



Data Preparation Protocol for Low Signal-to-Noise Ratio Fluorine-19 MRI

Ludger Starke, Thoralf Niendorf, and Sonia Waiczies

Abstract

Fluorine-19 MRI shows great promise for a wide range of applications including renal imaging, yet the typically low signal-to-noise ratios and sparse signal distribution necessitate a thorough data preparation.

This chapter describes a general data preparation workflow for fluorine MRI experiments. The main processing steps are: (1) estimation of noise level, (2) correction of noise-induced bias and (3) background subtraction. The protocol is supplemented by an example script and toolbox available online.

This chapter is based upon work from the COST Action PARENCHIMA, a community-driven network funded by the European Cooperation in Science and Technology (COST) program of the European Union, which aims to improve the reproducibility and standardization of renal MRI biomarkers. This analysis protocol chapter is complemented by two separate chapters describing the basic concept and experimental procedure.

Key words Magnetic resonance imaging (MRI), Fluorine, Data preparation, Mice, Rats

1 Introduction

Fluorine-19 (^{19}F) MR methods have found their application in a wide range of biomedical research areas including renal imaging [1–8]. The sensitivity constraints, low signal-to-noise ratio (SNR) and sparse signal distribution typical of fluorine-19 MRI necessitate a thorough data preparation. The main processing steps are: (1) estimation of noise level, (2) correction of noise-induced bias and (3) background subtraction. While variations on this theme are common in fluorine MRI studies, a standardized method of data preparation and analysis with well-documented implementation will improve accuracy and reproducibility of reported results.

Estimating the noise level is a prerequisite for all subsequent processing steps. Assuming data from a single channel RF coil, the signal in MR magnitude images follows a Rician distribution [9, 10] (*see Note 1* for information on multichannel RF coils). At $\text{SNR} > 10$, this distribution closely resembles a Gaussian

distribution. Yet, when analyzing low SNR fluorine MRI data, its diverging properties need to be taken into account. The MRI magnitude signal is computed from complex data, which contain signal with positive and negative values, and is corrupted by zero mean Gaussian noise. Mapping positive and negative signals to the positive magnitude results in a pronounced upward bias at low SNRs. Additionally, the standard deviation of the measured signal is reduced (Fig. 1a). To ensure accurate measurement of signal intensities, the noise-induced bias must be corrected. It is also essential to consider the signal level-dependence of the measured standard deviation for correct noise level estimation.

Background subtraction is the process of classifying image voxels as either signal or noise. Most fluorine MRI studies are dealing with signals that are close to the detection threshold; in this case the background subtraction method strongly influences the sensitivity of the measurement and most importantly the reliability and reproducibility of the obtained results. Ideally, reporting of the used background subtraction method should include the expected false positive rate (FPR).

In this chapter we present a data preparation protocol for low SNR fluorine MRI data (Fig. 2). An open source toolbox is used for bias correction (Fig. 2b) of the magnitude data (Fig. 2a). We expand upon the common method of SNR-based thresholding [2, 4, 11] (Fig. 2c) by proposing an additional outlier removal processing step (Fig. 2d). The shown example is based on a dataset with fluorine nanoparticle labeled immune cells showing the sites of inflammation in a mouse model of multiple sclerosis (*see* Setup 4 in Starke et al. [12] for details). The data preparation protocol is equally applicable to low SNR ^{19}F MRI of the kidney.

A self-contained script reproducing Fig. 2 and illustrating the data preparation step-by-step, from noise level estimation to generation of a proton/fluorine-19 overlay is provided online together with the example data.

This analysis protocol chapter is complemented by separate chapters describing the basic concept and experimental procedure, which are part of this book.

2 Materials

2.1 Software Requirements

The processing steps outlined in this protocol require:

1. A software development environment. The provided code examples follow the syntax of *MATLAB*[®] (The MathWorks, Natick, Massachusetts, USA; [mathworks.com](https://www.mathworks.com)) and *Octave* ([gnu.org/software/octave/](https://www.gnu.org/software/octave/)). The code could also be adapted for other scientific software development environments like *Python* ([python.org/](https://www.python.org/)) or *Julia* ([julialang.org/](https://www.julialang.org/)).

