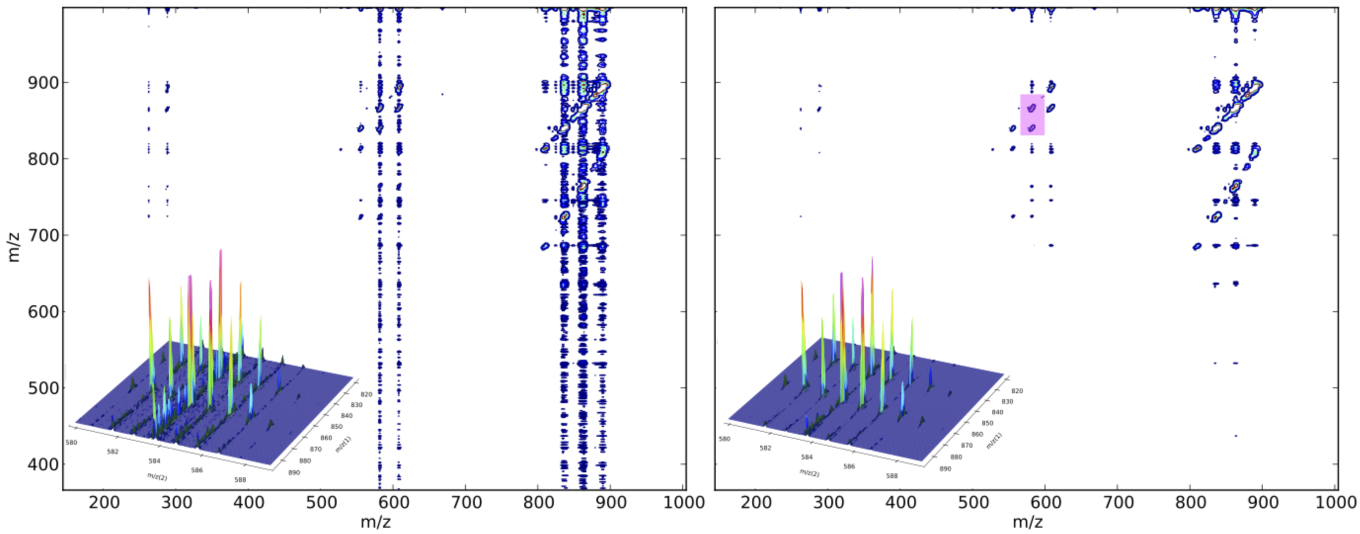

Figure 5: 2D IRMPD FT-ICR MS spectrum of triacylglycerols (TAG) extracted from human plasma showing strong scintillation noise. The dataset is 2k $\times$ 128k points. *inset*) zoom on the pattern centered at *m/z*(F1) 845 and *m/z*(F2) 584 (highlighted in pink). The two groups of peaks give the isotopic patterns of lithiated TAG(16 :0/16 :0/18 :1) at *m/z* 839.7674 and lithiated TAG(16 :0/18:1/18 :1) at *m/z* 865.7831 respectively loosing a palmitic acid (MW 256.2396) and an oleic acid (MW 282.2553) in order to yield a lithiated diacylglycerol DAG(16:0/18:1) at *m/z* 583.5278(33). SNR was measured on the zoomed zone to 22.2 dB and 42.8 dB for the standard and denoised datasets respectively.

5 Conclusion

An efficient separation between noise and genuine signal is usually associated with the requirement of a model of the observed phenomenon. However, the choice of an appropriate model is often problematic since slight deviations from the model may lead to data misinterpretation. This drawback is circumvented here by the use of random projection that allows the extraction of weak long-range correlations together with the preservation of some freedom in the data analysis. Based on this approach, we propose here a robust denoising procedure much faster than classical SVD and easily performed in a few minutes on a standard desktop computer with data sizes over one million points. Randomness, in a counterintuitive manner, provides us with a fast and robust approach to denoising.

