ComparativeDeerAnalyzer 2.0

G. Jeschke
ETH Zürich, Department of Chemistry and Applied Biosciences, gjeschke@ethz.ch

Abstract

Automatic processing of primary DEER/PELDOR data to distance distributions is desirable for avoiding bias by subjective user decisions. ComparativeDeerAnalyzer (CDA) performs such automatic analysis by the neural network approach DEERNet and by single-step background fitting and Tikhonov regularization as implemented in DeerLab. Mean distance and standard deviation of the distribution are derived by simultaneous fitting of a single-Gaussian distribution and background decay to the data. The results are compared and a consensus distribution is derived. The program generates a standardized report, including quality parameters of the input data set, individual vector graphic figures in PDF format. Fit results and distance distributions are output in a standardized file format.

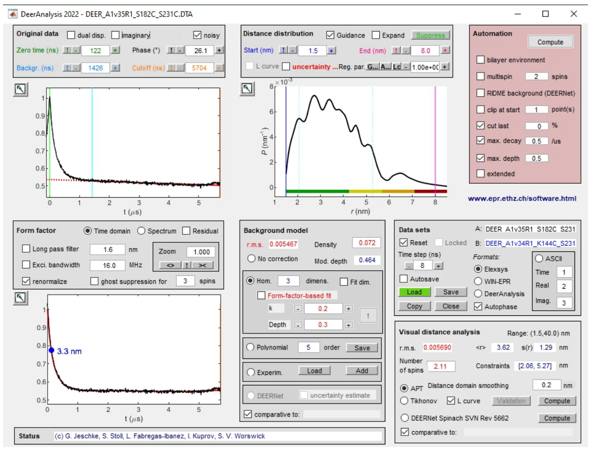
1 General Considerations

DEER, also named PELDOR, is an electron paramagnetic resonance (EPR) experiment for measuring the distribution of distances between electron spins. The primary data is a time trace of integral echo intensity that features dipolar modulation due to close coupling partners and a background decay due to remote spins. In most application work, the electron spins belong to two nitroxide radicals [3]. The modulation depth ranges between 0.05 and 0.5. The data is acquired deadtime free with the four-pulse DEER sequence and quadrature detection, providing a complex signal. Such data can be processed automatically. The pre-processing includes phase correction, determination of zero time, and possibly cutoff of compromised data points at the beginning and end of the trace. The thus generated signal can be analyzed in terms of a distance distribution, if orientation selection is negligible. This analysis involves separation of background and modulation and conversion of the modulation component to the distance distribution. The last step is an ill-posed problem, meaning that small deviations of the input signal from the ideal expectation can cause large changes in the output signal. Three major approaches exist to stabilize the solution: (i) Regularization, (ii) Multi-Gauss fitting, (iii) Neural network analysis. They all involve model assumptions. Regularization, at least in the usually applied form of Tikhonov regularization, assumes uniform resolution across the whole distance range. This resolution is governed by a regularization parameter that controls smoothing of the distribution [1]. Multi-Gauss fitting assumes that the distance distribution can be modelled as a linear combination of a small number of Gaussian components. The number of such components is selected by a statistical criterion. Neuronal network analysis assumes that the distance distribution and background fall into the range represented by the training set and that the trained network is reasonably robust against deviations of the signal from ideal expectation [4, 2].

Uncertainty of the result is determined by quality of the input data, mainly by its signal-to-noise ratio (SNR), and by the ratio of the maximum observation time to the third power of the longest distance. Additional uncertainty can ensue from poor choices of processing parameters by the user and from the data analysis approach taken. Confirmation bias ensues if users select parameters such that the result becomes consistent with their expectations. Confirmation bias can be avoided by renouncing all user-adjustable parameters except for reliable information on the sample and on measurement conditions. This approach is followed by the neural network DEERNet 2.0 [2]. ComparativeDeerAnalyzer (CDA) 2.0 adheres to the same principle. Choice of a data analysis approach can introduce model bias. CDA assesses and reduces model bias by comparing results from three approaches, neuronal network analysis [4, 2], Tikhonov regularization as implemented in DeerLab [1], and computation of mean distance and standard deviation by fitting of a single Gaussian distribution. CDA 2.0 conforms to the community standards that were recently set for data analysis and reporting [3]. In particular, the output fully documents data processing and uncertainties.