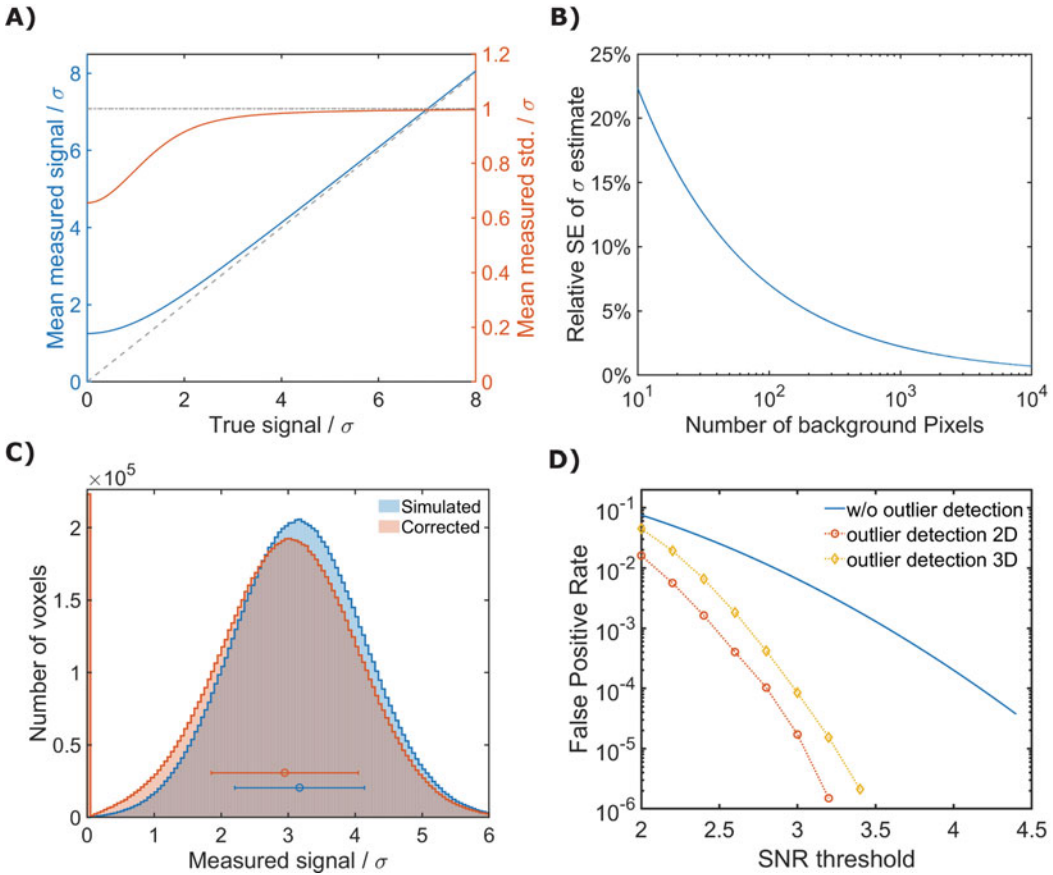


Fig. 1 (a) Mean measured signal and mean measured standard deviation (std.) of MR magnitude data. σ denotes the signal level independent standard deviation of the Gaussian noise in the underlying complex data. (b) The size of the used background region determines the uncertainty of the standard deviation estimate. The relative standard error (SE) shown here is defined as the standard deviation of the standard deviation estimate divided by σ itself. Note the logarithmic scale of the x-axis. (c) Histogram of simulated Rician data for true signal = 3σ (blue). The corrected data was computed using the tool described in Subheading 3.3 (red). The circles and bars indicate the mean and standard deviations of the two distributions. (d) Expected false positive rate (FPR) for different SNR thresholds with and without outlier detection. The FPR is defined as number of false positive voxels divided by number of true negative voxels. Without outlier detection background subtraction is performed by simple thresholding. The expected FPR can be computed by the steps in **Note 5**. Outlier detection is performed by removing all groups of less than three connected signal voxels after SNR based thresholding. In 2D 8-connectivity is used and in 3D 18-connectivity. Here simulated data is shown