The procedure depends depends on a single user parameter K, which is related to the expected number of frequencies. However, in contrast to SVD, the result of the denoising procedure is only marginally dependent on this parameter, provided that it is significantly higher that the number of signals. Iterating the procedure improves the quality of the denoising and the robustness against the value of K.

The two algorithms, rQRd and urQRd are equivalent in terms of results, but differ in their implementation and processing speed. We showed that the application of urQRd denoising procedure on an experimental FT-ICR MS 512k points interferogram allows a noise reduction of about 30 dB (a factor of 30) in less than an hour on a desktop computer with a non-optimized implementation. We demonstrated that both methods are valuable tools in the field of high resolution spectroscopy, where very large datasets are corrupted by various sources of noise.

Acknowledgments

The authors gratefully thank Fabrice Bray (Université Lille 1, Sciences et Technologies) for providing the 2D FT-ICR-MS dataset.

This work was supported by the French Infrastructure for Integrated Structural Biology (FRISBI ANR-10-INSB-05-01), by Instruct, part of the European Strategy Forum on Research Infrastructures (ESFRI), by the Agence Nationale de la Recherche (grant 2010-FT-ICR-2D) and by the MASTODONS project by CNRS (grant 2013-MesureHD). M.v.A. thanks the Région Nord-Pas-de-Calais for postdoctoral funding. The FT-ICR mass spectrometer and the proteomics platform used for this study are funded by the European Community (FEDER), the Région Nord-Pas-de-Calais (France), the IBISA network, the CNRS, and Université Lille 1, Sciences et Technologies, and this funding is gratefully acknowledged.

References

[1] Eisenstein DJ, et al. (2008) SDSS III: Massive spectroscopic surveys of the distant universe, the Milky Way galaxy, and extra-solar planetary systems. *Astron J* 142:72.

[2] Brumfiel G (2011) High-energy physics: down the petabyte highway. *Nature* 469:282–283.

[3] Schadt EE, Linderman MD, Sorenson J, Lee L, Nolan GP (2010) Computational solutions to large-scale data management and analysis. *Nat Rev Genet* 11:647–657.

[4] DiLeo MV, Strahan GD, den Bakker M, Hoekenga OA (2011) Weighted correlation network analysis (WGCNA) applied to the tomato fruit metabolome. *PLoS ONE* 6:e26683.

[5] Aggarwal CC, Han J, Wang J, Philip SY (2007) in *Data Streams* (Springer), pp 9–38.

[6] Rajaraman A, Ullman JD (2010) *Mining of massive datasets* (Cambridge University Press, Stanford University).

[7] Donoho DL (1995) De-noising by soft-thresholding. *IEEE Trans Inf Theory* 41:613–621.

[8] Prony G (1795) Essai Expérimental et Analytique: sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'alkool, à différentes températures. *J. de l'Ecole Polytechnique (Paris)* 1:24–76.

[9] Beylkin G, Monzon L (2005) On approximations of functions by exponential sums. *Appl Comput Harmon Anal* 19:17–48.

[10] Beylkin G, Monzon L (2010) Approximation by exponential sums revisited. *Appl Comput Harmon Anal* 28:131–149.

[11] Makhoul J (1975) Linear prediction: a tutorial review. *Proc IEEE* 63:561–580.

[12] Koehl P (1999) Linear prediction spectral analysis of NMR data. *Prog Nucl Magn Reson Spectrosc* 34:257–299.

[13] Golub GH, Loan. CFV (1996) *Matrix Computations* (Johns Hopkins University Press, Baltimore).

[14] Cadzow JA (1988) Signal enhancement-A composite property mapping algorithm. *IEEE Trans Acoust, Speech, Signal Processing* 36:49–62.

[15] Golyandina N, Nekrutkin V, Zhigljavsky A (2001) *Analysis of time series structure: SSA and related techniques* (Chapman & Hall/CRC, New York).

[16] Simon HD (1984) The Lanczos algorithm with partial reorthogonalization. *Math Comp* 42:115–142.

