J Biomol NMR (2017) 67:77–94
DOI 10.1007/s10858-016-0085-2

ARTICLE

INFOS: spectrum fitting software for NMR analysis

Albert A. Smith[1]

Received: 13 September 2016 / Accepted: 23 December 2016 / Published online: 3 February 2017

Abstract Software for fitting of NMR spectra in MATLAB is presented. Spectra are fitted in the frequency domain, using Fourier transformed lineshapes, which are derived using the experimental acquisition and processing parameters. This yields more accurate fits compared to common fitting methods that use Lorentzian or Gaussian functions. Furthermore, a very time-efficient algorithm for calculating and fitting spectra has been developed. The software also performs initial peak picking, followed by subsequent fitting and refinement of the peak list, by iteratively adding and removing peaks to improve the overall fit. Estimation of error on fitting parameters is performed using a Monte-Carlo approach. Many fitting options allow the software to be flexible enough for a wide array of applications, while still being straightforward to set up with minimal user input.

Keywords Spectrum fitting · Quantitative NMR · Data analysis · Multi-dimensional NMR

Introduction

Nuclear magnetic resonance (NMR) is a powerful method for obtaining atomic-level structure and dynamics of molecules. Using NMR spectra to determine molecular information is a particularly challenging problem for large molecules, addressed by many programs to aid in the analysis of spectral information. These include programs designed for assignment of resonances such as CARA, CCPN, NMRView, MCAssign2, and Sparky (Goddard and Kneller ; Hu et al. 2011; Johnson and Blevins 1994; Keller 2004; Skinner et al. 2016; Vranken et al. 2005), determination of dihedral angles such as TALOS and DANGLE (Cheung et al. 2010; Cornilescu et al. 1999), atomic structure calculation such as ARIA2, CYANA, UNIO, and NIH-XPLOR (López-Méndez and Güntert 2006; Rieping et al. 2007; Schwieters et al. 2003; Serrano et al. 2012), and dynamics analysis such as relax and ModelFree (Mandel et al. 1995; Morin et al. 2014; Palmer et al. 1991), to name only a few of those most commonly used.

Such programs may take into account peak positions, to determine information from local electronic structure (chemical shift) or to determine which resonances are correlated, indicating bonding or spatial proximity. In this case, a number of algorithms exist to identify resonance positions (peak picking) that go beyond simple searches for local extrema (Alipanahi et al. 2009; Cheng et al. 2014; Corne and Johnson 1992; Garrett et al. 1991; Koradi et al. 1998). However, one may want to utilize fits to specific lineshapes to improve peak identification, or one requires additional information about resonances, such as accurate linewidth, amplitude, or integral values. For example, relaxation measurements for dynamics analysis or quantitatives measurement of couplings relies on series of spectra, for which amplitudes of peaks in those spectra are fitted to recoupling curves (for example, the REDOR or RFDR experiments (Gullion and Schaefer 1989; Sodickson et al. 1993)).

In this case, it is often necessary to go beyond peak-picking and simple amplitude or integral measurement and

Electronic supplementary material The online version of this article (doi:10.1007/s10858-016-0085-2) contains supplementary material, which is available to authorized users.

✉ Albert A. Smith
alsi@nmr.phys.chem.ethz.ch

[1] Physical Chemistry, ETH Zürich, Vladimir-Prelog-Weg 2, 8093 Zurich, Switzerland

fit experimental NMR data to calculated spectra, in particular when resonances are not well resolved. Fits of such data can be performed in the time domain or the frequency domain. In the frequency domain, spectral resolution allows fits of different spectral regions to be affected by different fit parameters, making optimization simpler compared to the time domain, where nearly every data point is strongly affected by every fit parameter. On the other hand, apodization and truncation of data before Fourier transform leads to complicated shapes in the frequency domain, which often are modeled with simpler Gaussian or Lorentzian approximations of those lineshapes (van den Boogart et al. 1994). A number of programs exist to fit NMR data, with some of the more general frequency-domain solutions found in NMRPipe (Delaglio et al. 1995) and dmfit (Massiot et al. 2002). Time-domain fitting programs include those by (de Beer and van Ormondt 1992; van Dijk et al. 1992; Van Huffel et al. 1994).

In the interest of being able to fit spectra with many resonances while keeping the optimization problem simple, the frequency domain approach is used here. Figure 1 illustrates the problem of fitting frequency domain lineshapes, which result from Fourier transformation of truncated and apodized time-domain data, to Gaussian and Lorentzian functions. One can see that neither Gaussian nor Lorentzian functions matches the signal well, particularly near the baseline. This problem can be resolved by using lineshapes calculated via processing and Fourier transform of simulated signals (Chylla et al. 1998; Slotboom et al. 1998), an option in NMRPipe (Delaglio et al. 1995). However, performing Fourier transforms at every step in the fit optimization is computationally expensive, particularly for large, multi-dimensional spectra.

A second challenge is how practical usage of spectrum fitting software is. In a typical spectrum with significant peak overlap, an initial peak-picking does not always identify all peaks. Therefore, achieving a good spectrum fit often requires user intervention, to observe what regions are poorly fit, and make judgments as to where additional peaks should be added to make improvements. A spectrum-fitting algorithm can perform this step to reduce the necessary user input, a method that has previously been used to refine NMR data fitted in the time-domain, by periodically adding resonances and verifying if the overall fit improves (Chylla and Markley 1994). Finally, once a spectrum fit has been obtained, it is useful to have an estimate of the error

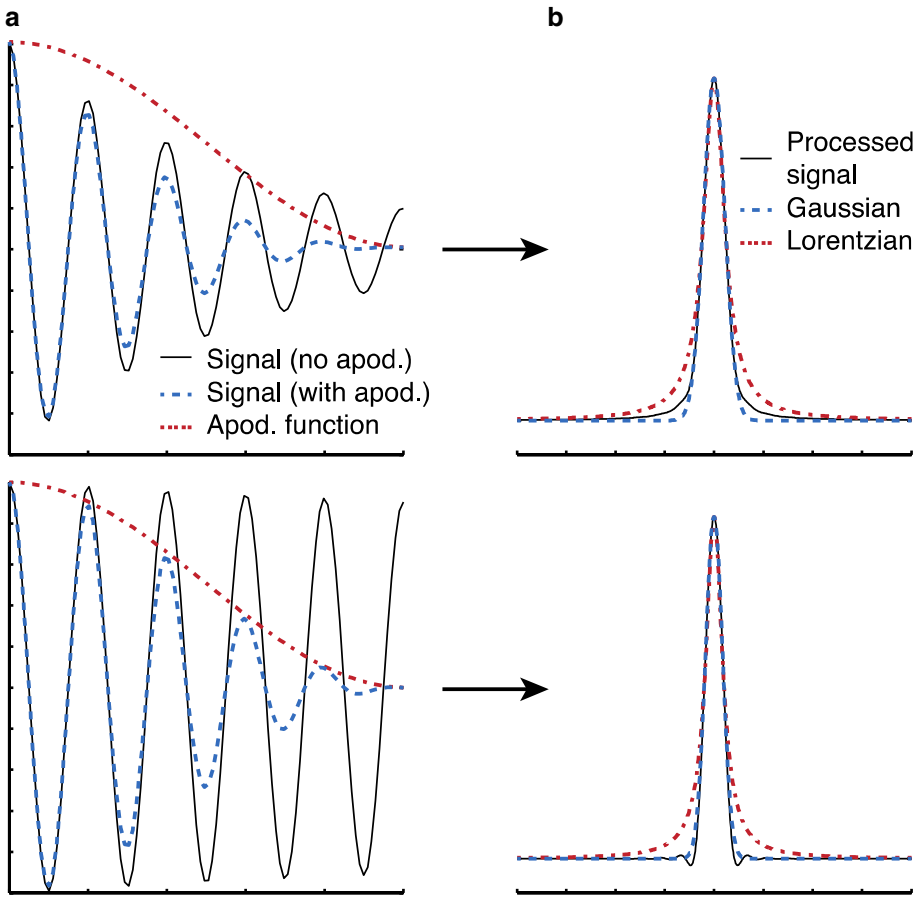

Fig. 1 Comparison of Gaussian and Lorentzian lineshapes to processed signals. Two signals are shown in **a**, one with fast and one with slow signal decay (*black*), which are then apodized with a squared-sine function (apodization function is *red*, apodized signal is *blue*). The Fourier transformed signals are given in **b** (*black*). These signals are compared to a Gaussian (*blue*) and Lorentzian (*red*) signal with the same position, amplitude, and full width at half maximum

on the fitting parameters, a problem that may be addressed using Monte-Carlo methods (Metropolis and Ulam 1949).

The INtelligent Fitting Of Spectra (INFOS) software has been developed to combine the advantages of frequency and time domain fitting, while also minimizing user input. INFOS runs in the MATLAB environment (Mathworks 2013b), providing a convenient interface and allowing easy integration with other MATLAB programs for further analysis. Programs provided are open source software (general public license), and are distributed at infos.sourceforge.net. INFOS attempts to satisfy several requirements, that it is accurate, simple, flexible, and fast:

1. Accurate: The shape of each peak simulated is determined using acquisition and processing settings so that simulated lineshapes are well-fitted to experimental shapes. Linewidth is allowed to vary by changing the rate of signal decay in the time domain, rather than simply scaling the peak width. This results in lower fitting error- particularly near the edges of peaks, reducing the influence that peaks have on the fits of their neighbors.
2. Simple: INFOS can be run fully automatically with all fitting settings determined by the software. Furthermore, INFOS iteratively determines an optimized peak list, if the user does not provide one.
3. Flexible: The user is given full (but optional) control of all fitting settings in order to make improvements to the spectrum fit. INFOS also gives the user the ability to restrict the fitting parameters (position, linewidth, amplitude).
4. Fast: Methods are implemented to accelerate the calculation of spectra, including performing fits of spectra in sections (sub-spectra) to reduce the complexity of fits, gradient-based optimization for efficient fitting, and use of parallel processing where possible.

The result of these requirements is that the user may easily start using INFOS, but then refine its behavior in order to solve their particular type of problem, without using an excessive amount of computational time. A summary of the capabilities of INFOS is given here:

Fit n-dimensional spectra.
Calculate lineshapes using acquisition and processing parameters.
Determine peak list while fitting –or– Fit spectra with input peak list.
Characterize spectrum noise.
Determine optimal fitting settings –or– Allow user-optimized settings.
Restrict/fix fitting parameters (peak position, linewidth, amplitude).
Analyze error.
Fit user-defined functions to series of spectra.
Parallel processing.
Spectrum editing (truncating, slicing, adding, projecting).
Display 2D and 3D spectra.
Interactive 2D fitting.

The full list of options is quite extensive, so that users should refer to the user manual, found in the supplementary information and on the distribution website (http://infos.sourceforge.net), although it is straightforward to get started with INFOS without knowing all options.

Methodology

The tasks performed by the INFOS software can be divided into several major categories, which are discussed here. These are spectrum calculation, spectrum fitting with a peak list, peak list determination, and spectrum noise-analysis and determination of fitting settings. The procedure for fitting a spectrum without any initial information is outlined in Fig. 2. Additionally, methods for analysis of fitting error are discussed, as are methods for fitting a series of spectra to user-defined functions.