2. A MATLAB / Octave tool for noise bias correction downloadable from github.com/LudgerS/MRInoiseBiasCorrection. This tool includes detailed documentation to facilitate adaptation for other software development environments or use with multichannel RF coil data.
3. A collection of data preparation specific MATLAB / Octave subfunctions found under github.com/LudgerS/19fMRIdataPreparation. It is provided together with a script

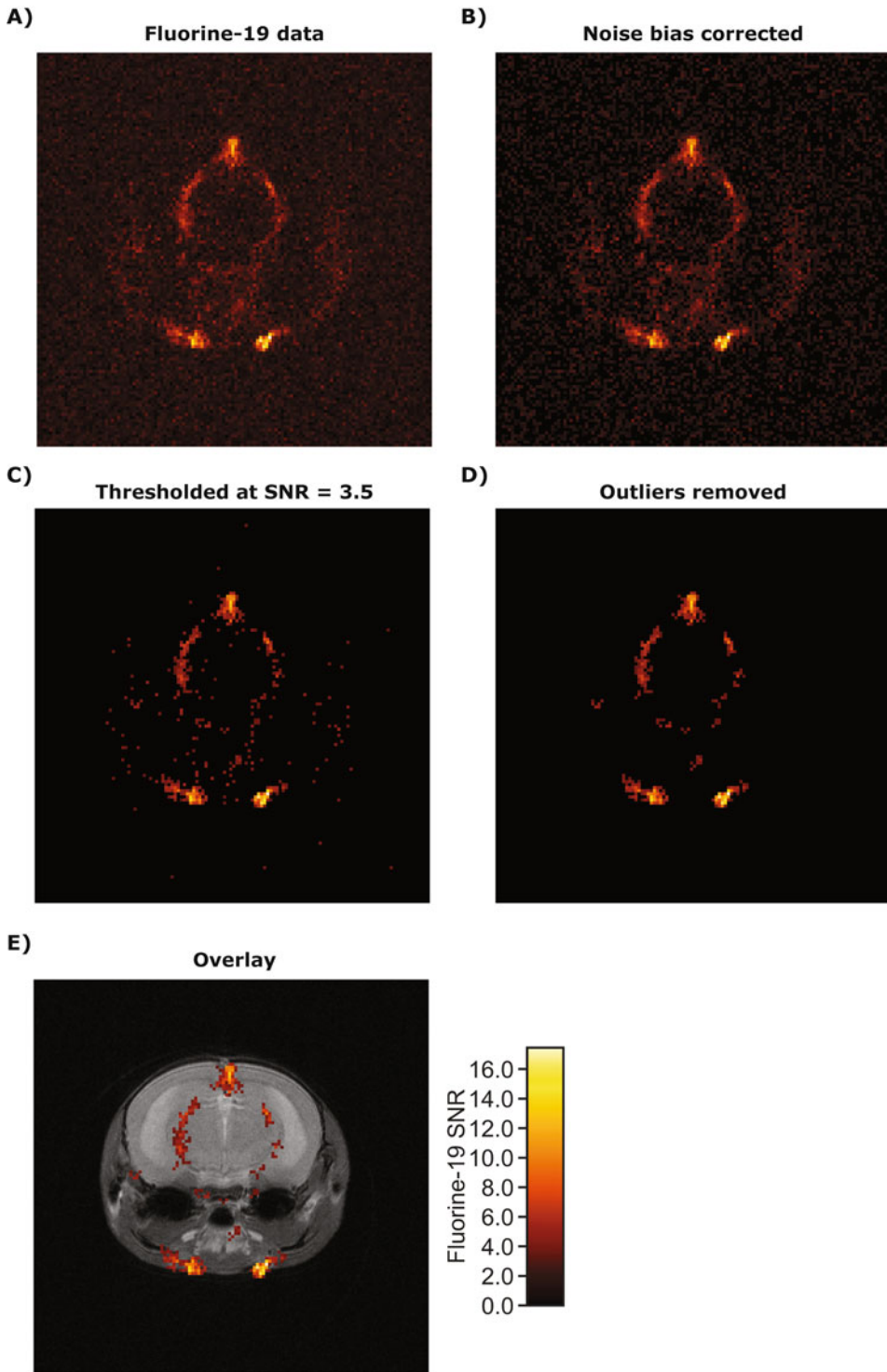


Fig. 2 Example analysis for fluorine-19 MRI data acquired in a mouse model of multiple sclerosis. Immune cells were labeled in situ with fluorine-19 nanoparticles such that fluorine signal shows sites of inflammation.

collecting all processing steps of this protocol as well as an example dataset allowing the reproduction of Fig. 2.