[17] Browne K, Qiao S, Wei Y (2009) A Lanczos bidiagonalization algorithm for Hankel matrices. *Linear Algebra Appl* 430:1531–1543.

[18] Cai JF, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. *SIAM J Optimiz* 20:1956–1982.

[19] Martinsson P, Rokhlin V, Tygert M (2006) A randomized algorithm for the approximation of matrices. (Yale CS research report YALEU/DCS/RR-1361), Technical report.

[20] Oropeza V, Sacchi M (2011) Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis. *GEOPHYSICS* 76:25–32.

[21] Johnson WB, Lindenstrauss J (1984) Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics* 26:189–206.

[22] Drineas P, Kannan R, Mahoney MW (2006) Fast Monte Carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM J Comput* 36:158–183.

[23] Liberty E, Woolfe F, Martinsson PG, Rokhlin V, Tygert M (2007) Randomized algorithms for the low-rank approximation of matrices. *Proc Natl Acad Sci USA* 104:20167–20172.

[24] Halko N, Martinsson PG, Tropp JA (2011) Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev* 53:217–288.

[25] Frieze A, Kannan R, Vempala S (2004) Fast Monte-Carlo algorithms for finding low-rank approximations. *J ACM* 51:1025–1041.

[26] Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) *Automatic subspace clustering of high dimensional data for data mining applications*, SIGMOD '98 (ACM, New York, NY, USA), pp 94–105.

[27] Achlioptas D (2003) Database-friendly random projections. *J Comput System* 66:671–687.

[28] Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. *Psychometrika* 1:211–218.

[29] Korobeynikov A (2010) Computation- and space-efficient implementation of SSA. *Statistics and Its Interface* 3:357–368

[30] Hu Q, et al. (2005) The Orbitrap: a new mass spectrometer. *J Mass Spectrom* 430–443.

[31] Villas-Bôas SG, Mas. S, Åkesson M, Smedsgaard J, Nielsen J (2005) Mass spectrometry in metabolome analysis. *Mass Spectrom Rev* 24:613–646.

[32] Marshall AG, Rodgers RP (2008) Petroleomics: chemistry of the underworld. *Proc Natl Acad Sci USA* 105:18090–18095.

[33] Aizikov K, O'Connor PB (2006) Use of the filter diagonalization method in the study of space charge related frequency modulation in Fourier transform ion cyclotron resonance mass spectrometry. *J Am Soc Mass Spectrom* 17:836–843.

[34] Pfändler P, Bodenhausen G, Rapin J, Houriet R, Gäumann T (1987) Two-dimensional Fourier transform ion cyclotron resonance mass spectrometry. *Chem Phys Lett* 138:195–200.

[35] McLafferty FW, Stauffer DB, Loh SY, Williams E (1987) Hadamard transform and "no-peak" enhancement in measurement of tandem Fourier transform mass spectra. *Anal Chem* 59:2212–2213.

[36] Ross CW, Guan S, Grosshans PB, Ricca TL, Marshall AG (1993) Two-dimensional Fourier transform ion cyclotron resonance mass spectrometry/mass spectrometry with stored-waveform ion radius modulation. *J Am Chem Soc* 115:7854–7861.

[37] van Agthoven MA, Delsuc MA, Bodenhausen G, Rolando C (2013) Towards analytically useful two-dimensional Fourier transform ion cyclotron resonance mass spectrometry. *Anal Bioanal Chem* 405:51–61.

[38] van der Rest G, Marshall AG (2001) Noise analysis for 2D tandem Fourier transform ion cyclotron resonance mass spectrometry. *Int J Mass Spectrom* 210:101–111.

[39] van Agthoven MA, Chiron L, Coutouly MA, Delsuc MA, Rolando C (2012) Two-dimensional ECD FT-ICR mass spectrometry of peptides and glycopeptides. *Anal Chem* 84:5589–5595.