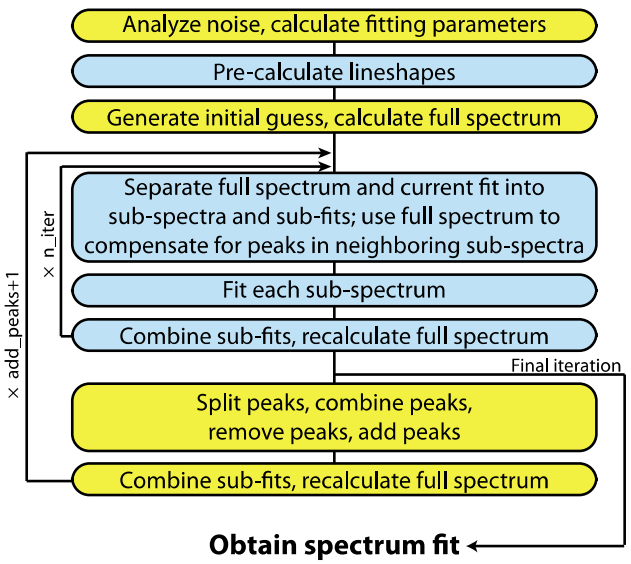

Fig. 2 Flow chart for spectrum fitting. The full procedure for calculating a spectrum without any initial input is shown. Steps that are only necessary if the peak list needs to be determined (no initial peak list) are highlighted in yellow, whereas steps that also used when fitting from a peak list are highlighted in blue

Spectrum calculation

Fast and accurate spectrum calculation is a critical component of the INFOS software. In order to obtain accurate lineshapes one must perform the same processing steps used for the experimental spectrum to a calculated free-induction decay (FID); therefore the performance of INFOS depends critically on having acquisition and processing parameters as input. The steps to generating a lineshape are (Ernst et al. 1987; Hoch and Stern 1996):

Generate FID → Zero-fill FID → Apodize FID
→ Apply Fourier Transform.

For a multi-dimensional spectrum, however, this is an expensive computation, and would be prohibitively slow to use for optimization of a spectrum fit. Fortunately, there are several properties of this operation that allow it to be accelerated. First, all steps in the processing are linear operations, and therefore the initial FID may contain just a single peak. The final spectrum is then just a simple sum of all the spectra obtained from processing of individual peaks. Also, the shape of a peak is not affected by its frequency- so that it is possible to generate different frequencies simply by shifting the location of the processed peak in the spectrum, allowing recycling of the processed peaks (of course, the linewidths will affect the shape). Finally, a multi-dimensional peak can be obtained from the Kronecker product of the one-dimensional peaks– and therefore one processes the one-dimensional FIDs and only as a final step, takes the Kronecker product of the one-dimensional peaks to obtain the multi-dimensional peak.

The following procedure can then be used for calculation of a spectrum, giving fast calculation but accurate lineshapes (illustrated in Fig. 3): At the beginning of spectrum fitting, for each dimension INFOS generates a series of FIDs, spanning a range of decay rates (discussed further below). Each FID is on resonance, and so does not oscillate. The FIDs are processed, generating one-dimensional lineshapes that are exactly in the center of the spectrum. Then, these shapes are stored for each dimension, creating a 'catalog' of lineshapes (shapes with a range of linewidths) to be later used in spectrum calculation during the fitting procedure. Then during spectrum calculation, for each peak lineshapes with the correct linewidth are selected out of the catalog in each dimension (therefore the possible linewidths take on only a discreet set of values in the fitting- although enough linewidths are stored in the catalog that it is unlikely one can distinguish error resulting from the discretization). The shape for each dimension is shifted so that it appears at the correct position. Once the peak is calculated for all dimensions, the Kronecker product is taken to obtain the peak in the full spectrum, and the peak is scaled to the correct amplitude. This is simply repeated for all peaks and then added together. This is given for a spectrum with N peaks by

$$\mathbf{I}_{full} = \sum_{n=1}^{N} A_n \vec{I}_n^{(1)} \otimes \vec{I}_n^{(2)} \otimes \cdots \otimes \vec{I}_n^{(m)} \quad (1)$$

where $\otimes$ is the Kronecker product, which expands the dimensionality, the $\vec{I}_n^{(k)}$ are the one-dimensional lineshapes up to m dimensions (functions of position, $\delta_n^{(k)}$ and linewidth, $\Delta_n^{(k)}$), A_n are the peak amplitudes, and $\mathbf{I}_{Full}$ is the resulting spectrum.

The generation of the FID itself is also an important component of this process. The user may specify how the FID should decay in each dimension. The options are for the FID to have Gaussian decay ('gauss'), exponential decay ('lorentz'), or some fractional mix of the two types

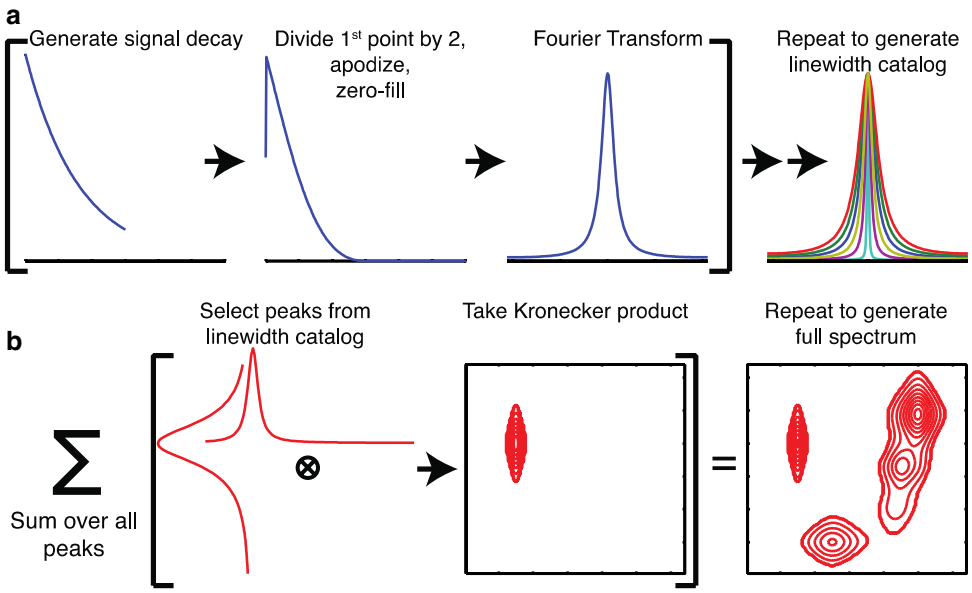

Fig. 3 Spectrum calculation method. INFOS calculates spectra by pre-generating a lineshape catalog for each dimension, which is produced by inputting a series of signal decays with different decay rates and processing them to generate accurate lineshapes, which is illustrated in **a**. A full spectrum is calculated by selecting peaks from the linewidth catalog in each dimension, shifting them to the correct resonance frequency, taking the Kronecker product, and scaling to the correct amplitude. This is repeated for each peak and the result is summed together, as illustrated in **b**

of decay ('mixXX'). One also can add a fixed amount of Lorentzian or Gaussian broadening to all peaks in a given dimension. Note that the peak linewidths are parameterized only by a single variable in each dimension, so all lineshapes in a given dimension will be calculated using the same FID decay type.

Spectrum fitting with a peak list

The 'FitSpec' function in INFOS can fit spectra either from an initial peak list for which the total number of peaks is fixed (Fig. 2, blue sections), or can generate a peak list for which peaks are iteratively added and removed in order to fully optimize the fit (Fig. 2, all sections). Here, the former case is discussed. In order to simplify the fitting procedure, the full spectrum is first broken into sections (sub-spectra) by applying a grid to the spectrum. Each sub-spectrum is fitted separately, greatly reducing the number of variables in each fit, and reducing the size of the spectrum to be calculated with each fit iteration. The main problem with this approach is that when fitting a sub-spectrum, peaks in neighboring sections will overlap into it, distorting the fit. Therefore, before fitting of the individual sub-spectra begins, and initial calculation of the full spectrum is performed, from which peak overlap between sub-spectra is calculated. For a given sub-spectrum, any overlapping peaks that are outside of that sub-spectrum are then subtracted away before fitting begins. After all sub-spectra have been fitted, the full spectrum calculation is updated, the overlap is re-calculated, and the fitting process is repeated. This is done for several iterations (four, by default) in order to refine the fit, and minimize influence of peaks from neighboring sub-spectra. Note that peaks that are centered near the edge of a sub-spectrum tend to not be fitted as well, because much of their intensity will be in a separate sub-spectrum. In order to remedy this, the size of the sub-spectra is changed between the fitting iterations so that different peaks are on the edge of sub-spectra for the different fitting iterations.

In order to fit each sub-spectrum, INFOS uses a gradient-based fitting routine – the Levenberg–Marquardt algorithm with a trust region (Levenberg 1944; Marquardt 1963; Sorenson 1982) as implemented in the MATLAB Optimization Toolbox in the 'lsqcurvefit' function (Mathworks 2013a). This algorithm takes a sub-spectrum and attempts to minimize the difference between the calculated and the experimental spectrum. In order to do so optimally, a functional form of the Jacobian matrix for the calculated spectrum should be provided, where the Jacobian gives the derivative of each point in the spectrum with respect to each of the fitting parameters. For a given peak, n, in dimension k, the derivative of the spectrum with respect to the position ($\delta_n^{(k)}$) and the linewidth ($\Delta_n^{(k)}$) is given as the Kronecker product of the lineshapes in all dimensions other than k with the one-dimensional derivative of the peak shape in dimension k. The derivative with respect to the amplitude of peak n is simply the product of the lineshapes in all dimensions.

$$\begin{aligned}
\frac{dI_{full}}{d\delta_n^{(k)}} &= A_n I_n^{(1)} \otimes \cdots \otimes \frac{dI_n^{(k)}}{d\delta_n^{(k)}} \otimes \cdots \otimes I_n^{(m)} \\
\frac{dI_{full}}{d\Delta_n^{(k)}} &= A_n I_n^{(1)} \otimes \cdots \otimes \frac{dI_n^{(k)}}{d\Delta_n^{(k)}} \otimes \cdots \otimes I_n^{(m)} \\
\frac{dI_{full}}{dA_n} &= I_n^{(1)} \otimes \cdots \otimes I_n^{(m)}
\end{aligned} \tag{2}$$

The one-dimensional derivatives found in (2) are calculated at the beginning of the fitting routine, as is done with the peak shape catalog, and are then stored for later use with optimization. No pre-calculated derivative is necessary for the Jacobian with respect to the peak amplitude, since this only depends on the lineshapes, which are already stored in the catalog.

Peak list determination

INFOS does not need to be given a peak list to fit a spectrum – the 'FitSpec' function may also generate the peak list itself. The goal of peak list determination is to correctly place enough peaks in the spectrum to maximize information content and accuracy of the resulting fit, while minimizing the amount of noise that is incorporated into the fit parameters. One therefore wants to add peaks where the difference between the experimental and calculated spectra (the residual) is high, but at the same time remove peaks where the residual is not significantly improved by the peak. This is achieved in INFOS with an iterative process of peak list modification: a peak list is generated using a simple search for local extrema, a fit is performed as described in sect "Spectrum fitting with a peak list", and the peak list is then modified to improve the fit, with several iterations performed. Four methods of peak list modification are performed: 'add peaks', 'remove peaks', 'split peaks', and 'combine peaks'. The first three methods depend on a cutoff – a parameter that specifies that if the fit residual is below a certain height it should be assumed to be noise (this parameter is also used for the initial peak list generation). The combine peaks method depends on another parameter, noise-per-peak, which specifies approximately how much noise a peak can fit. The four methods are detailed here, and the method of determining the 'cutoff' and 'noise-per-peak' parameters are discussed later in sect "Noise analysis and determination of fitting settings".