4. The MATLAB *Image Processing* toolbox is needed to run the function `fluorineOverlay.m` which creates proton/fluorine overlays and is part of the aforementioned `19fMRIdataPreparation` repository.
5. A tool for data import. This tool will depend on the data format:

DICOM data—MATLAB contains a built-in function to import dicom data: `dicomread`. For Octave, Python and Julia, dicom support is offered by supplemental packages ([octave.sourceforge.io/dicom/](https://sourceforge.io/dicom/), pydicom.github.io/, github.com/JuliaIO/DICOM.jl).

Bruker data—Bruker also offers a toolbox to import data into MATLAB directly. To obtain Bruker's *pvtools*, contact Bruker software support (mri-software-support@bruker.com).

3 Methods

3.1 Data Import and Scaling

These steps should be executed for both the fluorine MRI data and the noise scan data. In the following sections it is assumed that the fluorine data was stored in variable `imageData` and the noise scan in variable `noiseData`.

3.1.1 DICOM Data

1. To import dicom data execute the command

```
data = dicomread('filePath');
```

where `'filePath'` is a string containing the relative or absolute path of the `.dcm` file.

3.1.2 Bruker Data

For a Linux or MacOS system, substitute the file separator `\` by `/`.

1. Ensure that `pvtools` is on the search path:

```
addpath(genpath('\...\pvtools'))
```

where `'...\pvtools'` specifies the path of the `pvtools` folder.

Fig. 2 (continued) Acquisition parameters: 2D-RARE, ETL = 32, $[20 \times 20]$ mm² FOV, 128×128 matrix, 3.2 mm slice thickness. The noise level was determined from a corresponding noise scan. Details regarding the animal model and acquisition can be found in Starke et al. [12] under Setup 4. (a) Fluorine-19 MRI magnitude data. (b) Data after application of the Rician noise bias correction. (c) Thresholding was performed at SNR = 3.5. (d) Removal of groups of <3 connected pixels. (e) Overlay of the fluorine data on an anatomical image

2. Store the path to the scan you want to load in variable `scanFolder`.
3. Read the `visu_pars` parameter file:

```
visuPars = readBrukerParamFile([scanFolder, '\visu_pars']);
```

4. Load the Paravision reconstruction:

```
[data, ~] = readBruker2dseq([scanFolder, '\pdata\1\2dseq'],  
visuPars);
```

3.2 Correct Data Scaling

There are vendor-specific conventions regarding the averaging of multiple image acquisitions. In some cases, for example in Bruker data, the individual acquisitions are simply added instead of computing the mean. To ensure comparability between different scan times, determine the employed convention (*see Note 2*) and if necessary divide the data by the number of averages. This processing step will be required if a pure noise scan has been employed to determine the noise level.

1. Assuming `nAverages` is the number of averages acquired for the fluorine data and the noise scan was acquired with a single average, simply execute

```
imageData = imageData/nAverages;
```

3.3 Noise Level Estimation

The amount of random variation in MR magnitude images, as measured by the standard deviation, is signal level dependent (Fig. 1a). By convention, the noise level is reported as the asymptotic standard deviation at high SNR, which we denote by σ [13, 14]. This value is equal to the standard deviation of the Gaussian noise in the underlying complex data. While a larger number of noise level estimation methods has been proposed [13], two are of practical relevance for fluorine MRI: estimation based on a background region, which is known to be without signal, or estimation based on a dedicated pure noise scan. Although the former yields adequate results for most applications, we strongly recommend the use of a noise scan due to its independence of user input and improved accuracy. The uncertainty of the standard deviation estimate depends on the number of samples n . For $n > 10$, we can approximate the standard error as $SE_{\sigma} \approx \frac{\sigma}{\sqrt{2n}}$ (Fig. 1b) [15].