Efficient denoising of very large experimental datasets. Application to FT-ICR mass spectrometry. Supporting Information

L. Chiron[1] M.A. van Agthoven[2] B. Kieffer[1] C. Rolando[2] M-A. Delsuc[1]

1 Institut de Génétique et de Biologie Moléculaire et Cellulaire, Institut National de la Santé et de la Recherche, U596; Centre National de la Recherche Scientifique, Unité Mixte de Recherche 7104; Université de Strasbourg, 67404 Illkirch-Graffenstaden, France

2 Miniaturisation pour la Synthèse, l'Analyse et la Protéomique, Centre National de la Recherche Scientifique, Unité de Service et de Recherche 3290, and Protéomique, Modifications Post-traductionnelles et Glycobiologie, Université de Lille 1 Sciences et Technologies, 59655 Villeneuve d'Ascq Cedex, France Glycobiologie, IFR 147, Université de Lille 1 Sciences et Technologies, 59655 Villeneuve d'Ascq Cedex, France

1 Algorithms

The algorithms for **rQRd** and **urQRd** are given here in formal representation.

Algorithm S1 rQRd

given a time series X*, rank* K *and order* M*, returns* $\tilde{X}$ *a denoised approximation of* X

Require: $X, K, M \quad K \leq M \leq length(X)/2$

Require: Function $\textsc{Random} : n, p \mapsto \Omega$ ▷ Ω a $\mathcal{N}(0,1)$ $n \times p$ matrix

Require: Function QR : $A \mapsto Q, R$ ▷ the QR decomposition of A

```
L ← LENGTH(X)
N ← L − M + 1
for i ← 1, M   j ← 1, N do
    H_ij ← X_{i+j−1}                    ▷ H is a M × N matrix
end for
Ω ← RANDOM(N, K)
Y ← HΩ
(Q, R) ← QR(Y)
H̃ ← QQ*H
for l ← 1, L do
    X̃_l ← ⟨H_ij⟩_{i+j=l+1}
end for
return X̃
```

The largest objects stored in memory are the matrices H and $\tilde{H}$. This represents a memory burden proportional to $O(MN) \lesssim O(L^2)$.

The slowest step is the computation of $\tilde{H} = QQ^*H$ in $O(KMN)$ while the computation of $\tilde{X}$ is in $O(LM)$. This results in a theoretical time dependence in $O(KMN + LM)$ The initial computation of Y is also non-negligible, but in all cases the computation of the QR decomposition seems to be negligible in our implementation.

Algorithm S2 Fast Hankel Matrix product

Require: Function $\mathcal{F} : f_i \mapsto F_j$, $\quad \triangleright$ compute F_j the Digital Fourier Transform of f_i

function $\textsc{Fhv}(X, V)$

given a time series X, and a vector V, returns the result of the matrix product of H by V, where H is the Hankel matrix constructed from X as in Algorithm S1

$L \leftarrow \textsc{length}(X)$

$N \leftarrow \textsc{length}(V)$

$W \leftarrow \{ \underbrace{0, \ldots, 0}_{M-1 \text{ values}}, V_N, V_{N-1}, \ldots, V_1\}$ $\quad \triangleright$ so that length of W is L

$X' \leftarrow \mathcal{F}(X)$

$W' \leftarrow \mathcal{F}(W)$

$S' \leftarrow \{X'_1 W'_1, \ldots, X'_L W'_L\}$

$S \leftarrow \mathcal{F}^{-1}(S')$

$R \leftarrow \{S_1, \ldots, S_{L-N}\}$

return R

end function

function $\textsc{Fhm}(X, A)$

given a time series X, and a matrix A, returns the result of the matrix product of H by A, where H is the Hankel matrix constructed from X as in Algorithm S1

$N, P \leftarrow \textsc{Shape}(A)$

for $p \leftarrow 1, P$ **do**

$A^{(p)} \leftarrow \{A_{1,p}, \ldots, A_{N,p}\}$

$B^{(p)} \leftarrow \textsc{Fhv}(X, A^{(p)})$

end for

return matrix B where $B_{i,j} = B_j^{(i)}$ $\quad \triangleright$ B is a $M \times P$ matrix

end function

Function $\textsc{Fhv}()$ is in $O(L \log(L))$ and function $\textsc{Fhm}()$ is in $O(NL \log(L))$.