The 'add peaks' method (Fig. 4a) simply looks for peaks in the residual spectrum (the difference between the

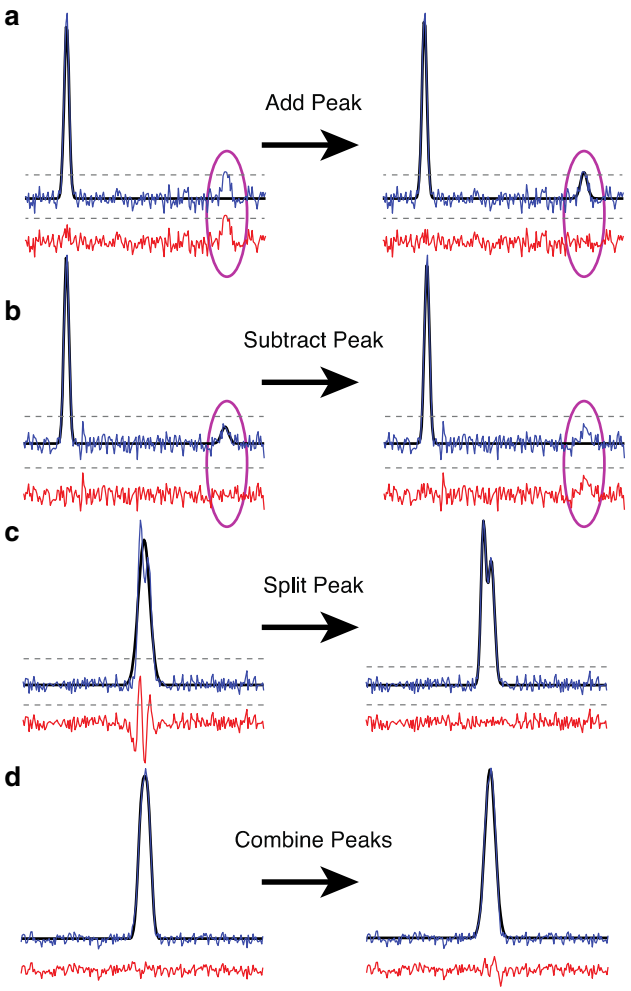

Fig. 4 Methods of peak list modification. Each plot shows an experimental spectrum (*blue*), a fitted spectrum (*black*), and a fit residual (*red*), before and after execution of one of the peak list modification methods. Dotted lines show the cutoff level for the experimental and residual spectra. **a** Shows addition of a new peak where the fit residual is above the cutoff. **b** Shows elimination of a peak where the resulting residual is less than the cutoff. **c** Shows splitting of a peak where two maxima are found in the fit residual on either side of the peak (a resolved peak pair is shown to better illustrate this). **d** Shows combination of two peaks, resulting in slightly higher error in the residual, but which is not high enough to justify fitting the region with two peaks

experimental spectrum and the calculated spectrum), which have amplitudes higher than the cutoff, and adds peaks at these points. The 'remove peaks' method (Fig. 4b) is similar to 'add peaks'; it looks for peaks included in the fit that when removed, do not cause the spectrum residual to become higher than the cutoff.

The 'split peaks' method (Fig. 4c) finds single peaks in the fit that are likely to be two peaks in the experimental spectrum. This situation often arises if two nearby peaks (usually of similar amplitude) in the experimental spectrum are unresolved. Then, the initial peak pick identifies them as only one peak. However, INFOS will not be able to fit the lineshape correctly, since the shape of the two nearby, unresolved peaks is typically different than the shape of a single peak (depending on how nearby the peaks are). Two peaks being fitted by a single peak usually causes the residual spectrum to also have two peaks, one on each side of the fit peak. Therefore, INFOS searches around each fitted peak for peaks in the residual that are above the cutoff value, and that fall within the linewidth of the original peak. If two or more peaks are found in this region, then the original peak is removed and new peaks are placed at the position of the peaks originally found in the residual (allowing for peaks to be split into two or more new peaks). Note that the 'split peaks' method is always run before the 'add peaks' method, since multiple errors in the residual around existing peaks would always be corrected by the 'add peaks' method. However, 'add peaks' places two (or more) new peaks in addition to the existing peak, whereas 'split peaks' eliminates the original peak so that only the two (or more) new peaks are being used to fit the region. This avoids over-fitting such regions, and should usually result in a better fit.

The final method, 'combine peaks' (Fig. 4d) attempts to eliminate regions in the fitted spectrum that are being over-fit. Although the other methods try to avoid placing too many peaks in a given region in the spectrum, occasionally the fit optimization (sect "Spectrum fitting with a peak list") will move two peaks towards each other so that they partially overlap and in fact a single peak could fit the region in the experimental spectrum without significantly increasing the error. This method works by first finding pairs of peaks that are separated by less than half the sum of their linewidths. For such pairs, the program then determines if the sum of the peaks yields only one resolved maximum in the calculated spectrum. If two maxima are found, then the peaks will not be combined. Otherwise, INFOS tries to replace the two peaks with a single peak. The fit residual is determined for fits both with two peaks and one peak (strictly speaking, the sum of squares of all peak maxima in the fit residual are determined). The fit with two peaks is almost always better, however, INFOS requires that it is better at least by the 'noise-per-peak' parameter; if it is better by this amount then the two peaks are kept, otherwise they are replaced by a single peak. Note that the 'remove peaks' method cannot play the role of removing over-fitting because often the two peaks are of similar amplitude, and so neither peak on its own fits the criteria for removal, but the two peaks together may fit the criteria for combination. It is possible during the fit editing that the split peaks method determines that a peak should be split but the combine peaks method then recombines the same peaks, because the two methods depend on different

criteria for splitting/combining, although this is not usually detrimental to fitting.

These methods are used in-between steps of gradient-based fit optimization, in an iterative manner. Between steps of fit optimization, the methods always run in the following order: split peaks, combine peaks, remove peaks, add peaks. Note that before the last step of fit refinement, only the combine peaks and remove peaks methods are run, to avoids adding peaks before the last refinement step that may not help improve the fit. Additionally, the 'cutoff' parameter is set higher for the initial peak list determination, and lowered for the first two iterations of fit optimization and peak list modification. This allows large peaks to be fit first, so that noise that is elevated by nearby peaks is not mistakenly picked as a peak.

Noise analysis and determination of fitting settings

INFOS is able to fully determine its own fitting settings (although it is possible for the user to override these settings, to help optimize the fit). Particularly critical to optimum spectrum fitting are determination of the cutoff (the amplitude below which a peak is assumed to be noise), and the noise-per-peak parameter (determines when two peaks over-fit a region and should be combined, see sect "Peak list determination"). Determining these requires a good analysis of the distribution of noise in the spectrum. The most straightforward method to do so is to select a large, empty region of a spectrum and calculate the RMS (root mean squared deviation from zero). However, such regions are not always available, or easy to identify. An alternative method is to take a histogram of all amplitudes in a spectrum, and fit the histogram to the expected noise distribution (typically Gaussian), a method used in magnetic resonance imaging, and in digital signal processing in general (Brummer et al. 1993; Caglioti and Maniezzo 1995; Sijbers et al. 2007; Smith 1999).

INFOS uses a similar method to estimate the noise level. However, because peaks are added only at local extrema, INFOS only includes peaks from the spectrum in the histogram, in order to eventually obtain a probability distribution of noise-peak heights. In order to separate signal from noise, INFOS first calculates a noise spectrum, by processing pseudo-random white noise with the same parameters that the spectrum was processed with (see Fig. 5a). Then, rather than taking a Gaussian probability distribution, the histogram of the calculated noise spectrum is taken and fit to the experimental histogram. The variables of this fit are the RMS of the noise and the total number of peaks in the noise (the latter is necessary because signal peaks cover some noise, reducing the number of noise-peaks in the experimental spectrum). An example fit of the noise-peak histogram is shown in Fig. 5b. The fitted distribution (black line) can be used to calculate the spectrum RMS, and furthermore can be used to compute the probability of a peak of given height being noise.

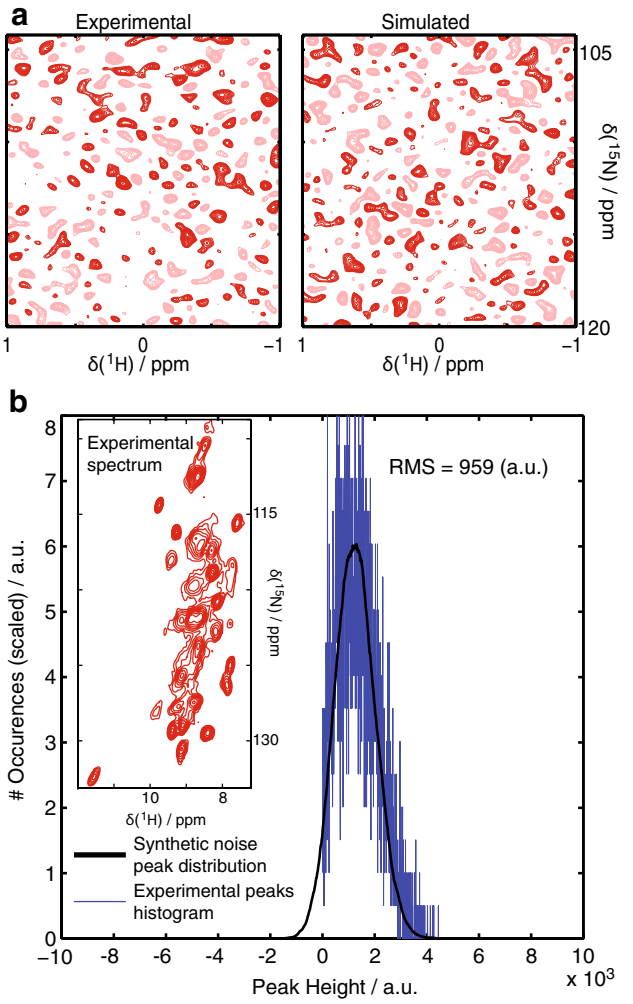

Fig. 5 Fitting experimental noise to a calculated distribution. **a** Shows plots of experimental noise (*left*) and simulated noise (*right*), with the experimental noise taken from an empty region in the spectrum. One can see that the simulated noise has the same characteristics as the experimental noise. **b** Shows a histogram of experimental peaks fitted to a synthetic noise distribution. A histogram of all concave-up (downward pointing) peaks with negative amplitude (*blue*) is fitted to a distribution of peak intensities of synthetic noise (*black*). The inset shows the experimental spectrum being analyzed. The RMS determined using this method is 959 (arbitrary units). The RMS determined from a large, empty region in the spectrum is 956

Use of a histogram reduces the impact that signal peaks have on the noise estimation. INFOS further only includes negative, concave up peaks in the experimental histogram to further improve accuracy (unless the fit includes negative peaks, in which case all peaks are used). Using this method in the example in Fig. 5 yielded an RMS of 959 (arbitrary units), compared to an RMS of 956 determined by analyzing a large, empty spectrum region. Fitting a Gaussian to a

histogram of all points in the spectrum is notably less accurate, yielding an RMS of 1079 (although setting a threshold for amplitudes to include in the histogram could improve this).

Note that the accuracy of this method will be reduced when the amount of empty space in the spectrum is reduced. It also fails if acquisition and processing parameters are not supplied to INFOS or if they are incorrect, and the method is sensitive to baseline distortion (in which case the user will need to supply the cutoff and noise-per-peak settings). Once a distribution of noise is determined, the cutoff is then calculated so that an approximate number of noise peaks will be fitted in the final analysis. By default, INFOS sets the cutoff so that ~1 % of fitted peaks are noise, but this setting may be changed in a variety of ways (see INFOS manual).