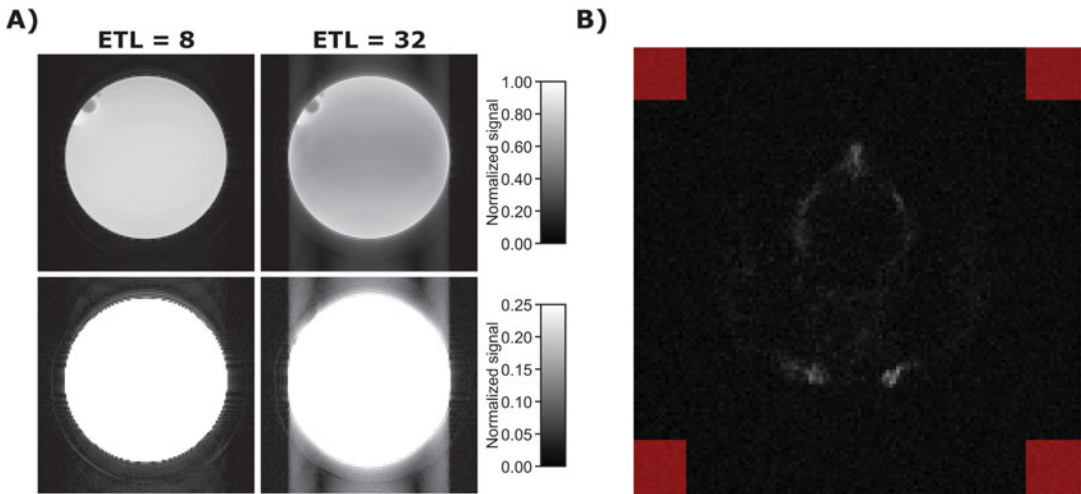


Fig. 3 (a) Point spread function artifacts can corrupt background regions without being visually obvious. In this water phantom example, the RARE image with echo train length 8 (ETL = 8, top left) does not show artifacts. However rescaling (bottom left) shows that a noise level estimate based on a background region would lead to erroneous values if the region is not chosen just in the corners of the image. The example on the right (ETL = 32) shows an exaggerated case where the effect is obvious in the image itself. (b) The red regions show a background mask that would yield reliable values for the example shown in Fig. 2

3.3.1 Estimation Based on a Background Region

The largest possible background region free of major signal artifacts should be used to improve accuracy (also *see* **Note 3**). However, point spread function artifacts can corrupt image regions even distal from signal features without being necessarily visible (Fig. 3a). We recommend using quadratic regions in all four corners of the image.

1. Create a mask of the background region for logical indexing

```
backgroundMask = createBackgroundMask(size(imageData),
cornerSize);
```

The argument `size(imageData)` determines the dimensions of the mask and `cornerSize` sets the size of the four corners (Fig. 3b).

2. The noise level can then be computed as

```
sigma = std(imageData(backgroundMask))/0.6551;
```

where the factor 0.6551 accounts for the ratio between the background standard deviation and σ (Fig. 1a) [13]. It needs to be adjusted for data from multichannel RF coils (*see* **Note 4**).

3.3.2 Estimation Based on a Noise Scan

In this case the complete noise image constitutes the background region. It is not necessary to acquire the noise scan with multiple averages, as the dependence of the noise level on the number of

averages is well understood mathematically (*see Note 5* for further comments on noise scan acquisition).

1. If the noise scan was acquired with a single average and the number of averages of the image data is given by `nAverages`, the noise level can be computed as

```
sigma = std(noiseData(:))/(0.6551*sqrt(nAverages));
```

3.4 Bias Correction

Rician noise leads to a systematic upward bias in MR magnitude images at low SNR. Signal intensities need to be multiplied with a correction factor <1 to achieve correct signal estimation on average (Fig. 1a, c). While the expected measured signal $E[M]$ for a given true signal level A and noise level σ is known, there is no established inverse function $A(E[M], \sigma)$ [9, 10, 14]. The provided toolbox thus uses a lookup table to compute the correction factor [9]. Signal intensities below the mean background level are set to zero (Fig. 1c). Alternatively, the correction scheme of Koay and Basser could be used [14]. *See Note 6* for a comment on averaging over regions of interest (ROIs).