The $\textsc{Fhm}()$ function can be further optimized by allowing the vector S' computed in $\textsc{Fhv}()$ to be stored between each call, rather than recomputed.

Algorithm S3 urQRd

given a time series X, rank K and order M, returns $\tilde{X}$ a denoised approximation of X

Require: $X, K, M \quad K \leq M \leq length(X)/2$

Require: Function $\textsc{Random} : n, p \mapsto \Omega$ $\qquad \triangleright$ a $\sim \mathcal{N}(0,1)$ $n \times p$ matrix

Require: Function $\mathrm{QR} : A \mapsto Q, R$ $\qquad \triangleright$ the QR decomposition of A

Require: Function $\textsc{Fhv} : H, M, X \mapsto Y$

Require: Function $\textsc{Fhm} : H, M, A \mapsto B$

$L \leftarrow \textsc{length}(X)$

$N \leftarrow L - M + 1$

$\Omega \leftarrow \textsc{Random}(N, K)$

$Y \leftarrow \textsc{Fhm}(X, \Omega)$

$(Q, R) \leftarrow \mathrm{QR}(Y)$

$U \leftarrow \left[\ \textsc{Fhm}(X, Q^*)\right]^*$

for $k \leftarrow 1, K$ **do**

$\quad Q^{(k)} \leftarrow \{Q_{1,k}, \ldots, Q_{M,k}\}$

$\quad U'^{(k)} \leftarrow \{U_{k,N}, U_{k,N-1}, \ldots, U_{k,1}\}$

$\quad W^{(k)} \leftarrow \{\ \underbrace{0, \ldots, 0}_{N-1 \text{ values}}, Q_1^{(k)}, \ldots, Q_M^{(k)}, \underbrace{0, \ldots, 0}_{N-1 \text{ values}}\ \}$ $\qquad \triangleright$ $W^{(k)}$ are of length $L + N - 1$

$\quad Z^{(k)} \leftarrow \textsc{Fhv}(W^{(k)}, U'^{(k)})$ $\qquad \triangleright$ $Z^{(k)}$ are of length L

end for

$Z \leftarrow \sum_{k=1}^{K} Z^{(k)}$

for $l \leftarrow 1, L$ **do**

$$\tilde{X}_l \leftarrow \alpha_l Z_l \quad \text{with } \alpha_l = \begin{cases} 1/l & 1 \leq l \leq M \\ 1/M & M < l < N \\ 1/(L - l + 1) & N \leq l \leq L \end{cases}$$

end for

return $\tilde{X}$

The largest objects stored in memory are the matrices Y, Q and U. This represents a total memory burden proportional to $O(KL)$.

The slowest step is the loop on K for the computation of $\tilde{X}$ and its processing time is proportional to $O(KL\log(L))$. The computation of Y and of U are also non-negligible, but in all cases the computation of the QR decomposition seems to be negligible in our implementation.

2 Robustness against varying signal distortion

A synthetic dataset was used to test the robustness of **rQRd** relatively to various types of noise. A noise-free signal containing 20 random lines with intensities ranging from 1 to 20, is created and perturbed with a random process. In all cases, the random series is stationary with a Gaussian distribution, zero mean, and white Fourier Transform. It is obtained from the `numpy.random` library. For each realization, the perturbation level was chosen so that the apparent noise in the Fourier spectrum is approximately of the same intensity level. **rQRd** analysis is performed with $K = 50$.

Signal modifications are as follows:

- *additive noise* : a random signal is added the noise-free dataset. This is the case explicitly considered in the theoretical section.
- *scintillation noise* : the amplitude and the frequency of each signal component are subject to random variation of their value.
- *sampling noise* : each point of the series used to sample the theoretical signal is displaced by a random amount.
- *missing points* : some randomly chosen points of the signal series are set at 0.0

In all cases, the noise level is such that the SNR of noisy dataset is around 0 dB; except for the sampling case, where the SNR is 3 dB. All details can be found in the code deposited on the web site urqrd.igbmc.fr.