INFOS must also determine the 'noise-per-peak' parameter, which determines how much the fitting residual should be reduced by a peak when running the combine peaks method. The concepts used here are related to model selection using the reduced-χ^2 statistic, given by

$$\chi_\nu^2 = \frac{\chi^2}{\nu} = \frac{1}{\nu}\sum_{i=1}^{n}\frac{(O_i - C_i)^2}{\sigma_i^2} \quad (3)$$

Here, ν is the number of degrees of freedom, n is the number of observations, O_i is an observation, and C_i is a value calculated from a model with m fit variables. For linear models, with uncorrelated noise, one can equate degrees of freedom with number of observations minus number of fit variables ($\nu = n - m$). Then, addition of a parameter to the model that only fits noise is expected to reduce error ($(O_i - C_i)^2$), on average, by the noise variance (σ_i^2), therefore decreasing χ^2 by 1, and leaving the reduced-χ^2 roughly unchanged. A parameter that improves the model, as opposed to just fitting noise, leads to reduction in the reduced-χ^2 (Hughes and Hase 2010). However, data points in NMR spectra are not always independent, making the effective number of observables difficult to determine, and furthermore the model is not linear, leading to an effective number of degrees of freedom which is not necessarily given by $m - n$ (Buja et al. 1989).

Rather than determining the effective number of degrees of freedom, INFOS simply attempts to estimate how much the fit residual should be reduced when a peak fits only noise (the 'noise-per-peak'). Then, inclusion of a peak in the model that results in an improved model should reduce the total residual more than this amount. Therefore, the noise-per-peak is determined by taking a synthetic noise spectrum (generated as described above, using the same RMS as determined for the experimental spectrum) and attempting to fit 75 noise peaks in the noise spectrum. The sum of squared peak heights is determined before and after the fit, and the change in this value, divided by 75, is the noise-per-peak. Strongly overlapped peaks typically are much less efficient at reducing the fit residual (whether they are fitting actual peaks or noise), and so if inclusion of a peak in the fit cannot at least reduce the residual as much as a peak fitting only noise, then it is removed.

A number of additional settings control the INFOS optimization. The number of iterations taken during the gradient-based minimization, the number of iterations of fitting individual sub-spectra and reconstructing the full fit and full spectrum, and the number of iterations to edit the peak list are all set by INFOS—however, they do not depend on spectrum signal-to-noise, and so are simply fixed values which give good, but quick fitting of spectra. The size of the grid for fitting sub-spectra is also optimized- to give approximately 3 peaks per sub-spectra, but also is set to give less than 20,000 data points in a sub-spectrum (adjustment of the grid is sometimes necessary – more sub-spectra give faster fitting, but lower quality fits, especially if there are broad peaks).

Functional fitting

INFOS can fit a series of spectra for which amplitudes in the series can be fitted to some user-defined function, using the 'FitTrace' function (similar capabilities are found in the NMRPipe program (Delaglio et al. 1995)). The user-defined function must depend on some variable (called 'x' by INFOS), and is provided numerically for each possible value of the variable 'x' (and also for each spectrum in the series). Typically, this is something like a series of spectra for which exponential decay occurs and the relaxation rate corresponds to 'x', or some recoupling occurs and the size of the recoupled interaction corresponds to 'x'. The 'FitTrace' function takes such a series and performs an initial fit of the individual spectra. From this initial fit, amplitudes are extracted in a series for each peak in the spectra. These series of amplitudes are then fitted initially to the user-defined function (determining an estimate of the peak amplitude and 'x' for the complete series). INFOS then constructs a new spectrum for which the last dimension is now the user-defined function. The complete series of spectra is fitted at the same time, with the last dimension being fitted to the user-defined function. Note that 'FitTrace' must be supplied with an initial peak list.

Error estimation

INFOS can estimate the error of fitting parameters, with the 'FitError' function, using a simple Monte-Carlo approach (Metropolis and Ulam 1949). The basic

methodology is relatively simple: INFOS uses the fit parameters determined for an experimental spectrum and calculates a 'noiseless' spectrum. Synthetic noise is added to this spectrum, with the same RMS as is determined from the experimental spectrum (noting that this is processed noise, as described in 1.4). Then, this fully synthetic spectrum is re-fit. The process is repeated a number of times (typically ~ 100 s of times), with a different set of noise added each time, and statistics are then performed on the resulting fit parameters. The result is essentially a simulation of experimental repetition. Note that recently, a similar 'bootstrapping' approach has been reported, which instead of using synthetic noise, uses noise sampled in sections from the fit residual (Waudby et al. 2016).

To accelerate this process, INFOS determines for each peak in a fit, what other peaks are sufficiently nearby to affect the fitting of that peak. Then, only a peak and its neighbors are refit in a truncated spectrum. This truncated spectrum is fit with only one sub-spectrum, eliminating the need to use iterative refitting in order to correct for neighboring sub-spectra (as is done with a full spectrum). This greatly accelerates the process of error analysis, since refitting a full spectrum hundreds of times can be very computationally expensive.

Note that the error reported by this method estimates how a fitted parameter would vary when an experiment is repeated many times. One should be very cautious, then, in the extent that this error analysis can be used to estimate the real error, i.e. can be interpreted as the confidence that a fitted parameter is within a certain range of the true value. For example, if a peak in the fit actually is fitted to two resonances in the spectrum (which have not been resolved with spectrum fitting), then the fitted peak can potentially be much further away from the two 'correct' peak positions than the error resulting from this analysis estimates. For the error reported here to be a good estimate of the true error, the spectrum must be well fit: in a given region, the true number of peaks matches the number of fitted resonances, and the lineshapes should be good matches between experiment and fit. It is also difficult to estimate error for peaks with intensities near the spectrum noise level, since the fit parameters describing them are underdetermined. Finally, the dominant source of error must be white noise in the time domain. Error due to experimental fluctuations (temperature, sample quality variation, etc.) cannot be accounted for using this method, nor can spectrum artifacts.

Parallel processing

INFOS performs some operations using parallel processing, using the 'parfor' function in MATLAB (Mathworks 2013b). Fitting of sub-spectra is performed in parallel, if multiple processer cores are available. However, peak list editing is not performed in parallel, and significant communication overhead reduces gains from parallel processing. Also, error analysis using the 'FitError' function is fully parallel, and the 'FitTrace' function uses parallel processing for the initial fitting of individual spectra in the series (and uses parallel processing for fitting of sub-spectra in the subsequent simultaneous fitting of spectra).

Supporting functions

INFOS provides a number of additional functions for manipulating spectra. Truncating spectra ('clip_spec_nD'), spectrum projections ('proj_nD'), slice extraction ('slice_nD'), and addition ('add_spec_nD') are all available functions. Although these are relatively straightforward operations, acquisition and processing parameters need to be edited to be consistent with the new spectra, and so it is necessary to use the provided programs for these operations. INFOS also provides the 'FitEditor2D' function, which allows interactive fitting of 2D spectra. Currently INFOS exports peak lists using the XEasy format (Bartels et al. 1995), and can import and export spectra in the Bruker Topspin format (Bruker Biospin 2016) or import spectra processed in NMRPipe (Delaglio et al. 1995). Note that CCPN can convert between a number of different peak list formats, including XEasy (Skinner et al. 2016; Vranken et al. 2005).

Examples and discussion

Basic fitting

The fit of a DARR spectrum (of HET-s fibrils (Siemer et al. 2006; Van Melckebeke et al. 2010; Wasmer et al. 2008)) is shown in Fig. 6. For this example, the experimental spectrum was the only information provided to INFOS, in addition to the acquisition and processing information. For a spectrum stored in the Bruker Topspin format, this can be achieved by the following:

```
spec=getSpecBruker(path_to_spectrum);

fit=FitSpec(spec);
```

Here, 'path_to_spectrum' is a string containing the location of the spectrum, and the variable 'spec' subsequently stores that spectrum, in addition to information on acquisition and processing of the spectrum. The variable 'fit' is a structure containing all the information from fitting (peak positions, amplitudes, linewidths, and integrals, as well as all settings). The resulting fit included 391 peaks, and was fitted

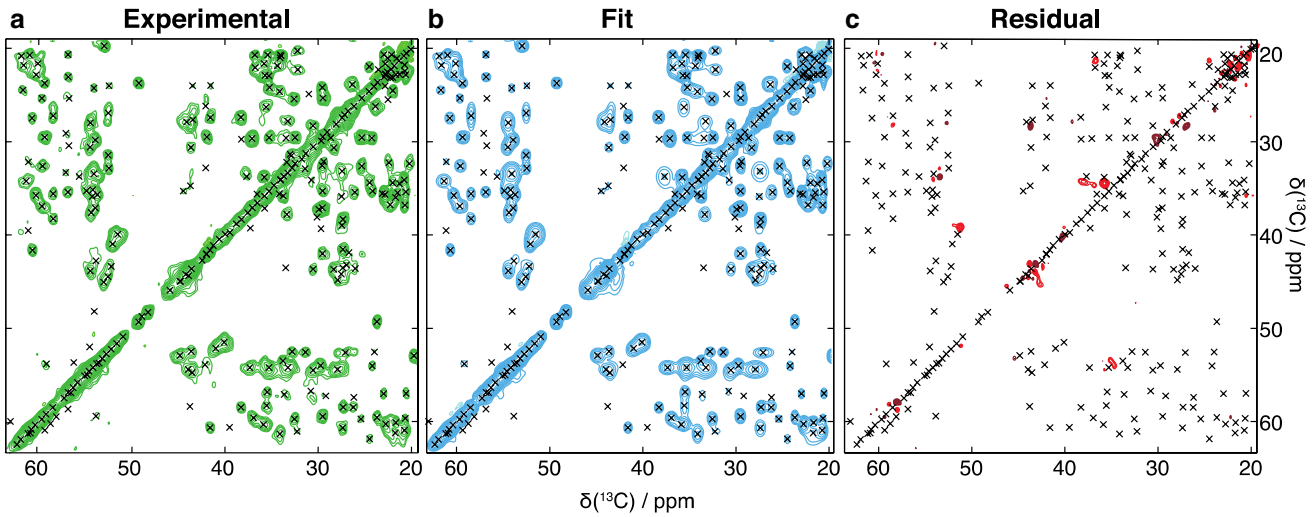

Fig. 6 Fit of a 50 ms C–C DARR spectrum (HET-s fibrils). **a** Shows the experimental spectrum **b** shows the calculated spectrum after fitting. **c** Shows the fitting residual (experimental minus calculated), where only a few poorly fit regions appear. The fit shown here was performed without any user instruction to the 'FitSpec' function, beyond supplying the spectrum, which contained acquisition and processing information. The lowest contour level is set to 1.2 % of the spectrum maximum, which corresponds to the cutoff value determined by 'FitSpec'

in 213 s (MacBook Pro Retina mid-2012, 8 processes). As one can see, a fairly accurate reproduction of the spectrum has been achieved, without any instruction supplied by the user.

