Supplementary Information: Monitoring polydispersity by NMR diffusometry with tailored norm regularisation and moving-frame processing

Mateusz Urbańczyk,[a,d] Diana Bernin,[b] Alan Czuroń,[c] and Krzysztof Kazimierczuk *[d]

1 Iteratively Re-weighted Least Squares with ℓ_p-norm

We have chosen the iteratively re-weighted least squares (IRLS) algorithm [1–3] as a method to implement the ILT regularised by ℓ_p-norm with arbitrary p. The penalty function with ℓ_p-norm can be described as a pseudo-ℓ_2-norm regularization

$$\|A\|_{\ell_p}^p = \sum_i w_i \cdot |A_i|^2, w_i = |A_i|^{p-2}, \tag{SI.1}$$

and thus the Equation 8. can be written as a least squares problem

$$\min_{A \geq 0} \|\Phi A - \Psi\|_{\ell_2}^2 + \|WA\|_{\ell_2}^2, \tag{SI.2}$$

where W is the diagonal matrix $\mathbf{W}_{ii} = \tau w_i$. In order to avoid division by zero, the weights are regularised as previously described [1]. W depends on A and the problem is solved iteratively with W set in the kth iteration based on A from iteration $k-1$ [1]. The solution in each iteration takes the following form

$$A = W^{-1}\Phi^{-1}\left(\Phi W^{-1}\Phi^{-1} + \mathbb{I}\right)^{-1}\Psi. \tag{SI.3}$$

2 Matlab Code

2.1 Functions

```
function [NormResiduum, Result,Pnormlist]=TailoredNorm(Decay, K, D_grid,options)
%Input:
%Decay=the vector of the Diffusion Decay from the PGSE experiment
%K  vector of k in eq I=exp(-D*k)
%D_grid - diffusion scale vector
%options.tau - proportion between first and second term
%options.epsilon - regularisation of weight to avoid diving by zero
%options.no_of_iterations=number of iterations
%Output:
%NormResiduum - The signal reconstruction residuum for different p.
%Result - the matrix of the reconstruction for different p.
%List of p
if isfield(options,'no_of_iterations')
    no_of_iterations = options.no_of_iterations;
else
    no_of_iterations = 1e2;
end
if isfield(options,'tau')
    tau = options.tau;
else
    tau = 2e-6;
end
if isfield(options,'epsilon')
    epsilon = options.epsilon;
else
    epsilon = 1e-2;
end


```

[a] Faculty of Chemistry, Biological and Chemical Research Centre, University of Warsaw Żwirki i Wigury 101, 02-089 Warszawa, Poland
[b] Swedish NMR Centre, University of Gothenburg, BOX 465, 40530 Göteborg, Sweden
[c] Institute of Mathematics Polish Academy of Sciences ul. Śniadeckich 8, 00-656 Warszawa, Poland
[d] Centre of New Technologies University of Warsaw ul. Banacha 2C, 02-097 Warsaw, Poland, E-mail:k.kazimierczuk@cent.uw.edu.pl

```
30
31     for i=1:21
32           Pnorm=1+((i-1)/20);
33           Pnormlist(i)=Pnorm;
34           matrix=ILTmatrix(D_grid,K);
35           A=IRLS(no_of_iterations,Decay',matrix,tau,Pnorm,epsilon);
36           PsiA=matrix*A;
37           NormResiduum(i)=norm(PsiA-Decay');
38
39           Result(:,i)=A';
40     end
41 end
42 function [A]=IRLS(no_of_iterations,Decay,matrix,tau,Pnorm,epsilon);
43     ILT=ctranspose(matrix);
44     fn=size(matrix,2);
45     A=ILT*Decay;
46     L=eye(size(matrix,1));
47     k=0;
48     while (k<no_of_iterations)
49         if(Pnorm==0)
50             tau=(k+1)/Niter;
51             W=(1./tau)*diag(abs(A).^(1.+tau*1.)+eps^(1.+tau*1.));
52         elseif(Pnorm==1)
53             W=(1./tau)*diag(abs(A)+eps);
54         else
55             W=(1./tau)*diag(abs(A+eps).^(2.-Pnorm));
56         end
57         G2=matrix*W;
58         B=G2*ILT+L;
59         A=G2'*pinv(B,2e8)*Decay;
60         A(A<0)=0;
61         k=k+1;
62     end
63 end
64 function [matrix]=ILTmatrix(D,K)
65     for i=1:max(size(D))
66         u=0;
67         for j=1:max(size(K))
68             u=u+1;
69             matrix(j,i)=exp(-(D(i))*K(u));
70         end
71     end
72 end
```