2 Work flow

2.1 Using CDA in DeerAnalysis2022


DeerAnalysis 2022 - DEER_A1v35R1_S182C_S231C.DTA

Original data □ dual disp. □ imaginary ☑ noisy
Zero time (ns) 122 Phase (°) 26.1
Backgr. (ns) 1426 Cutoff (ns) 5704

Distance distribution ☑ Guidance □ Expand
Start (nm) 1.5 End (nm) 8.0
□ L curve □ uncertainty ... Reg. par. 1.00e+0(

Automation
□ bilayer environment
□ multispin 2 spins
□ RIDME background (DEERNet)
□ clip at start 1 point(s)
☑ cut last 0 %
☑ max. decay 0.5 /us
☑ max. depth 0.5
□ extended

www.epr.ethz.ch/software.html

Form factor ● Time domain ○ Spectrum □ Residual
□ Long pass filter 1.6 nm
□ Exci. bandwidth 16.0 MHz
Zoom 1.000
☑ renormalize □ ghost suppression for 3 spins

Background model
r.m.s. 0.005467 Density 0.072
○ No correction Mod. depth 0.464
● Hom. 3 dimens. □ Fit dim.
□ Form-factor-based fit
k 0.2
Depth 0.3
○ Polynomial 5 order
○ Experim.
○ DEERNet □ uncertainty estimate
☑ comparative to:

Data sets
A: DEER A1v35R1 S182C S231
B: DEER_A1v34R1_K144C_S231
☑ Reset □ Locked
Time step (ns) 8
□ Autosave
Formats: ● Elexsys ○ WIN-EPR ○ DeerAnalysis ☑ Autophase
○ ASCII Time 1 Real 2 Imag. 3

Visual distance analysis Range: (1.5,40.0) nm
r.m.s. 0.005690 <r> 3.62 s(r) 1.29 nm
Number of spins 2.11 Constraints [2.06, 5.27] nm
● APT Distance domain smoothing 0.2 nm
○ Tikhonov ☑ L curve
○ DEERNet Spinach SVN Rev 5662
☑ comparative to:

Status (c) G. Jeschke, S. Stoll, L. Fabregas-Ibanez, I. Kuprov, S. V. Worswick

Figure 1: Graphical user interface of DeerAnalysis2022. Automatic analysis is started by pushing the top left `Compute` button.

CDA is the engine for automatic processing in DeerAnalysis2022. The `Compute` button in the **Automation** panel (Figure 1 is enabled as soon as a dataset is loaded. For most datasets, analysis is as easy as pushing this button. For relaxation-induced dipolar modulation enhancement (RIDME) data (only spin $S = 1/2$) the `RIDME background` checkbox. In this case, only DEERNet neural network analysis is performed. For symmetric multispin arrangements, as they can occur in protein homooligomers, the `multispin` checkbox should be enabled and the number of spins specified in the corresponding edit field. For reconstituted membrane proteins, it is advantageous to tick the `bilayer environment` checkbox. This specifies a lower background dimension in case that the background must be analyzed by bilevel optimization with regularization. This checkbox has no effect if DEERNet considers its background separation as reliable.

The remaining elements of the `Automation` panel are required only if a first automated analysis indicates problems with the dataset. Some datasets feature compromised data points are the start or end (Figure 2. Abnormally high intensity at the start results from a too short initial delay between the second observer pulse and the pump pulse. If only a single point at the beginning is higher than the maximum of the dipolar evolution function, CDA usually recognizes the problem and corrects it automatically. If the problem is very serious or not automatically recognized, the user can specify the number of points to be clipped at the beginning. The corresponding checkbox then needs to be activated.