The previous example represents a relatively straightforward case, and so to push the limits of the INFOS fitting routine, a 400 ms C–C DARR spectrum is tested. The region between 0 and 100 ppm was fit, yielding 4009 peaks (~11000s with 12 processes, Intel IvyBridge at 2.9 GHz). The resulting fit, shown in Fig. 7, is very good, with a reduced-χ^2 of 1.75. The complexity of the spectrum required some optimization of settings, so that the

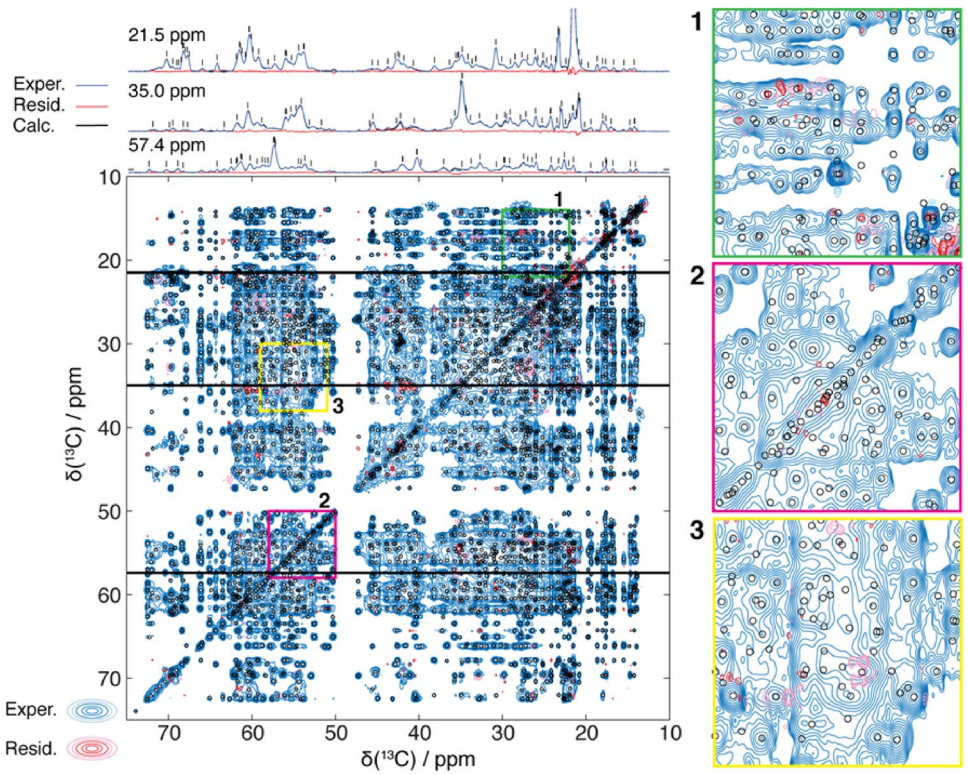

Fig. 7 Fit of a C–C DARR spectrum (Ubiquitin, 400 ms). The fit was performed between 0 and 100 ppm, with a truncated region shown. The experimental spectrum is shown in *blue/light blue* for positive/negative intensities. The fit residual is shown in *red/light red*. The fit was performed with a cutoff of 0.2 % of the spectrum maximum, and the contour minimum is at 0.15 % of the maximum. Three slices are extracted with experimental (*blue*), residual (*red*) and calculated (*black*) spectra shown. Dashes mark peak positions that are within +/− 0.3 ppm in the vertical dimension. Three regions are expanded to show fit quality and peak placement. Experimental spectrum is courtesy of Kathrin Szekely

cutoff was adjusted to 0.2 % of the spectrum maximum, the gridding of the spectrum was adjusted to be 23×23, and the number of peak additions was changed to 6 (adjusted from defaults of 0.14 %, 45×46, and 4 respectively). The automatic settings in INFOS are a compromise between speed and fit quality, so that highly complex spectra usually require some adjustment. One notices that the diagonal has been fitted with many peaks– this is because although deviations of the experimental peak shape from the calculated shapes are relatively small, the large amplitude of the diagonal enhances those errors, so that INFOS adds additional peaks to refine the fit. Similar over-fitting can also occur elsewhere, although it is possible to reduce this by increasing the 'noise-per-peak' parameter, which will lead to more unresolved peaks being combined.

Comparison of fitting programs and lineshapes

In order to assess when INFOS is most useful, it is compared to the fitting routines provided in NMRPipe (Delaglio et al. 1995). Also, the use of lineshapes determined from acquisition and processing parameters is compared to fitting with simple Gaussian lineshapes. For the first example the C–C DARR (50 ms) spectrum shown in Fig. 6 is refitted using Gaussian lineshapes, and also is refitted with NMRPipe, using Gaussian lineshapes. Note that it is necessary to set groups in NMRPipe to determine which peaks are fit simultaneously, via grouping. This has a critical effect on performance, where it is important that overlapping peaks are placed in the same group, but large groups increase computational time. Here, groups were set by visual inspection so that overlap within a group occurs only at a contour level where significant noise is also visible (resulting in 37 groups, the largest of which had 111 peaks). The results are given in Table 1.

One sees that the best fit is obtained with INFOS, when using lineshapes derived from acquisition and processing parameters, yielding $\chi^2_{\text{reduced}} = 1.28$, versus 1.49 for Gaussian lineshapes. Compared to NMRPipe, the computational time is also significantly improved, and an additional boost is obtained from parallel processing. The lower fit quality obtained using NMRPipe is probably due to some overlap between groups. It is possible to reduce the number of groups in NMRPipe to improve the fit, but it comes at the cost of higher computational time. One must similarly trade computational time for fit quality in INFOS, where smaller sub-spectra lead to faster fitting but lower quality fits, however compensation for overlap of peaks in neighboring sections reduces the cost in fit quality (see sect "Spectrum fitting with a peak list"). In INFOS, lineshapes derived from acquisition and processing parameters also outperformed Gaussian lineshapes for the 400 ms C–C DARR spectrum shown in Fig. 7, with χ^2_{reduced} of 1.75 and 1.99, respectively.

Table 1 Comparison of INFOS and NMRPipe Performance

Program (Lineshapes)	INFOS (Acq./Proc.)	INFOS (Gaussian)	NMRPipe (Acq./Proc.)	NMRPipe (Gaussian)
Spectrum: 50 ms C–C DARR, see Fig. 6				
$\chi^2_{reduced}$	1.28	1.49	–	3.18
Time (1 process)	200 s	200 s	–	1200s
Time (12 processes)	31 s	31 s	–	–
Spectrum: 400 ms C–C DARR, see Fig. 7				
$\chi^2_{reduced}$	1.75	1.99	–	–
Time (12 processes)	2700s	2800s	–	–
Spectrum: Hα–Cα correlation, see Fig. 8				
$\chi^2_{reduced}$	1.26	1.57	5.33	1.27
Time (1 process)	46 s	49 s	1600s	46 s
Time (12 processes)	14 s	14 s	–	–

$\chi^2_{reduced}$ is defined as given in (3), where ν is number of points in the spectrum minus number of fit variables (5 × number of peaks). Computational times are for a server with Intel IvyBridge processors at 2.9 GHz, either using a single process or 12 processes. Computational times are for fitting from an initial peak list, whereas times listed in the text are fits without an initial list

Although INFOS is efficient in fitting spectra with non-Gaussian lineshapes, an alternative approach to optimizing fits is to make lineshapes as Gaussian as possible, using the Lorentz-to-Gauss apodization function (Ernst 1966). Figure 8 shows an Hα–Cα correlation spectrum processed with Lorentz-to-Gauss apodization, and fitted with INFOS vs. NMRPipe. In this case, fitting with lineshapes calculated from acquisition and processing parameters in INFOS yields nearly the same error as fitting with Gaussian shapes in NMRPipe (see Table 1). Because peaks are already nearly Gaussian, it is possible to get high quality fits with Gaussian lineshapes. Computation time using NMRPipe is also now identical to the time using INFOS, since the group size in NMRPipe is relatively small. Note that for a group in NMRPipe and a sub-spectrum in INFOS with the same size and number of peaks, NMRPipe fits considerably faster when using Gaussian lineshapes. However, for crowded spectra, groups in NMRPipe need to include many more peaks than the sub-spectra in INFOS to obtain high quality fits, yielding faster fitting overall by INFOS. Fit quality using acquisition and processing parameters in NMRPipe is considerably lower, and computational time is much longer (see Table 1).

Note that to take full advantage of gains from lineshapes, it is necessary to use the correct type of signal decay in the time domain (for example exponential vs. Gaussian decay).

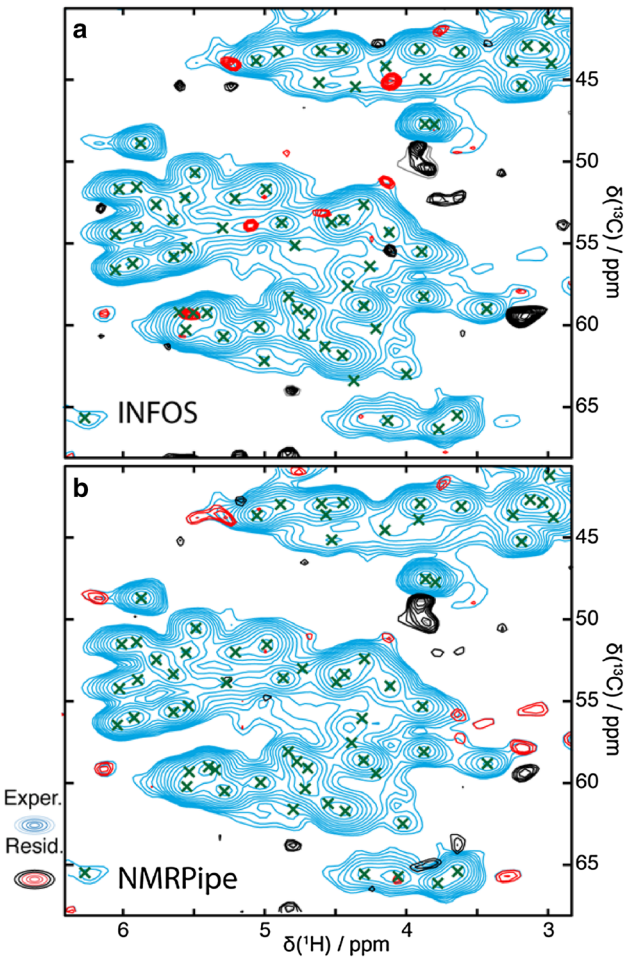

Fig. 8 Fit of an Hα–Cα correlation spectrum using INFOS and NMRPipe. The experimental spectrum was processed with Lorentz-to-Gauss apodization functions in both dimensions to generate Gaussian shaped lineshapes. **a** Shows the experimental spectrum (*blue*/*grey* for positive/negative) and fit residual (*red*/*black* for position/negative), along with peak positions (*green* x's) after fitting with INFOS using lineshapes calculated from acquisition and processing parameters. **b** Shows the same, after fitting with NMRPipe using Gaussian lineshapes. The lowest contour level is drawn at 2 % of the experimental spectrum maximum

In the previous example of an Hα–Cα correlation spectrum, decay parameters were set as follows:

```
par.d1.Broad='Gauss';%Gaussian decay
in 1st (13C) dimension
par.d2.Broad='Lorentz';%Exponential
decay in 2nd (1H) dimension
fit_AP=FitSpec(spec,par);  %Fit the
spectrum
```

Curve fitting and the FitTrace function

The next example demonstrates the benefits of using spectrum fitting to extract peak intensities that will then be used to fit to some type of functional curve. In this case, a series of synthetic spectra with four peaks are considered. Each peak in the spectra decays with a different rate, according to an exponential decay function ($R = 5\ s^{-1}$, $1\ s^{-1}$, $0.5\ s^{-1}$, and $2\ s^{-1}$ for peaks 1–4). Also, noise is added to each spectrum (the first spectrum of the series is shown in Fig. 9a). Then, spectrum fitting is used to extract the peak intensities of the synthetic spectra, and the rate of decay is extracted from those intensities. A synthetic data set is used, so that it is possible to know the *correct* rate and therefore test the method. In order to extract intensities, a reference spectrum is fit (in this case, the first spectrum in the data set), while allowing intensities, linewidths, and positions to vary. Then, all spectra are fit, but with linewidths and positions fixed to match those of the reference fit (this is also the method used in (Smith et al. 2016), where INFOS was used for data analysis). The resulting intensities are plotted for each of the four peaks in Fig. 9b (blue circles). For comparison, intensities are also extracted by simply taking the amplitude of the spectrum at the given peak position (Fig. 9b, red circles). First, one sees that the intensities extracted using spectrum fitting match the real intensities (Fig. 9b, black dashed line) better than those extracted simply from the spectrum amplitude. Second, when fitting the resulting peaks to exponential decay, the rates are usually better reproduced. Note the severity of the disagreement between the rate predicted using only the peak amplitude and the correct rate for peak 4 ($1.25\ s^{-1}$ vs. $2.00\ s^{-1}$). This is caused in part by the strong overlap between peaks 3 and 4, but this problem is almost entirely resolved by spectrum fitting ($2.00\ s^{-1}$ given using spectrum fitting).