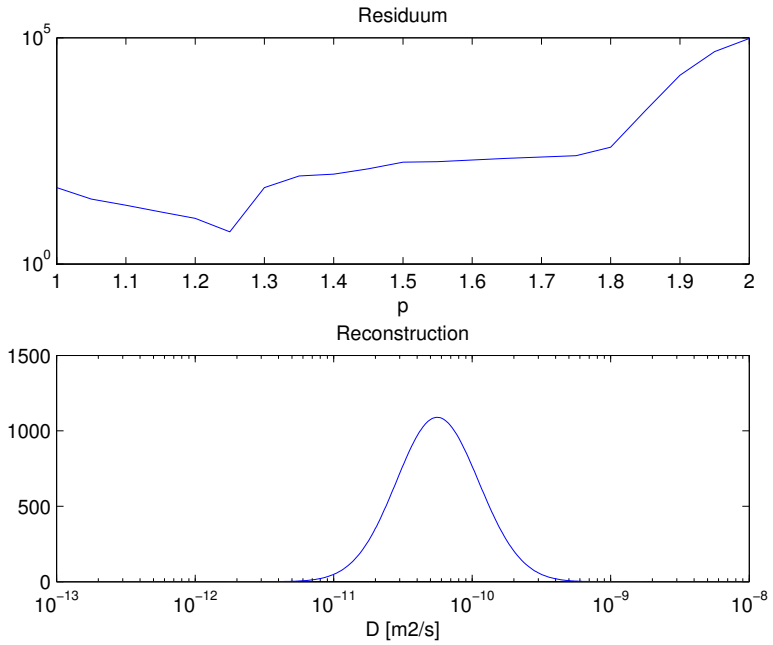
2.2 Example

2.2.1 Input

```
1  K=[27969939.3652834, 31850023.2544088, 36268365.4068087, 41299634.8157993, 47028858.8081274, 53552859.9867592,
       60981892.5111957, 69441505.3680887, 79074664.1210091, 90044166.9964695, 102535396.137508,
     116759450.520409, 132956713.480145, 151400915.134930, 172403758.363937, 196320186.516186,
     223554381.873686, 254566596.241519, 289880929.100324, 330094184.770193, 375885958.270086,
     428030120.321524, 487407895.590545, 555022755.187300, 632017375.103181, 719692947.179286,
     819531168.957255, 933219284.036046, 1062678596.11106, 1210096938.57650, 1377965648.41998,
     1569121669.25994, 1786795495.05772, 2034665764.74233, 2316921430.38496, 2638332549.55113,
     3004330898.20593, 3421101766.51175, 3895688488.84754, 4436111474.58309, 5051503751.20198,
     5752265310.42164, 6550238865.72526, 7458909991.56866, 8493635026.56977, 9671900592.19377,
     11013619112.7413, 12541465330.8620, 14281259505.6288, 16262403769.1372, 18518379015.9553,
     21087310722.7473, 24012613260.2947, 27343723586.6444, 31136936720.6321, 35456357115.1003,
     40374982007.7996, 45975935058.3678, 52353871119.5689, 59616575883.1166, 67886787433.7365,
     77304270495.3216, 88028178423.4806, 100239742861.620];
2  Decay=[83404.0250362637, 83370.4506110911, 83414.2624602818, 83388.1520260110, 83304.3180159704,
     83243.6787060199, 83266.1546335731, 83173.5537480279, 83097.7778532905, 83073.3193017468,
     82949.3092563797, 82867.8989459299, 82812.0302292043, 82735.4195567696, 82549.7746801276,
     82406.5856119987, 82238.7428236193, 82121.2059939558, 81913.6666997866, 81693.1048290188,
     81380.3073543495, 81113.5839397958, 80730.0865341128, 80355.2098122292, 79945.3149460123,
     79470.9537684617, 78891.9542316392, 78327.3319010280, 77623.2533430469, 76847.2101863967,
     76014.4953412066, 74977.0495917831, 73939.0503716736, 72753.3251709937, 71379.0586581385,
     69883.7855109229, 68258.5820909276, 66481.5549042536, 64566.1430687698, 62453.4587668726,
     60184.5552300506, 57764.2266509588, 55151.9222561379, 52356.4896937735, 49459.5467690467,
     46415.9046967481, 43304.1816497189, 40046.6076481351, 36774.7720285385, 33494.2954637877,
```

```
    30222.6859124238, 26960.5773408252, 23845.5492726671, 20890.9059669158, 18101.3940385555,
    15414.2930568451, 13032.9057574235, 10800.9105006795, 8855.07954389013, 7198.32791408085,
    5748.46760777400, 4492.65667480655, 3467.55054003967, 2632.90241791157];
options.tau=1e-7;
options.epsilon=1e-3;
no_of_iterations=1e2;
D_grid=logspace(-13,-8,512);
[NormResiduum,  Result, Pnormlist]=TailoredNorm(Decay,  K,  D_grid, options);
subplot(2, 1, 1)
semilogy(Pnormlist, NormResiduum)
title('Residuum')
xlabel('p')
[minimuum, index]=min(NormResiduum);
subplot(2, 1, 2)
semilogx(D_grid, Result(:,index))
title('Reconstruction')
xlabel('D [m2/s]')
```