1. After determining the noise level, the corrected image (Fig. 2b) can simply be computed as

```
correctedImage = correctNoiseBias(imageData, sigma, 1);
```

where the last argument specifies the number of receive elements in the RF coil (*see Note 1*).

3.5 Background Subtraction

In fluorine MRI, background subtraction commonly involves thresholding by removing all voxels below a certain SNR (Fig. 2c). However, the threshold necessary to prevent the occurrence of false positives could compromise sensitivity, especially for larger datasets such as those from 3D MR data (Fig. 1d, *see Note 7* for an analytical formula). Based on the assumption that all relevant fluorine features should at least comprise a minimum of three connected voxels, we recommend an additional outlier correction that excludes isolated voxels (Fig. 2d). To determine features with a minimum of three connected voxels, we recommend to use 8-connected neighborhoods (connection via edges or corners) for 2D images and 18-connected neighborhoods (connection via faces or edges but not corners) for 3D images. This additional processing step drastically reduces the expected FPR. In the 2D case, for example, thresholding at $\text{SNR} = 2.74$ with outlier detection reduces the FPR to the same level as thresholding at $\text{SNR} = 4$ without outlier detection (Fig. 1d).

1. Copy the bias corrected image to a new variable

```
thresholdedImage = correctedImage;
```

2. Set all voxels below the chosen threshold `snrThreshold` to zero

```
thresholdedImage(thresholdedImage < snrThreshold*sigma) = 0;
```

3. Then apply the provided function for outlier removal on the thresholded image:

```
cleanedImage = removeIsolatedVoxels(thresholdedImage, 3);
```

The second argument specifies the minimum size of preserved features.

3.6 Proton/ Fluorine-19 Overlays

In order to locate the source of fluorine signal within an in vivo context, it will need to be coregistered with an anatomical image. While it might be appealing to overlay transparent fluorine images to avoid concealing of anatomical detail, we advise against this practice for quantification purposes, as manipulation of the data will result in ambiguous readings. If no external standard is used to determine fluorine concentrations, it is best practice to show the fluorine signal in terms of the SNR as this also conveys a standard means of information. For the provided function the anatomical image should be acquired with the same field of view as the fluorine MR image.

1. An overlay of fluorine on proton MR images (Fig. 2e) can be achieved simply by calling

```
fluorineOverlay(anatomyData, cleanedImage, colorMap, resizeF, fileName)
```

`anatomyData` is the anatomical image, `colorMap` specifies the fluorine color map in the conventional MATLAB format, `resizeF` is a factor by which the image size is increased to ensure faithful display and `fileName` the file name of the saved image.

2. To complement the fluorine image with a corresponding color bar (Fig. 2e) execute

```
plotFluorineColorBar(colorMap, fileNameCB, max(cleanedImage(:))/sigma, 'Fluorine-19 SNR')
```

`fileNameCB` states the path and file name of the saved color bar image. The third argument specifies the range of the color map. Dividing the maximum of the fluorine image by σ establishes the SNR scaling. The fourth argument determines the label of the color bar.