If one has a series of spectra for which the amplitudes follow some functional form throughout the series, it is possible to fit the complete series simultaneously. In contrast to the previous example, all spectra are fitted together, so that all spectra have the same peak positions and linewidths, and the amplitudes in each individual spectrum are restricted so that the series of amplitudes follows exponential decay. This option is limited to functions that are characterized by a single variable. However, beyond this limitation, any function may be used for which the derivative with respect to the function variable is defined. In this case, the user provides the function in a structure 'trace' and the spectra in a cell (see manual for details). An initial spectrum fit must be given for this type of fitting.

```
fit0=FitSpec(spec0,par);      %Initial
spectrum is fitted
trace.x=0:.1:50;%Possible values for
relaxation rate
trace.Fx=exp(-t*trace.x); %Exponential
functions
```

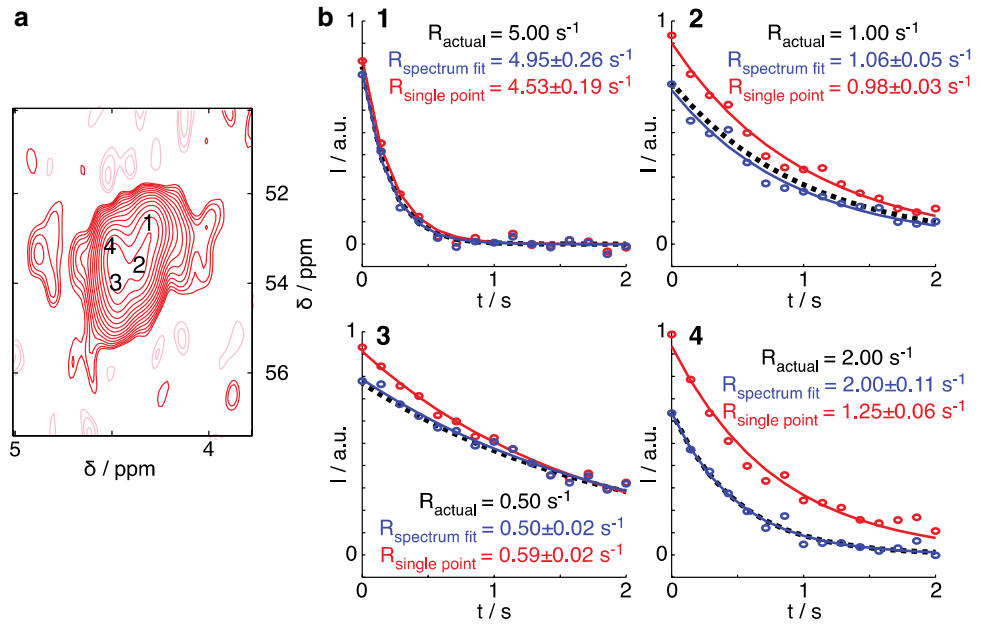

Fig. 9 Curve fitting using spectrum fitting. A time series of 15 spectra is fitted, to extract intensities that are then fit to exponential functions. **a** Shows the first spectrum in the time series, with the peak positions numbered from 1 to 4. Note that noise with RMS that is 3 % of the maximum of the first spectrum is added to all spectra, and the lowest contour level is set to the level of the RMS **b** shows the intensities for each peak (labeled 1–4) extracted using spectrum fitting (*blue circles*), and the fit of those curves to exponential (*blue lines*), as compared to the actual decay curve (*black line*). Intensities are also extracted by simply taking the amplitude of the spectrum in the center of the peak (*red circles*), and these are also fitted to an exponential (*red lines*)

```
%Note that t is a column vector with
the experimental delays
fit=FitTrace(spec,trace,par,fit0);%Fit
series of spectra
```

Figure 10 shows the fitting of a spectrum for which the individual peaks undergo exponential decay. Here, one sees that the behavior of the series of experimental spectra is well reproduced by the calculated spectrum, where the peak heights all are required to decay exponentially.

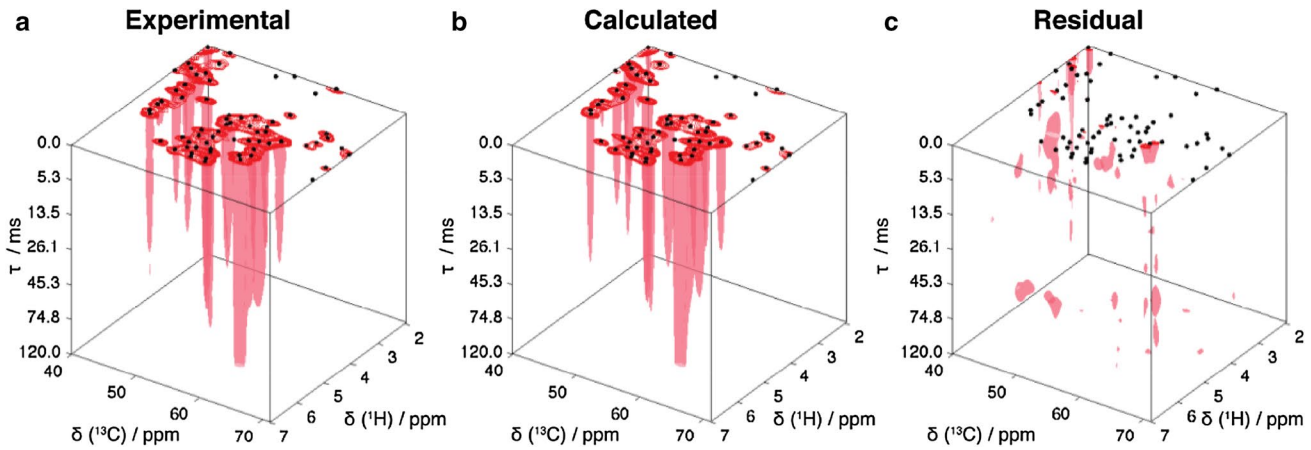

Fig. 10 Simultaneous fit of a series of spectra with exponential signal decay ($R_{1\rho}$ relaxation). Each plot shows the first 2D spectrum (with peak positions marked) over an isosurface of the series of spectra. Then, narrowing of the isosurface indicates signal decay. **a** Shows the series of experimental spectra, with the isosurface plotted at 40 % of the spectrum maximum. **b** Shows the calculated spectrum, also with the isosurface at 40 %. **c** Shows the residual of the fit, with the isosurface plotted at 7 % of the spectrum maximum. The minimum contour level of the first spectrum in c is also plotted at 7 %

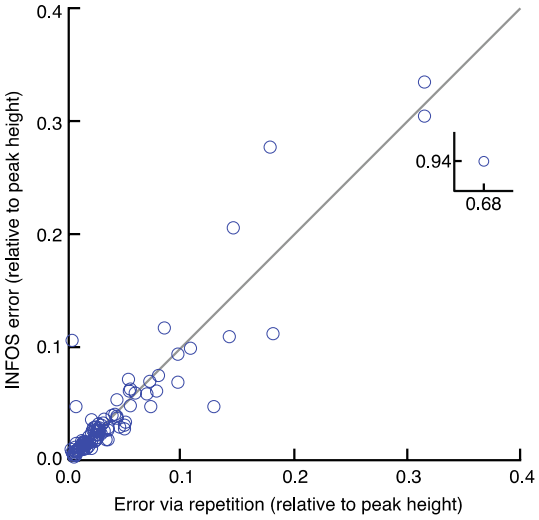

Fig. 11 Comparison of methods of error analysis. An Hα–Ca correlation spectrum (see Fig. 8) has been used to obtain a time series (13 time points), and repeated in triplicate. The standard deviation of peak amplitudes for the repeated time points has been determined and averaged over all time points, to give an estimate of the standard deviation in amplitude for each fitted peak. This value is plotted on the x-axis. The standard deviation of the peak amplitude estimated by INFOS error analysis (using only the first spectrum of the series) is plotted on the y-axis, showing good agreement of the two methods. The inset axis shows an additional point falling outside the main axis, and the grey line shows the diagonal for which the two methods give the same result

Peak amplitude error analysis

In the next two examples, the use of efficient spectrum fitting as a means of error analysis for peak positions and amplitudes is investigated. Error analysis is executed as follows, after first fitting the spectrum:

```
fit=FitSpec(spec,par);%Fit spectrum
err=FitError(fit); %Determine error
```

In the first example, the standard deviation of the peak height for a spectrum is estimated both via repetition and via the error analysis implemented in INFOS (see sect "Functional fitting"). A time series was measured using a 2D Hα–Cα correlation experiment (see Fig. 8), with all time points acquired in triplicate. Therefore, for each peak and each time point, it was possible to calculate the standard deviation of the amplitude. Averaging these together across the time series gives reasonable estimates of the standard deviation of each peak height (data in triplicate alone does not give good enough estimates of the standard deviation). This can be compared to the error estimated using the INFOS error analysis, as applied to only the first spectrum in the data series (note that error analysis and fitting of the time series were performed with only peak amplitudes variable; positions and linewidths were