2.2.2 Output



3 Robustness to (mis)setting of τ

Fig. SI.1 Test of the robustness to the (mis)setting of τ. Upper panels show the residuum and the corresponding $A(D)$ reconstructed with optimal norm is found below. Solid line - reconstruction with optimal ℓ_p-norm, dashed line - simulated profile.

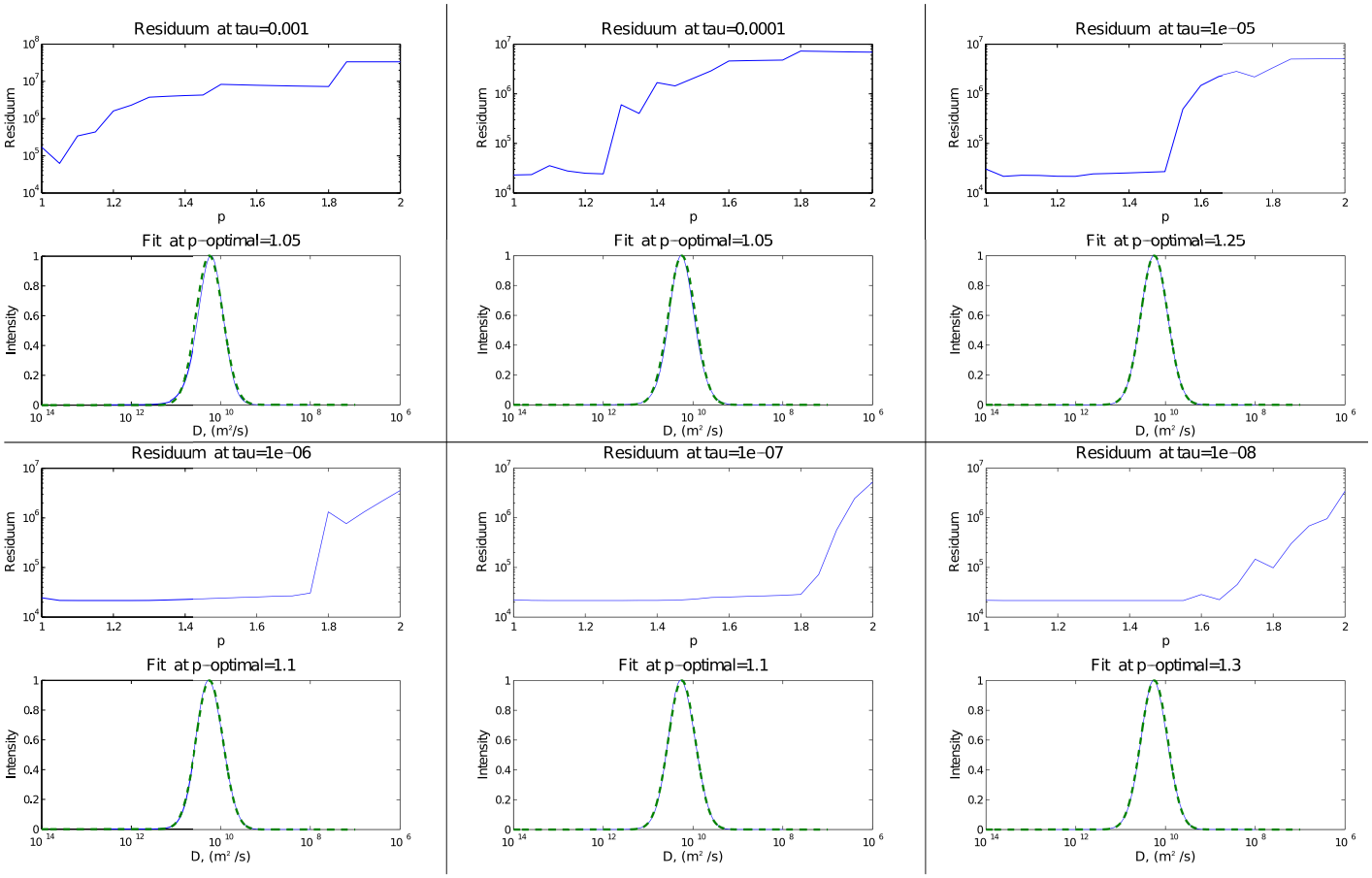
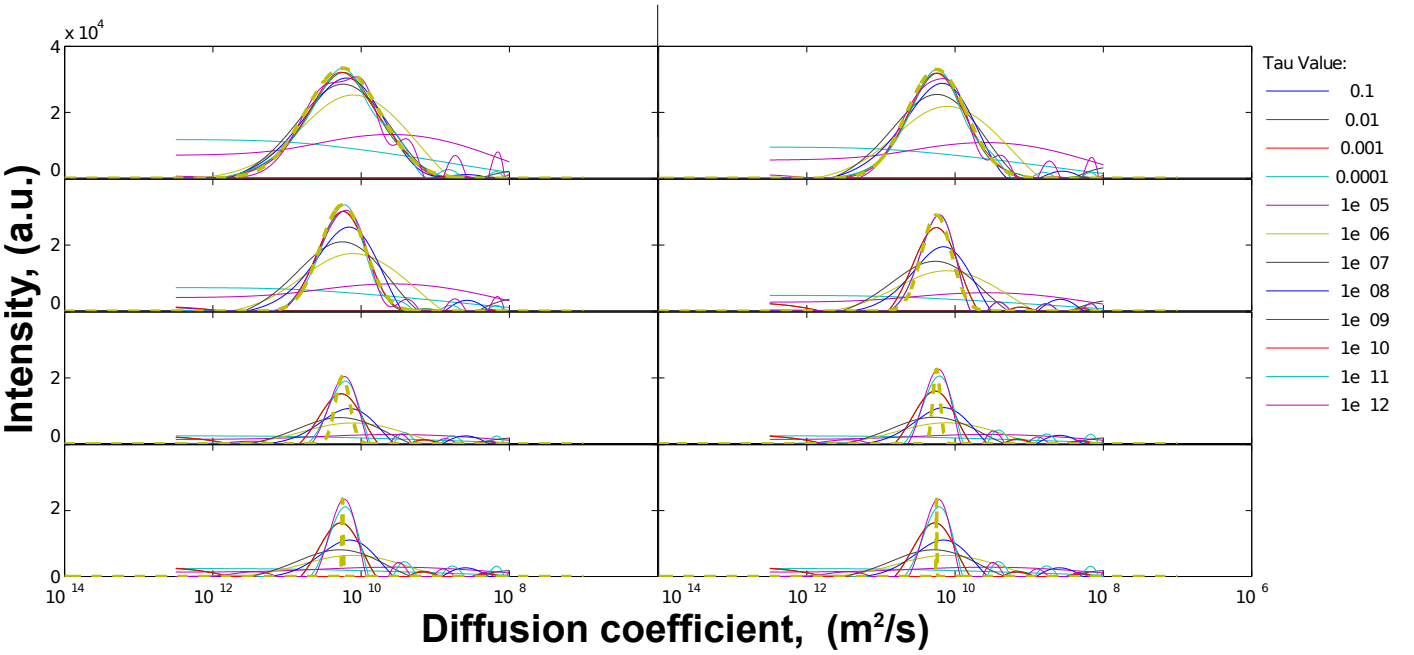


Fig. SI.2 Behavior of ℓ_2-norm regularisation with different values of τ for peaks of different polydispersity. As mentioned in main text, the ℓ_2-norm regularisation with choosing optimal τ is not efficient for narrow diffusion peaks and over-smooths them. This can be explained by inherent smoothing features of ℓ_p-norms ($p > 1$) - see below.