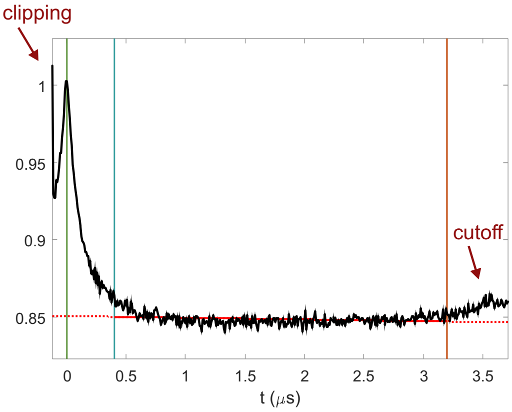


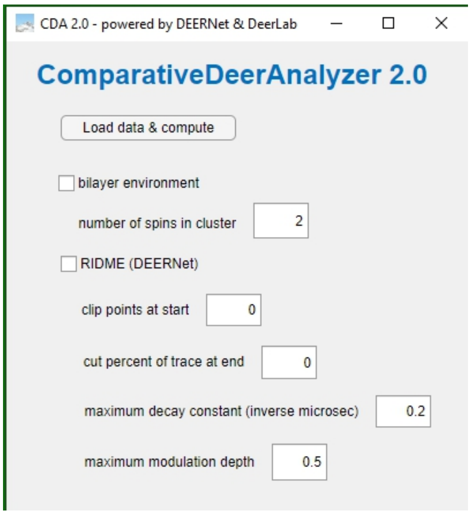
Figure 2: Clipping and cutting. This dataset has a spurious data point at the start, which is automatically clipped and an excitation band overlap artefact at the end, which is mostly cut off.

Second, data may contain an excitation band overlap at the end of the trace. Experts can analyze such data by a multi-pathway approach in DeerLab. CDA implements an automatic, but rather conservative cutoff. The longest consecutive segment and the end of the data is cut off, where more than half ob the points deviate from an initial DEERNet fit by more than two times the noise level. This segment is limited to 15% of total trace length. The remaining deviations are usually very well tolerated by DEERNet and do not substantially affect regularization with a standard kernel. If absolutely necessary, the user can specify a longer cutoff than is automatically applied. Note that many datasets do not require any cutoff.

In rare cases, standard parameter ranges in the DeerLab workflow may be insufficient. This happens only if DEERNet background separation fails and either background decay is very strong or modulation depth is abnormally high. If automatic analysis fails and one of these problems might be the cause, the limits for these values can be increased. This should not be done preemptively.

If the checkbox `Extended` is activated, background for regularization is computed by bilevel optimization even if DEERNet background separation was considered as successful. In this mode, confidence intervals for the regularized distance distribution result from independent variation of decay rate and modulation depth of the background function rather than from the ensemble of DEERNet background guesses. Such computation takes much longer and is not considered to be necessary. After computation of automatic analysis, the report is displayed in a PDF reader and the **Automation** panel turns green.

2.2 Using the CDA App from Matlab


CDA 2.0 - powered by DEERNet & DeerLab
ComparativeDeerAnalyzer 2.0
Load data & compute

□ bilayer environment

number of spins in cluster 2

□ RIDME (DEERNet)

clip points at start 0

cut percent of trace at end 0

maximum decay constant (inverse microsec) 0.2

maximum modulation depth 0.5

Figure 3: Graphical user interface of the CDA App.

The CDA App can be run separately by typing **CDA** at the Matlab prompt. Note that the whole DeerAnalysis package with all its subdirectories must be on the Matlab path. The graphical user interface is very basic and offers only the choices of the **Automation** panel of DeerAnalysis (Figure 3). When you click the `Run` button, a window opens for data file selection. The PDF report file is generated by the App as well, but is not automatically opened.