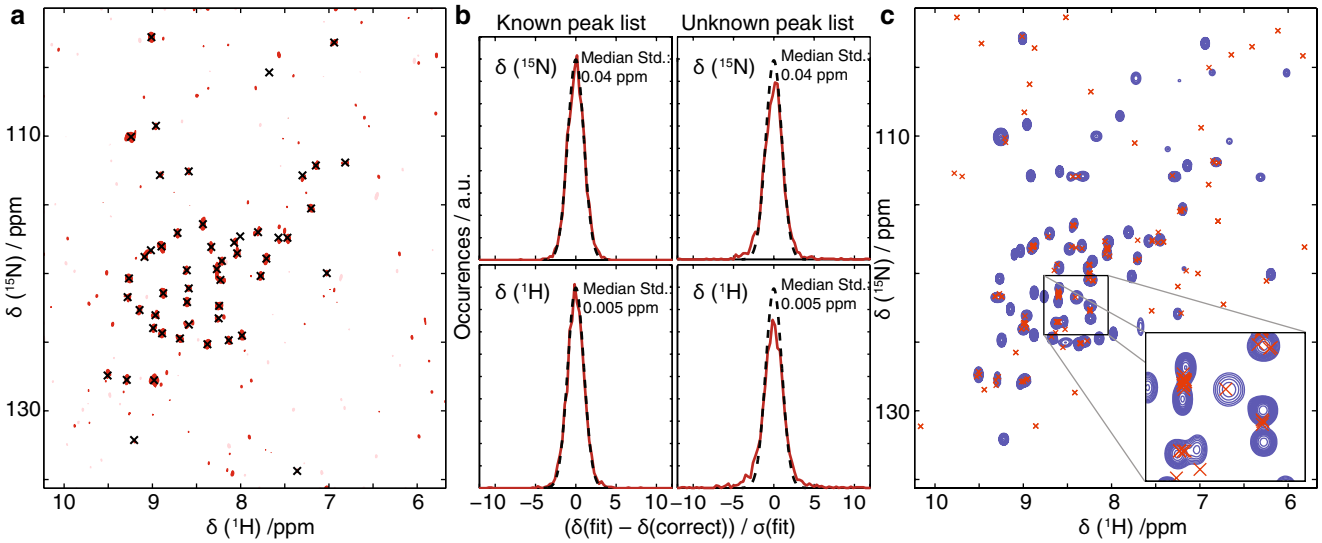

Fig. 12 Estimation of peak position error using INFOS error analysis. A series of 60 synthetic spectra with low signal-to-noise ratio were fitted and analyzed, with an example spectrum shown in **a**. Each of the 60 spectra were fitted and analyzed for error, first using an initial fit that included all peaks in the synthetic spectrum (Known peak list), and second allowing INFOS to determine the peak list (Unknown peak list). **b** Shows histograms of the deviation of the fitted peak positions to the nearest correct peak position, normalized by standard deviation of the peak as determined by error analysis (*red*). For the known peak list, the deviations are in very good agreement with a standard normal distribution (*black*, dashed), as is desired for useful error analysis. However, for the unknown peak list, agreement is lower, with the histogram of peak position deviation being wider than the standard normal distribution, indicating that the error analysis has underestimated the true error. **c** Marks peak positions (*red* x) on the test spectrum (without noise) that fall outside of 4 standard deviations of the correct peak center. These peaks tend to either be falling between two correct peaks (see expanded region), or are simply due to particularly high noise at some position. Example synthetic spectra were adapted from experimental Ubiquitin H–N correlations (Penzel et al. 2015)

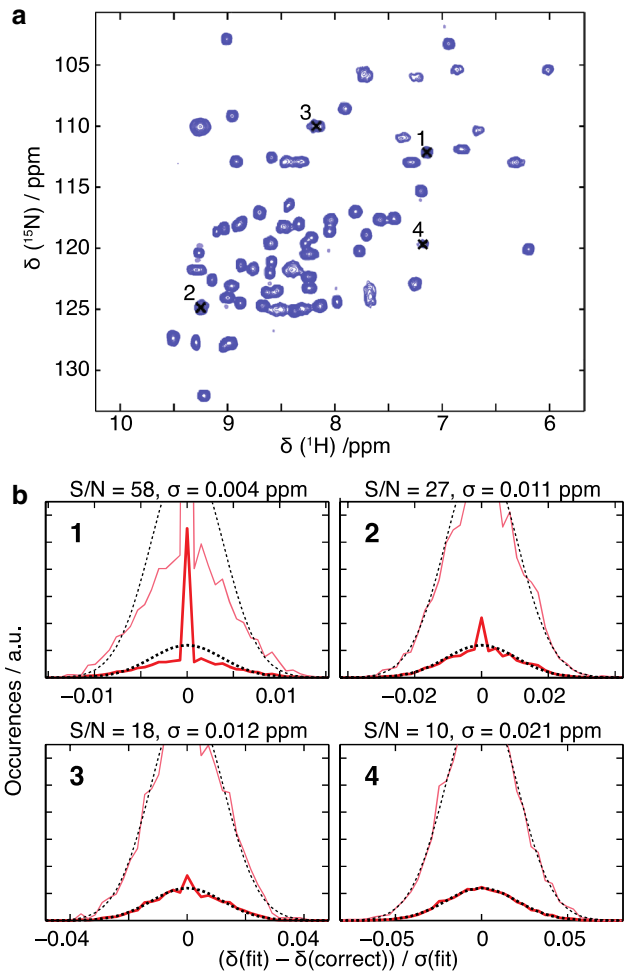

Fig. 13 Distribution functions for peak position error. Error is analyzed for the spectrum shown in **a**, refitting each peak 3000 times to obtain a distribution of peak positions for several peaks (numbered 1–4). The distributions are shown in **b** (*red*, with x7 blowup in lighter *red*), where it can be seen that the larger the signal-to-noise is, the smaller the standard deviation (S/N, σ, *top* of each plot). One also observes that peaks with increasing signal-to-noise deviate from having a normal distribution (*black, dotted line*)

fixed in all fits). In Fig. 11, a scatter-plot compares the two estimation methods for each of 98 peaks in this particular fit. Although there are a few outliers, the agreement of the two methods is quite good, as seen since most points fall near the diagonal (55 points fall below and 43 fall above the diagonal).

Peak position error analysis

One may also estimate error of peak positions using INFOS. In order to test this, a series of synthetic spectra were generated – synthetic spectra are used so that the correct peak positions are known exactly. The spectra had low signal-to-noise, with an example shown in Fig. 12a. These spectra were then fitted by INFOS, and subsequently analyzed for error. For each peak in the fits, the deviation of that peak in each dimension to the nearest correct peak was determined, and then normalized by the standard deviation of that peak (determined by INFOS error analysis). In the ideal case, the collection of all peak deviations analyzed this way should then have a standard normal distribution. This is the result obtained if INFOS is provided an initial peak list, as shown in Fig. 12b, left column, so that there are no peaks from the original spectrum missing in the fit. In fact, this merely verifies that the method of re-fitting only partial spectra as opposed to the full spectrum, as described in 1.5, does not distort the results of error analysis significantly. Nonetheless, it gives a good estimate of the response of a spectrum fit to experimental noise, keeping in mind that the primary source of error then must be white noise in the time domain, as opposed to other experimental fluctuations.

On the other hand, if INFOS is provided with a spectrum for which the signal-to-noise is too low to clearly distinguish all peaks, and INFOS must determine the peak list, then performance is considerably worse, as shown in Fig. 12b, right column. In this case, INFOS makes mistakes in peak picking due to the low signal-to-noise. These mistakes are not accounted for by the error analysis, and therefore lead to peaks in the fit that fall much further away from correct peaks than is predicted by the error analysis. For example, Fig. 12c marks peak locations occurring in the 60 fits for which the fitted peak position falls more than 4 standard deviations away from any correct peak, in one or both dimensions. The first cause of mistakes in peak placement is that two peaks in the original (noise free) spectrum are fitted by a single peak in the fit. This can be seen in the expanded region of Fig. 12c, where a number of peak pairs are fitted with a single peak in between the two original peaks. Additionally, throughout the 60 fits, approximately 35 peaks are placed far away from any peak in the original fit. In fact, these misplaced peaks are entirely expected: based on the cutoff level determined in 'FitSpec', it was expected on average to fit 0.54 "noise peaks" per spectrum (see sect "Noise analysis and determination of fitting settings"), and so for 60 spectra, statistically, it is expected that 32 peaks that are noise will exceed the peak threshold.

Caution should be taken when comparing the strong agreement of peak amplitude error via INFOS analysis and experimental repetition and the less promising agreement between deviations of fitted peaks positions from the true peak positions as compared to the predicted error. The difference between these two analyses is that in the former case, one examines how experimental repetition compares to simulated repetition. Here, one finds strong agreement, showing that if the best method of obtaining error is experimental repetition, then error analysis via INFOS can

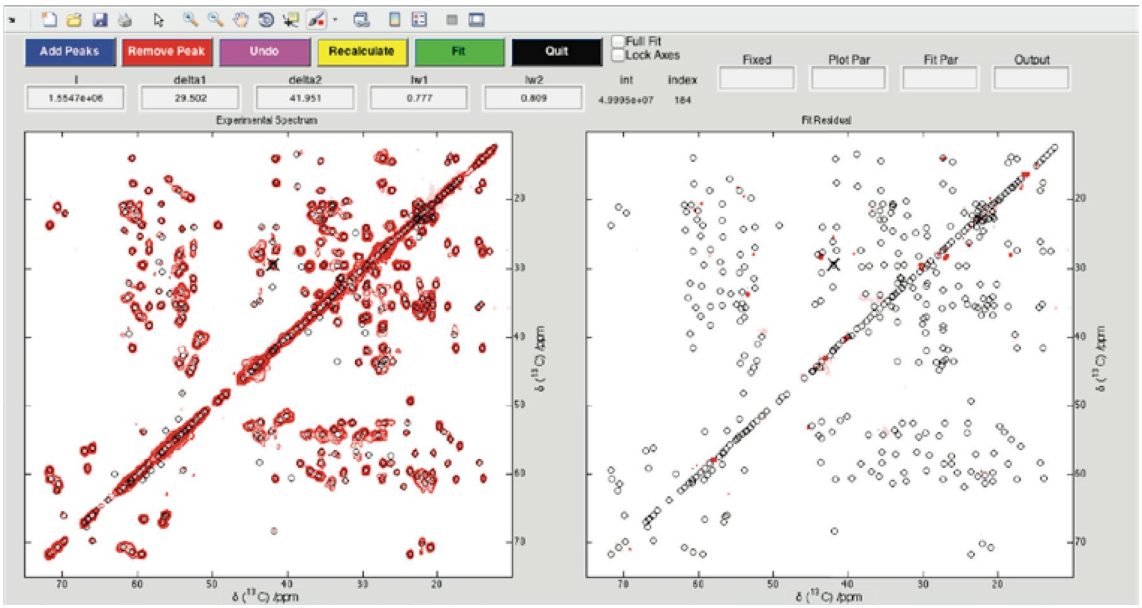

Fig. 14 Graphical user interface of the 'FitEditor' function, which allows interactive manipulation of a fit. The left plot shows the experimental spectrum, and the right plot shows the residual of the fit. Several buttons and entry fields allow editing and control of fitting. A second window shows the calculated spectrum (not shown)

provide a powerful and time saving alternative. The latter case shows that obtaining the distribution of fitted peak position vs. true peak position is not always possible if an accurate fit is not obtained. It is worth noting that experimental repetition also would not resolve this challenge, since one still requires an accurate fit.

When using error analysis, it is also important to know the form of the distribution, since it may not be a normal distribution. Several peaks are analyzed in the spectrum shown in Fig. 13a. The higher the signal-to-noise of a peak is, the smaller the error in position becomes (Fig. 13b). Additionally, peaks with higher signal-to-noise are increasingly not normally distributed. If one is calculating confidence intervals for the position of a peak, then the non-normality significantly complicates analysis; it becomes necessary to use many iterations of error analysis to obtain a well-determined distribution (3000 were used in Fig. 13), whereas if the position is normally distributed, then one simply needs the standard deviation, which can be determined much more quickly.