4 Non-unimodal/asymmetric distributions

We compared the Tailored Norm regularisation with Trust-Region Algorithm for the Inversion (TRAIn)[4]. The authors of TRAIn claim that the algorithm is specially suited for non-symmetrical distribution of diffusion coefficient. We have compared the simulated datasets with different asymmetrical distributions (Fig. SI.3) and with bimodal distributions (Fig. SI.4). Additionally, we compared the TRAIn reconstruction with our method for the case of heparin depolymerisation monitoring (Fig. SI.5). For the simulation we have used the MATLAB code of TRAIn taken from the Supporting Information of the publication that introduced the method[4]. All parameters were set to the values recommended by the authors.

The analysis of the results shows that Tailored Norm reconstructs the asymmetrical distributions equally good as TRAIn. Additionally, it is less vulnerable to noise. As for the bimodal distributions the Tailored Norm fails in the situation of two peaks with distinctively different polydispersity, as it cannot find the p that will be optimal for both peaks. It is worth mentioning that TRAIn method reconstructs such distribution very well, but only in the noiseless case. In case of the heparin degradation studies the noise vulnerability of the TRAIn method is even more evident (Figure SI.5).

Fig. SI.3 Comparison of Tailored Norm (yellow) and TRAIn (green) on simulations of asymmetrical distributions. Reference is shown in cyan. Each simulation is based on equally spaced 4 Gaussian peaks in the intensity ratio: 4:3:2:1. The distance between the peaks is increasing linearity with the simulation's number. For each simulation the reconstruction of the noiseless signal and signal with addition of white noise at the level of 0.1% of the first data point was performed.