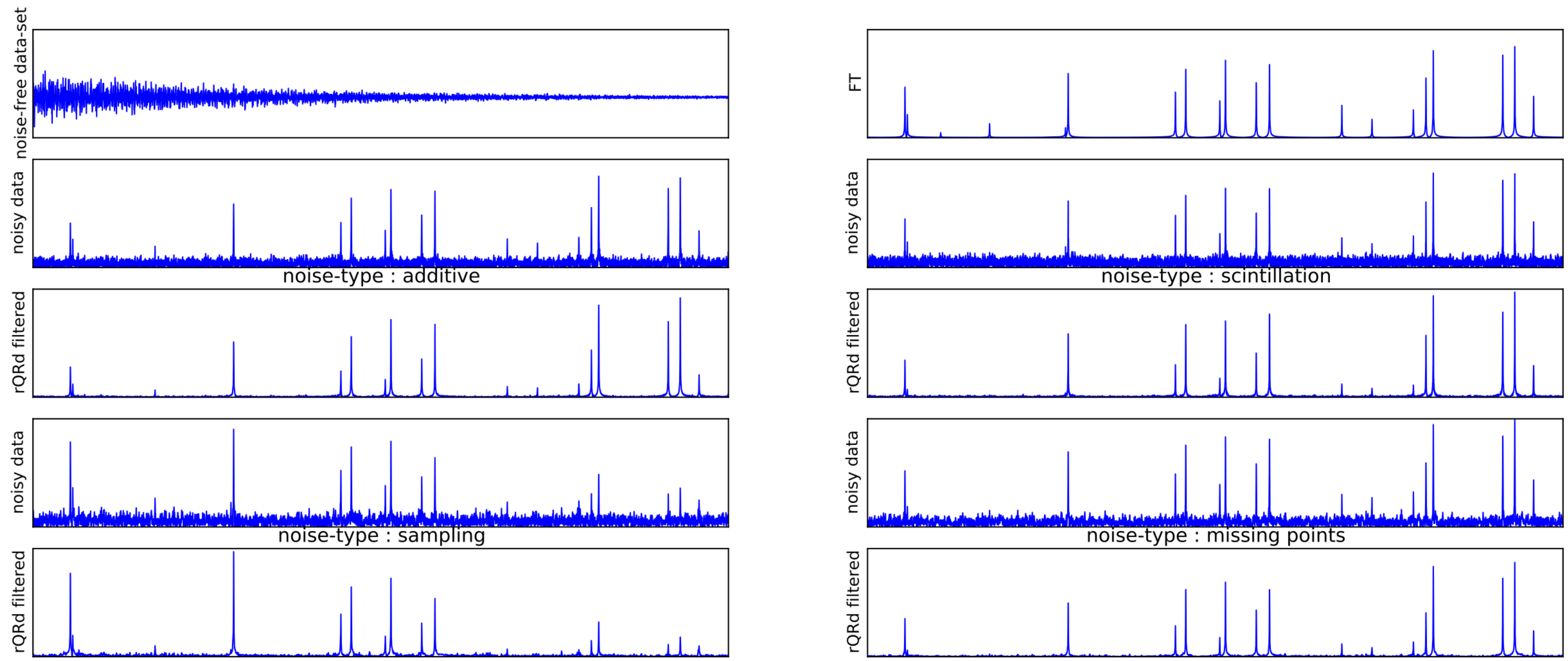

Figure S4: **rQRd** efficiency for various noise types on a synthetic datasets.
Topline the noise-free temporal signal and its Fourier transform.
Left column *second and third row* additive noise *forth and fifth row* sampling noise
Right column *second and third row* scintillation noise *forth and fifth row* missing point

3 Code and Data Deposition

3.1 Data Deposition

The data has been deposited on the site urqrd.igbmc.fr

3.2 Code Deposition

The code has been deposited on the site urqrd.igbmc.fr