The FitEditor2D function

Although most fitting can be performed using the 'FitSpec' function, it is also useful to be able to edit a fit interactively. This is possible for 2D spectra using the 'FitEditor2D' function. This function allows basic operations on a fit, such as adding and removing peaks, as well as adjusting parameters of existing peaks. It also allows refitting edited parts of the spectrum or refitting the complete spectrum. The graphical user interface of the 'FitEditor2D' function is shown in Fig. 14. The 'FitEditor2D' function must be called starting from an initial fit.

```
fit0=FitSpec(spec);  %Perform  an  ini-
tial fit
FitEditor2D(fit0);  %Start  the  FitEdi-
tor2D function
```

Conclusions

The INFOS software improves on common fitting methods by using peak lineshapes determined from acquisition and processing settings, as opposed to pure Gaussian or Lorentzian shapes. Pre-calculation and storage of these lineshapes greatly accelerates fitting speed, as is shown via comparison to similar options in NMRPipe. Additionally, INFOS provides an efficient, automated means of refining a peak list to improve spectrum fits without over-fitting. Use of spectrum fitting has been shown to be an effective method of extracting information from overlapping peaks, in particular when extracting amplitudes from a series of spectra. INFOS can both be used to fit individual spectra in a series and subsequently fitting the amplitudes, or fitting an entire series simultaneously, using a user-defined function to describe the series. Finally, spectrum fitting can be used to estimate error on spectrum fit parameters, by simulating experimental repetition, using Monte Carlo methods.

Acknowledgements I would particularly like to thank Matthias Ernst and Beat Meier for supporting and helping to guide my research – research that has necessitated the development of the work presented here. I would further like to thank Susanne Penzel, Joeri Verasdonck, John Ribeiro, and Thomas Bauer for testing and applying the programs presented here, and also additional thanks to Matthias and Susanne for helpful comments while preparing the paper. Thanks to Frank Delaglio for help with fitting with NMRPipe, and import of the NMRPipe spectrum format. This work has been supported by the Swiss National Science Foundation (Grants 200020_146757 and 200020_159707).

References

Alipanahi B, Gao X, Karakoc E, Donaldson L, Li M (2009) PICKY: a novel SVD-based NMR spectra peak picking method. Bioinformatics 25:i268–i275

Bartels C, Xia T-h, Billeter M, Güntert P, Wüthrich K (1995) The program XEASY for computer-supported NMR spectral analysis of biological macromolecules. J Biomol NMR 6:1–10

Bruker Biospin (2016) Topspin.

Brummer ME, Mersereau RM, Eisner RL, Lewine RRJ (1993) Automatic Detection of Brain Contours in MRI Data Sets. IEEE T Med Imaging 12:153–166

Buja A, Hastie T, Tibshirani R (1989) Linear Smoothers and Additive Models. Ann Stat 17:453–455

Caglioti V, Maniezzo V (1995) Mode determination in noisy bimodal images by histogram comparison. Pattern Recogn Lett 16:1237–1248

Cheng Y, Gao X, Liang F (2014) Bayesian peak picking for NMR spectra. Genomics Proteomics Bioinformatics 12:39–47

Cheung M-S, Maguire ML, Stevens TJ, Broadhurst RW (2010) DANGLE: A Bayesian inferential method for predicting protein backbone dihedral angles and secondary structure. J Magn Reson 202:223–233

Chylla RA, Markley JL (1994) Theory and application of the maximum likelihood principle to NMR parameter estimation of multidimensional NMR data. J Biomol NMR 5:245–258

Chylla RA, Volkman BF, Markley JL (1998) Practical model fitting approaches to the direct extraction of NMR parameters simultaneously from all dimensions of multidimensional NMR spectra. J Biomol NMR 12:277–297

Corne SA, Johnson AP (1992) An Artificial Neural Network for Classifying Cross Peaks in Two-Dimensional NMR Spectra. J Magn Res 100:256–266

Cornilescu G, Delaglio F, Bax A (1999) Protein backbone angle restraints from searching a database for chemical shift and sequence homology. J Biomol NMR 13:289–302

de Beer R, van Ormondt D (1992) Analysis of NMR data using time domain fitting procedures. In: Rudin M (ed) In-vivo magnetic resonance spectroscopy I: probeheads and radiofrequency pulses spectrum analysis. Springer, Berlin, Heidelberg, pp 201–248. doi:10.1007/978-3-642-45697-8_7

Delaglio F, Grzesiek S, Vuister GW, Zhu G, Pfeifer J, Bax A (1995) NMRPipe: A multidimensional spectral processing system based on UNIX pipes*. J Biomol NMR 6:277–293

Ernst RR (1966) Sensitivity Enhancement in Magnetic Resonance. Adv Magn Reson 2:1–135

Ernst RR, Bodenhausen G, Wokaun A (1987) Principles of nuclear magnetic resonance in one and two dimensions. Clarendon, Oxford

Garrett DS, Powers R, Gronenborn AM, Clore GM (1991) A Common Sense Approach to Peak Picking in Two-, Three-, and Four- Dimensional Spectra Using Automatic Computer Analysis of Contour Diagrams. J Magn Res 95:214–220

Goddard TD, Kneller DG Sparky 3. University of California, San Francisco

Gullion T, Schaefer J (1989) Rotational-Echo Double-Resonance NMR. J Magn Res 81:196–200

Hoch JC, Stern AS (1996) NMR Data Processing. John Wiley & Sons. Inc., Hoboken

Hu K-N, Qiang W, Tycko R (2011) A general Monte Carlo/simulated annealing algorithm for resonance assignment in NMR of uniformly labeled biopolymers. J Biomol NMR 50:267–276

Hughes I, Hase T (2010) Measurements and Their Uncertainties: A Practical Guide to Modern Error Analysis. Oxford University Press, New York

Johnson BA, Blevins RA (1994) NMRView: a computer program for the visualization and analysis of NMR data. J Biomol NMR 4:603–614

Keller R (2004) The Computer Aided Resonance Assignment Tutorial. Cantina Verlag, Goldau, Switzerland

Koradi R, Billeter M, Engeli M, Güntert P, Wüthrich K (1998) Automated peak picking and peak integration in macromolecular NMR spectra using AUTOPSY. J Magn Res 135:288–297

Levenberg (1944) A Method for the Solution of Certain Non-Linear Problems in Least Squares. Q Appl Math 2:164–168

López-Méndez B, Güntert P (2006) Automated Protein Structure Determination from NMR Spectra. J Am Chem Soc 128:13112–13122

Mandel AM, Akke M, Palmer AGI (1995) Backbone Dynamics of Escherichia coli Ribonuclease HI: Correlations with Structure and Function in an Active Enzyme. J Mol Biol 246:144–163

Marquardt DW (1963) An Algorithm for Least-Squared Estimation of Nonlinear Parameters. J Soc Indust Appl Math 11:431–441

Massiot D, Fayon F, Capron M, King I, Le Calvé S, Alonso B, Durand J-O, Bujoli B, Gan Z, Hoatson G (2002) Modelling one- and two-dimensional solid-state NMR spectra. Magn Res Chem 40:70–76

Mathworks (2013a) MATLAB and Optimization Toolbox Release 2013b. The Mathworks, Inc., Natick, Massachusetts, United States

Mathworks (2013b) MATLAB Release 2013b. The Mathworks, Inc., Natick, Massachusetts, United States

Metropolis N, Ulam S (1949) The Monte Carlo Method. J Amer Statistical Assoc 44:335–341

Morin S, Linnet TE, Lescanne M, Schanda P, Thompson GS, Tollinger M, Teilum K, Gagne S, Marion D, Griesinger C, Blackledge M, d'Auvergne EJ (2014) relax: the analysis of biomolecular kinetics and thermodynamics using NMR relaxation dispersion data. Bioinformatics 30:2219–2220

Norris M, Fetler B, Marchant J, Johnson BA (2016) NMRFx Processor: a cross-platform NMR data processing program. J Biomol NMR 65:205–216

Palmer AG, Rance M, Wright PE (1991) Intramolecular Motions of a Zinc Finger DNA-Binding Domain from Xfin Characterized by Proton-Detected Natural Abundance 13 C Heteronuclear NMR Spectroscopy. J Am Chem Soc 113:4371–4380

Penzel S, Smith AA, Agarwal V, Hunkeler A, Org M-L, Samoson A, Böckmann A, Ernst M, Meier BH (2015) Protein resonance assignment at MAS frequencies approaching 100 kHz: a quantitative comparison of J-coupling and dipolar- coupling-based transfer methods. J Biomol NMR 63:165–186

Rieping W, Habeck M, Bardiaux B, Bernard A, Malliavin TE, Nilges M (2007) ARIA2: automated NOE assignment and data integration in NMR structure calculation. Bioinformatics 23:381–382

Schwieters CD, Kuszewski JJ, Tjandra N, Clore GM (2003) The Xplor-NIH NMR molecular structure determination package. J Magn Res 160:65–73

Serrano P, Pedrini B, Mohanty B, Geralt M, Herrmann T, Wuthrich K (2012) The J-UNIO protocol for automated protein structure determination by NMR in solution. J Biomol NMR 53:341–354

Siemer AB, Ritter C, Steinmetz MO, Ernst M, Riek R, Meier BH (2006) 13 C, 15 N Resonance assignment of parts of the HET-s prion protein in its amyloid form. J Biomol NMR 34:75–87

Sijbers J, Poot D, den Dekker AJ, Pintjens W (2007) Automatic estimation of the noise variance from the histogram of a magnetic resonance image. Phys Med Biol 52:1335–1348

Skinner SP, Fogh RH, Boucher W, Ragan TJ, Mureddu LG, Vuister GW (2016) CcpNmr AnalysisAssign: a flexible platform for integrated NMR analysis. J Biomol NMR 66:111–124

Slotboom J, Boesch C, Kreis R (1998) Versatile frequency domain fitting using time domain models and prior knowledge. Magn Reson Med 39:899–911

Smith SW (1999) The Scientist and Engineer's Guide to Digital Signal Processing. California Technical Publishing, USA

Smith AA, Testori E, Cadalbert R, Meier BH, Ernst M (2016) Characterization of fibril dynamics on three timescales by solid-state NMR. J Biomol NMR 65:171–191

Sodickson DK, Levitt MH, Vega S, Griffin RG (1993) Broad band dipolar recoupling in the nuclear magnetic resonance of rotating solids. J Chem Phys 98:6742

Sorenson DC (1982) Newtons's Method with a Model Trust Region Modification. SIAM J Numer Anal 19:409–426

van Dijk JE, Mehlkopf AF, van Ormondt D, Bovée WMMJ (1992) Determination of Concentrations by Time Domain Fitting of Proton NMR Echo Signals Using Prior Knowledge. Magn Reson Med 27:76–96

Van Huffel S, Chen H, Decanniere C, Van Hecke P (1994) Algorithm for Time-Domain NMR Data Fitting Based on Total Least Squares. J Magn Res Ser A 119:228–237

Van Melckebeke H, Wasmer C, Lange A, AB E, Loquet A, Böckmann A, Meier BH (2010) Atomic-Resolution Three-Dimensional Structure of HET-s(218–289) Amyloid Fibrils by Solid-State NMR Spectroscopy. J Am Chem Soc 132:13765–13775

van den Boogart A, Ala-Korpela M, Jokisaari J, Griffiths JR (1994) Time and Frequency Domain Analysis of NMR Data Compared: An Application to 1D 1 H Spectra of Lipoproteins. Magn Reson Med 31:347–358

Vranken WF, Boucher W, Stevens TJ, Fogh RH, Pajon A, Llinas M, Ulrich EL, Markley JL, Ionides J, Laue ED (2005) The CCPN data model for NMR spectroscopy: development of a software pipeline. Proteins 59:687–696

Wasmer C, Lange A, Van Melckebeke H, Siemer AB, Riek R, Meier BH (2008) Amyloid Fibrils of the HET-s(218–289) Prion Form a β Solenoid with a Triangular Hydrophobic Core. Science 319:1523–1526

Waudby CA, Ramos A, Cabrita LD, Christodoulou J (2016) Two-Dimensional NMR Lineshape Analysis. Sci Rep 6:24826