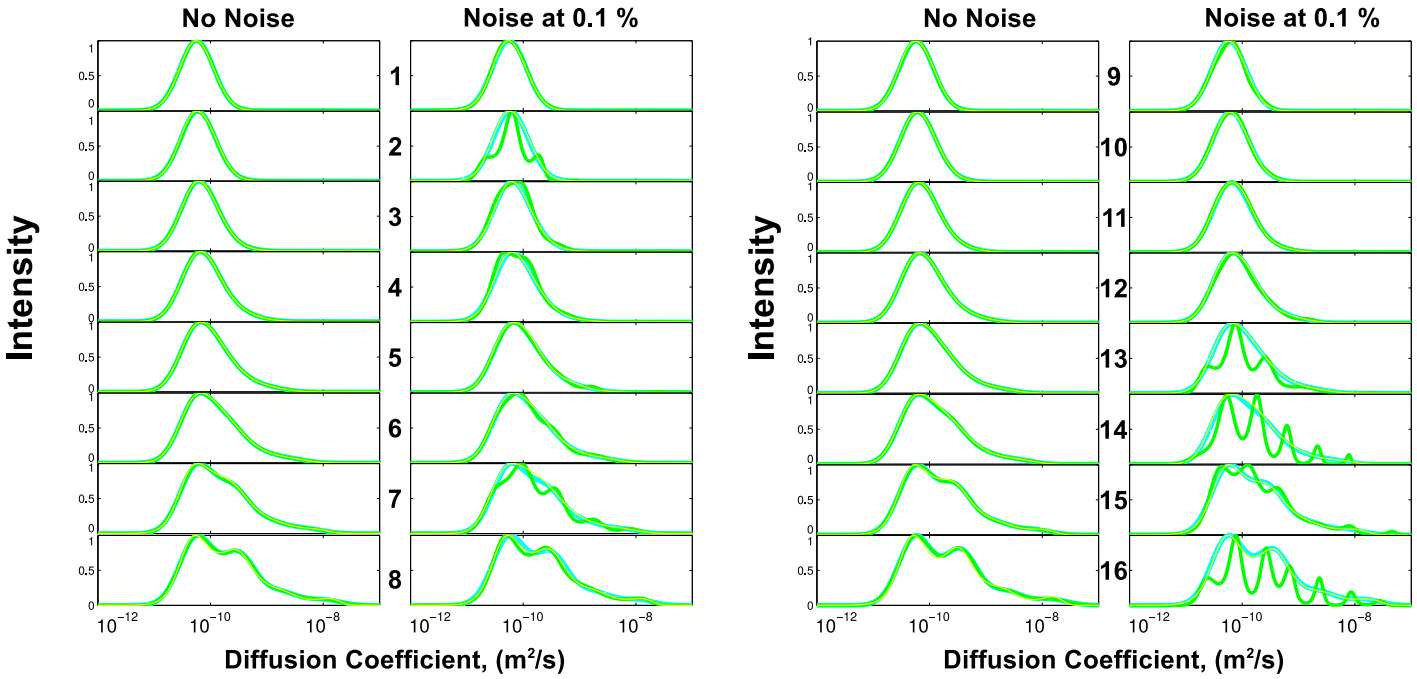


Fig. SI.4 Comparison of Tailored Norm (red-dotted) and TRAIn (green) on simulations of bimodal distributions. Reference is shown in cyan A- Two peaks with $\sigma_1 = 0.2$ and $\sigma_2 = 0.2$ centered at $D_1 = 10^{-10.75}\frac{m^2}{s}$ and $D_2 = 10^{-9.25}\frac{m^2}{s}$ B- Two peaks with $\sigma_1 = 0.1$ and $\sigma_2 = 0.2$ centered at $D_1 = 10^{-10.75}\frac{m^2}{s}$ and $D_2 = 10^{-9.25}\frac{m^2}{s}$ C- Two peaks with $\sigma_1 = 0.2$ $\sigma_2 = 0.4$ and centered at $D_1 = 10^{-10.75}\frac{m^2}{s}$ and $D_2 = 10^{-9.25}\frac{m^2}{s}$ For each simulation the reconstruction of the noiseless signal and signal with addition of white noise at the level of 0.1% of the first data point was performed.