4 Notes

1. In the case of multichannel RF coil data and sum-of-squares reconstruction, the MR signal follows a noncentral chi distribution instead of a Rician distribution. This case is described thoroughly in Constantinides et al. [16]. The bias correction toolbox used in this protocol also handles the more general case. It should be noted that the bias effects become more pronounced with increasing number of receive elements.
2. To determine the employed convention of averaging, acquire high SNR phantom data with varying numbers of averages while keeping all other parameters fixed. Import the data as described in Subheading 3.1. If all scans show the same signal magnitude, the scaling step described in Subheading 3.1 should be omitted. However, if, for example, doubling the number of averages also doubles the signal amplitude, rescaling should be performed.
3. Be aware that some vendors include heavy filtering and/or background masking into the automated reconstruction pipeline. This has a major impact on the background noise or sets background values to zero and hence renders estimation based on a background region not suitable for noise assessment.
4. Assuming a sum-of-squares reconstruction, the factor 0.6551 should be replaced by 0.6824, 0.6953, 0.7014 or 0.7043 for 2, 4, 8 or 16 receive element RF coil data [16].
5. A pure noise scan is acquired by setting the excitation flip angle and reference power to zero so that no excitation occurs and pure noise is acquired. The receiver gain needs to be set identical to all other scans for which the noise level should be determined. The number of averages can be reduced to one, as the resulting change in noise level is easily compensated. Instead of setting the reference power to zero in preparation of the noise scan, the output of the RF power supply can be disconnected following the system adjustments.
6. In a background region without true signal, the Rice distribution reduces to the Rayleigh distribution [10], which is equivalent to a chi-squared distribution. Thus the expected FPR can be computed via the cumulative distribution function of the chi-squared distribution. First calculate the expected signal [17] for the chosen SNR cutoff `snrThreshold`

$$\text{thresholdExpectation} = \text{sigma} \times \sqrt{\pi/2} \times \text{laguerreL}(1/2, 0, -\text{snrThreshold}^2 / (2 \times \text{sigma}^2)).$$
Then calculate the expected FPR as
$$\text{FPR} = 1 - \text{gammainc}((\text{thresholdExpectation}/\text{sigma})^2 / 2, 1, \text{'lower'}).$$
This formula is valid for simple thresholding without outlier removal and data from single receive element RF coils.

7. Often fluorine-MRI studies involve calculating signal-to-noise ratios for specific regions of interest (ROIs). Averaging over all pixels in the region should be performed before applying the noise bias correction. This is particularly relevant for phantom studies where the number of voxels in the assumed to be homogeneous ROI is large enough to obtain accurate values even for regions with $\text{SNR} < 2$. See **Note 8** for a comment on the uncertainty of SNR estimates.
8. SNR is conventionally reported as the bias corrected signal level divided by the asymptotic standard deviation σ [14]. The uncertainty of the SNR estimate is determined by the signal level, the size of the ROI and the uncertainty of the noise level. A convenient formula for the standard error is $\frac{\text{SE}_{\text{SNR}}}{\text{SNR}} \approx \sqrt{\frac{1}{m \cdot \text{SNR}^2} + \frac{1}{2n}}$, where m denotes the number of voxels in the ROI and n the number of voxels in the background region used to estimate the noise level. At high SNRs the uncertainty of the SNR estimate is dominated by the uncertainty of the noise level estimate.

Derivation: The standard errors of the mean ROI signal \bar{S} and noise level σ are $\text{SE}_{\bar{S}} = \frac{\sigma}{\sqrt{m}}$ and $\text{SE}_{\sigma} \approx \frac{\sigma}{\sqrt{2n}}$ [15]. Assuming independence of \bar{S} and σ , the variance formula [18] for the standard error of $\frac{\bar{S}}{\sigma}$ yields

$$\text{SE}_{\text{SNR}} \approx \sqrt{\left(\frac{1}{\sigma} \cdot \frac{\sigma}{\sqrt{m}}\right)^2 + \left(\frac{\bar{S}}{\sigma^2} \cdot \frac{\sigma}{\sqrt{2n}}\right)^2} = \sqrt{\frac{1}{m} + \frac{\text{SNR}^2}{2n}}.$$

Acknowledgment

This study was funded, in part (Sonia Waiczies, Thoralf Niendorf; DFG WA2804, DFG PO1869), by the Deutsche Forschungsgemeinschaft and, in part (T.N. and S.W.), by the German Research Foundation (Gefördert durch die Deutsche Forschungsgemeinschaft (DFG), Projektnummer 394046635, SFB 1365, RENOPROTECTION. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project number 394046635, SFB 1365, RENOPROTECTION).

This chapter is based upon work from COST Action PARENCH IMA, supported by European Cooperation in Science and Technology (COST). COST (www.cost.eu) is a funding agency for research and innovation networks. COST Actions help connect research initiatives across Europe and enable scientists to enrich their ideas by sharing them with their peers. This boosts their research, career, and innovation.