2.3 Using the CDA App as a standalone program

The CDA App is available in a compiled version that does not require a Matlab license. During installation, you need to download the Matlab Runtime engine, which is free. The user interface is the same as when running the CDA App from inside Matlab. The standalone CDA App can also be used if licenses for Matlab toolboxes, such as the Deep Learning toolbox, Reinforcement Learning toolbox, or Optimization toolbox are missing.

2.4 Calling CDA in scripts

The call

```
dataset = comparative_deer_analysis(fname)
```

initiates fully automated analysis and returns the results in variable `dataset`. Results are also automatically saved to files, including the report. The report is not automatically opened in a PDF viewer.

Options can be provided as a second argument. The original output of DEERNet can be retrieved as a second output argument, the original output of the last DeerLab workflow run as a third output argument. Empty output is returned upon complete failure. Depending on options, part of the output may be empty.

3 Input and Output

3.1 Input

Comparative DEER Analyzer requires input data in Bruker EleXsys (*.DTA, *.DSC), Bruker WinEPR (*.spc, *.dat) or text (ASCII) format (*.dat). ASCII data must contain the time axis (in nanoseconds or microseconds) in the first column and the real part of the signal in the second column. It is strongly recommended to acquire data with quadrature detection and include the imaginary part as the third column.

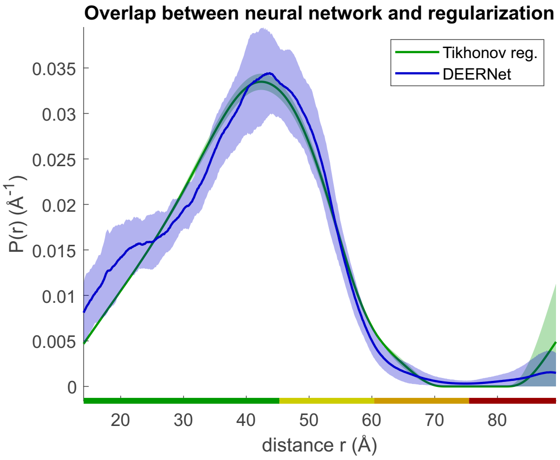


Figure 4: Superposition plot of the DEERNet (blue) and Tikhonov regularized (blue) distance distributions. The color bar indicates reliability ranges that should be considered together with the uncertainty bands. In general, features in the range marked red should not be interpreted.

3.2 Output

3.2.1 Report in PDF format

CDA automatically saves all essential output to files. The report is a PDF file that contains a superposition plot of the DEERNet and regularized distance distributions (Fig. 4) as well as a plot of the consensus distribution ((Fig. 5)). Fits and background separation are shown for DEERNet neural network analysis, DeerLab Tikhonov regularization, and for reconstruction from the consensus distribution. The final pages report quality parameters and metadata as well as warnings and the location of the output files. The individual figures are also saved as vector graphics in PDF format.

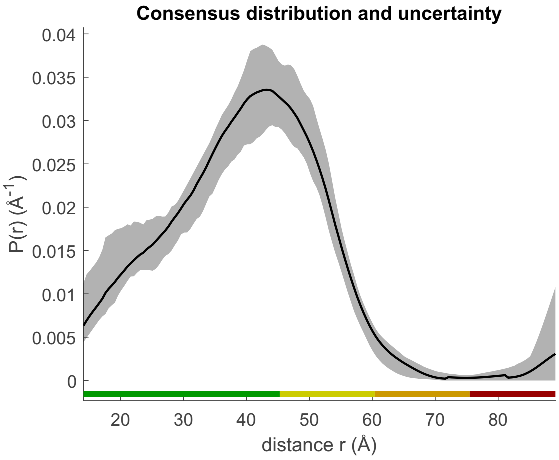


Figure 5: Consensus distance dsitribution plot of the DEERNet and Tikhonov regularized distance distributions. The uncertainty band includes uncertainty of both analyses in order to account for model bias.