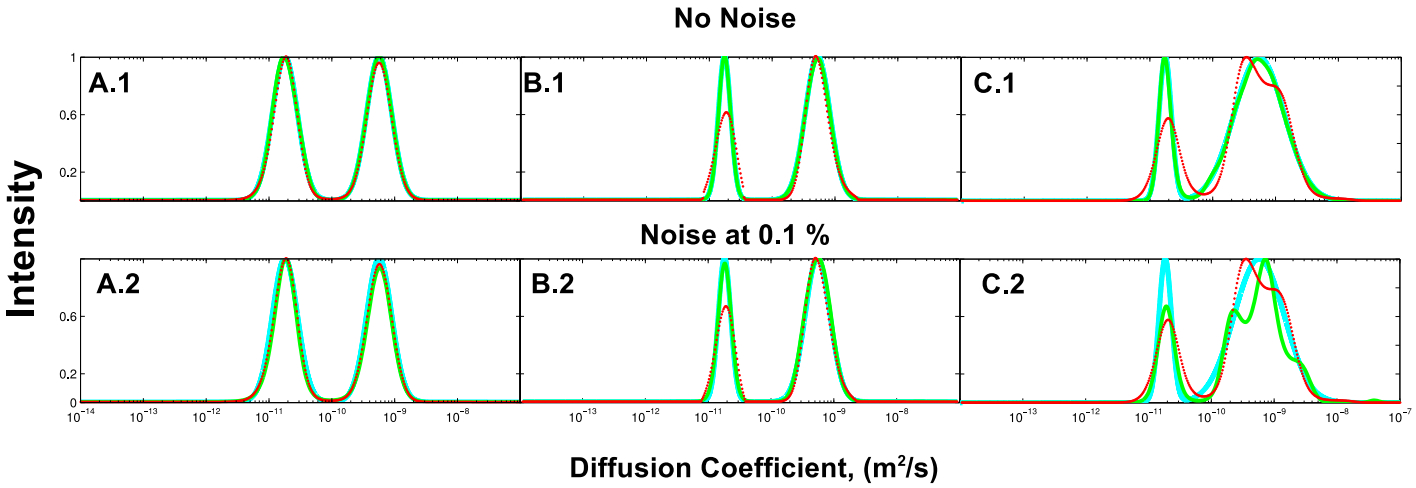
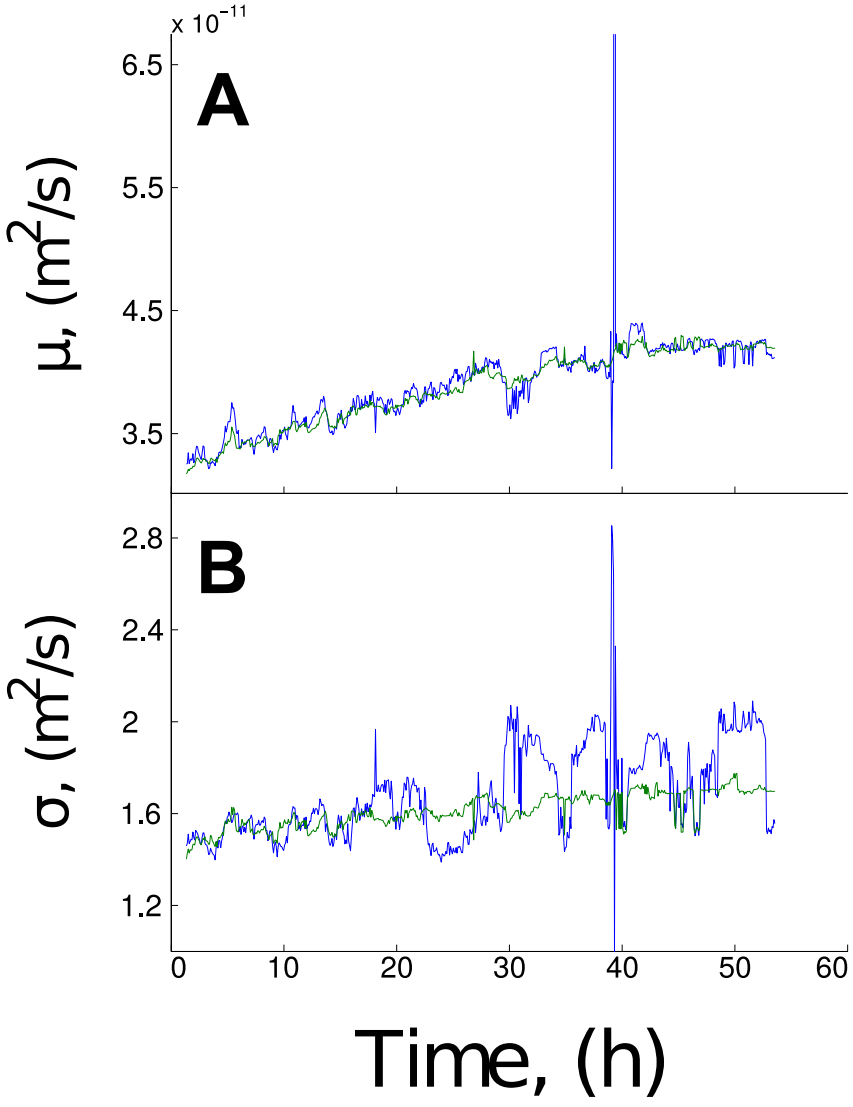


Fig. SI.5 Comparison of results of heparin degradation monitoring using Tailored Norm (green) and TRAIn (blue)



5 Smoothing features of ℓ_p-norms ($p > 1$)

The proposed method is using the regularisation term defined as

$$\Theta(A) = \|A\|_{\ell_p}, \tag{SI.4}$$

where $1 \leq p \leq 2$ and the algorithm seeks for the following minimum

$$\min_{A \geq 0} ||\Phi A - \Psi||^2_{\ell_2} + \tau ||A||_{\ell_p}. \tag{SI.5}$$

The solution might be any discrete function A, i.e. a vector in $V = \mathbb{R}^N$ space, where the dimension N is equal to the cardinality of the support of A. We note that $Res(A) = ||\Phi A - \Psi||^2_{\ell_2}$ is a real-valued function of $\mathbb{R}^N$. Then, we assume that $M_c \subset V$ is the surface given by $Res(A) - c = 0$ for any arbitrary $c \in \mathbb{R}$. If we restrict our consideration of minimizing function SI.5 only to the surface M_c, it is clear that our solution provides the minimal ℓ_p-norm among all points in M_c. Therefore the optimisation of p is straightforward and will further be illustrated with the following example. To start with, we consider the following vector

$$v = (1, \varepsilon), \tag{SI.6}$$

where ε is very small positive number. Now, consider the perturbations of this vector by $\delta > 0$ and the associated change of its norm.