PARENCHIMA (renalMRI.org) is a community-driven Action in the COST program of the European Union, which unites more than 200 experts in renal MRI from 30 countries with the aim to improve the reproducibility and standardization of renal MRI biomarkers.

References

1. Ruiz-Cabello J, Barnett BP, Bottomley PA, Bulte JW (2011) Fluorine (19F) MRS and MRI in biomedicine. *NMR Biomed* 24 (2):114–129
2. Ahrens ET, Helfer BM, O'Hanlon CF, Schirda C (2014) Clinical cell therapy imaging using a perfluorocarbon tracer and fluorine-19 MRI. *Magn Reson Med* 72(6):1696–1701
3. Prinz C, Delgado PR, Eigentler TW, Starke L, Niendorf T, Waiczies S (2019) Toward (19)F magnetic resonance thermometry: spin-lattice and spin-spin-relaxation times and temperature dependence of fluorinated drugs at 9.4 T. *Magma* 32(1):51–61
4. Waiczies S, Rosenberg JT, Kuehne A et al (2019) Fluorine-19 MRI at 21.1 T: enhanced spin-lattice relaxation of perfluoro-15-crown-5-ether and sensitivity as demonstrated in ex vivo murine neuroinflammation. *Magma* 32(1):37–49
5. Kuethe DO, Caprihan A, Fukushima E, Waggoner RA (1998) Imaging lungs using inert fluorinated gases. *Magn Reson Med* 39 (1):85–88
6. Hu LZ, Chen JJ, Yang XX et al (2014) Assessing intrarenal nonperfusion and vascular leakage in acute kidney injury with multinuclear H-1/F-19 MRI and perfluorocarbon nanoparticles. *Magn Reson Med* 71(6):2186–2196
7. Hitchens TK, Ye Q, Eytan DF, Janjic JM, Ahrens ET, Ho C (2011) 19F MRI detection of acute allograft rejection with in vivo perfluorocarbon labeling of immune cells. *Magn Reson Med* 65(4):1144–1153
8. Moore JK, Chen J, Pan H, Gaut JP, Jain S, Wickline SA (2018) Quantification of vascular damage in acute kidney injury with fluorine magnetic resonance imaging and spectroscopy. *Magn Reson Med* 79(6):3144–3153
9. Henkelman RM (1985) Measurement of signal intensities in the presence of noise in MR images. *Med Phys* 12(2):232–233
10. Gudbjartsson H, Patz S (1995) The Rician distribution of noisy MRI data. *Magn Reson Med* 34(6):910–914
11. van Heeswijk RB, Pilloud Y, Flögel U, Schwitter J, Stuber M (2012) Fluorine-19 magnetic resonance angiography of the mouse. *PLoS One* 7(7):e42236
12. Starke L, Pohlmann A, Prinz C, Niendorf T, Waiczies S (2019) Performance of compressed sensing for fluorine-19 magnetic resonance imaging at low signal-to-noise ratio conditions. *Magn Reson Med* 84(2):592–608
13. National Electrical Manufacturers Association (2001) Determination of signal-to-noise ratio (SNR) in diagnostic magnetic resonance imaging. NEMA standards publication MS 1-2001. NEMA, Arlington, VA
14. Koay CG, Basser PJ (2006) Analytically exact correction scheme for signal extraction from noisy magnitude MR signals. *J Magn Reson* 179(2):317–322
15. Ahn S, Fessler A (2003) Standard errors of mean, variance, and standard deviation estimators. EECs Department, The University of Michigan, Ann Arbor, MI. <http://web.eecs.umich.edu/fessler/papers/files/tr/stderr.pdf>
16. Constantinides CD, Atalar E, McVeigh ER (1997) Signal-to-noise measurements in magnitude images from NMR phased arrays. *Magn Reson Med* 38(5):852–857
17. Aja-Fernández S, Vegas-Sánchez-Ferrero G (2016) Statistical analysis of noise in MRI. Springer International Publishing, Cham
18. Ku HH (1966) Notes on the use of propagation of error formulas. *J Res Natl Bur Stand* 70 (4):263–273

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