3.2.2 Numerical data

Numerical data are output as comma-separated value files, which can be read easily by most software. Column headers in the first line are preceded by the percent character (`%`). This ensures that the data can be directly read with the Matlab `load` function. Excel also reads comma-separated value files, although international versions may be problematic. For instance, the German version expects and writes a semicolon instead of a comma.

Numerical outputs are the distance distribution with 95% confidence interval and the time-domain data with fit and reconstructed background. They exist for the consensus distribution, the DEEER-Net output, and the last DeerLab output (after optimizing all parameters). If an approach failed or is inappropriate for a selected option, the corresponding numerical output is missing. In particular, this applies to the RIDME option (no DeerLab output) and for data where DEERNet does not consider background or distribution as reliable (no DEERNet output). Except for the RIDME option, incomplete output indicates problems with the experimental data. The PDF report specifies these problems further and may recommend that parts of the output or even the whole output are not used in application work. The output is still provided to enable analysis of the problem.

Distance distribution file have four columns. The distance axis (first column) is given in units of Å, and the probability (second column) is given as probability density (unit $1/AA$). The third column contains the lower bound and the fourth column the upper bound of the confidence interval.

Fit files also have four columns. The time axis (first column) is given in units of μs. The second column reports the experimental data, the third column the fit, and the fourth column the background.

CDA also writes a binary Matlab output file with the full output of DEERNet and of the last DeerLab run (the one with optimized parameters). This allows for diagnosis, as the data are also saved in this form if problems were detected and they are not output in comma-separated value files.

3.2.3 Metadata

Metadata are saved as a comma-separated value file as well. In this file, each line reports an identifier of a parameter (first column) and the value of this parameter (second column). The metadata file conforms to the reporting requirements for DEER/PELDOR data [3].

3.3 Interpreting CDA output

When called from a Matlab script, CDA provides more extensive output that allows for additional quality assessment and for diagnosing what went wrong if a dataset could not be processed. Interpretation of this output requires substantial expertise in processing of pulsed dipolar spectroscopy data.

Callers, who do not have such expertise should not use any output beyond the following fields of variable `dataset`.

`.r` distance axis

`.P_comparative` the distance distribution

`.P_comparative_lb` lower bound of the 95% confidence interval of the distance distribution

`.P_comparative_ub` upper bound of the 95% confidence interval of the distance distribution

`.t` time axis after pre-processing (microseconds)

`.input_trace` the real input data trace after preprocessing

`.fit` the final fit

`.background` the best background estimate

`.r_mean` mean distance, reliable only if `.r_std` is positive

`.r_std` standard deviation of the mean distance, reliable only if positive

If any of these fields does not exist or is empty, this aspect could not be reliably assessed. In such a case, an expert may still be able to interpret the original data with some caution. The data should not be used without approaching such an expert.

The following metadata fields of `dataset` can be used for further information:

`.mod_depth` modulation depth

`.SNR` signal-to-noise ratio relative to modulation depth

`.t0` dipolar evolution zero time on the original time axis (ns)

`.dt` time increment of the original time axis (ns)

`.tmax` maximum time after cutoff (ns)

`.phase` phase correction (degree) that was applied to original data

`.regpar` regularization parameter

`.selregpar` selection criterion for regularization parameter

`.backg_dim` background dimension (stretched exponential)

4 Installation

The installation packages are available at epr.ethz.ch/software. Most users will download DeerAnalysis2022, which contains all Matlab sources, as well as the required DEERNet neuronal network files. This package comes as a single ZIP file that you need to unpack. You should add the DEERAnalysis directory with all its subdirectories to the Matlab path. DeerAnalysis requires Matlab 2020b or later and relies on the Deep Learning and Reinforcement Learning Toolboxes of Matlab. Dependence on the Parallel Computing toolbox was removed. CDA uses the Report Generator toolbox for creating a PDF report. If this toolbox is unavailable, CDA saves individual figures as PDF files and provides the report in plain text format without figures.