$$||v_1||^2_{\ell_2} = ||(1-\delta, \varepsilon)||^2_{\ell_2} = 1 - 2\delta + \delta^2 + \varepsilon^2 \tag{SI.7}$$

$$||v_2||^2_{\ell_2} = ||(1, \varepsilon - \delta)||^2_{\ell_2} = 1 + \delta^2 - 2\varepsilon\delta + \varepsilon^2. \tag{SI.8}$$

Accordingly, the norm in equation SI.7 is less then the norm in SI.8. Hence, if the two vectors v_1, v_2 have a similar residuum, the algorithm will prefer vector v_1. In other words, the solution will be over-smoothed, i.e. the values of the elements of the solution vector v_1 differ less compared to v. In contrast, the ℓ_1-norm regularisation avoids over-smoothed solutions. Moreover, the ℓ_1 space is not strictly convex and in many cases, its geometry enforces sparse solutions. Between the ℓ_1-norm and ℓ_2-norm, there is a whole continuum of norms. A similar approach shows that larger p values produce more smoothed solutions.

6 ^{1}H spectra of PEO polymers and heparin

Fig. SI.6 ^{1}H spectrum of PEO polymers. Highlighted is the peak from polymer.

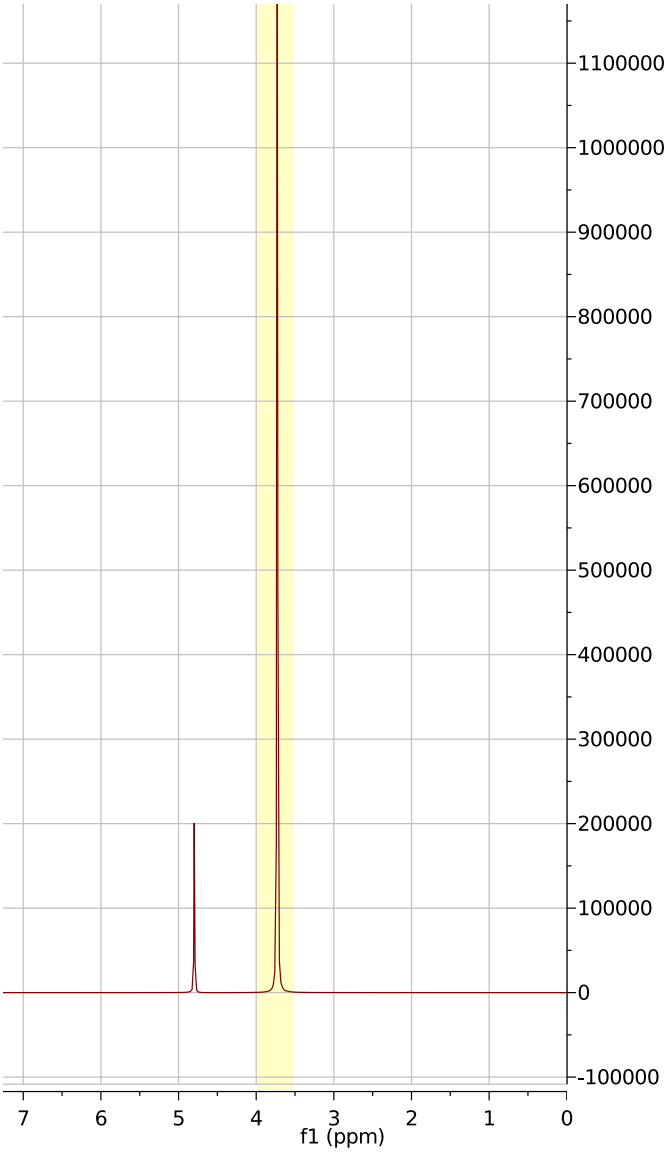


Fig. SI.7 ^{1}H spectra of Heparin at the begining (lower) and at the end (upper) of the reaction. Highlighted are the regions used for the analysis of reaction. The spectra are extracted from the moving-frame dataset and correspond to different gradients, which explains difference in S/N ratio.

References

1 E. J. Candes and M. B. Wakin, *IEEE Signal Process. Mag.*, 2008, **25**, 21–30.

2 I. Daubechies, R. Devore, M. Fornasier and S. Güntürk, *Commun. Pure Appl. Math.*, 2010, **LXIII**, 1–38.

3 K. Kazimierczuk and V. Y. Orekhov, *Angew Chem Int Ed Engl*, 2011, **50**, 5556–5559.

4 K. Xu and S. Zhang, *Anal. Chem.*, 2014, **86**, 592–599.