If you do not have Matlab or only an older version or you are missing Toolboxes, you can use the packaged CDA app (currently only for Windows, other platforms can be compiled upon request). The ZIP file contains this manual as well as a Windows executable *CDAInstaller_web.exe* that runs an installer. If the corresponding (free) Matlab runtime library is not already installed, it is downloaded during installation. If you install a newer version of the ComparativeDEERAnalyzer app, it is recommended to uninstall the older version before.

5 Performance

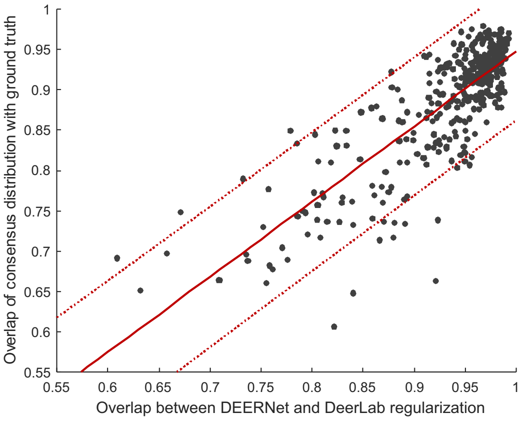


Figure 6: Correlation between the overlap of DEERNet neural network and DeerLab regularization solutions on the one hand and ground truth distance distributions on the other hand. The solid red line is a linear fit and the dotted lines denote the 95% confidence interval for predicting overlap with ground truth.

Perfomance of CDA 2.0 was tested with a set of 75 distance distributions that cover typical application scenarios [3]. Primary DEER datasets were simulated from these distributions at signal-to-noise (SNR) ratios of 8, 16, 32, 50, and 100 with respect to ground truth and were analyzed by CDA 2.0. Over the whole set of 375 DEER traces, the mean overlap of the consensus distribution with ground truth was 0.881 when taken as a geometric average and 0.885 when taken as an arithmetic average. DEERNet on its own performed at a similar level (geometric average 0.876, arithmetic average 0.880). At high SNR, differences between geometric and arithmetic average of overlap with ground truth and difference between the consensus and DEERNet distributions are insignificant. Mean overlap is 0.940 at SNR of 100, 0.924 at SNR of 50, and 0.913 at SNR of 32.

Figure 6 shows how the overlap between DEERNet and DeerLab solutions correlates with the overlap between the consensus solution and ground truth. The correlation coefficient is 0.817. CDA 2.0 uses a linear fit of these data (solid red line) for predicting overlap of the consensus solution with ground truth. Given the moderate correlation, this prediction is given as a range (95% confidence interval) visualized by the dotted red lines in Figure 6.

Acknowledgement

Comparative DEER Analyzer is based on computational engines developed by the Kuprov [4, 2] as

well as Stoll and Jeschke [1] groups.

References

[1] L. Fábregas Ibáñez, G. Jeschke, and S. Stoll. "DeerLab: a comprehensive software package for analyzing dipolar electron paramagnetic resonance spectroscopy data". In: *Magnetic Resonance* 1.2 (2020), pp. 209–224. DOI: `10.5194/mr-1-209-2020`. URL: `https://mr.copernicus.org/articles/1/209/2020/`.

[2] Jake Keeley et al. "Neural networks in double electron-electron resonance: a practical guide". In: *submitted* x.y (2021), aaa–bbb.

[3] Olav Schiemann et al. "Benchmark Test and Guidelines for DEER/PELDOR Experiments on Nitroxide-Labeled Biomolecules". In: *J. Am. Chem. Soc.* 143.43 (NOV 3 2021), 17875–17890. ISSN: 0002-7863. DOI: `{10.1021/jacs.1c07371}`.

[4] Steven G. Worswick et al. "Deep neural network processing of DEER data". In: *Science Advances* 4.8 (2018). DOI: `10.1126/sciadv.aat5218`. eprint: `https://advances.sciencemag.org/content/4/8/eaat5218.full.pdf`.