Chapter 36

Analysis Protocols for MRI Mapping of the Blood Oxygenation–Sensitive Parameters T_2* and T_2 in the Kidney

João S. Periquito, Ludger Starke, Carlota M. Santos, Andreia C. Freitas, Nuno Loução, Pablo García Polo, Rita G. Nunes, Thoralf Niendorf, and Andreas Pohlmann

Abstract

Renal hypoxia is generally accepted as a key pathophysiologic event in acute kidney injury of various origins and has also been suggested to play a role in the development of chronic kidney disease. Here we describe step-by-step data analysis protocols for MRI monitoring of renal oxygenation in rodents via the deoxyhemoglobin concentration sensitive MR parameters T_2* and T_2—a contrast mechanism known as the blood oxygenation level dependent (BOLD) effect.

This chapter describes how to use the analysis tools provided by vendors of animal and clinical MR systems, as well as how to develop an analysis software. Aspects covered are: data quality checks, data exclusion, model fitting, fitting algorithm, starting values, effects of multiecho imaging, and result validation.

This chapter is based upon work from the PARENCHIMA COST Action, 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 experimental protocol chapter is complemented by two separate chapters describing the basic concept and data analysis.

Key words Magnetic resonance imaging (MRI), Kidney, Mice, Rats, T_2, T_2*, BOLD

1 Introduction

The parametric mapping of the transverse relaxation times T_2* and T_2 (or relaxation rates $R_2 = 1/T_2$* and $R_2 = 1/T_2$) has the potential to yield inferences regarding renal oxygenation, since both parameters are sensitive to blood oxygenation [1]. The underlying mechanism rests on the inherent difference in the magnetic properties of oxygenated hemoglobin (diamagnetic) vs. deoxygenated hemoglobin (paramagnetic) [2]. The presence of deoxyhemoglobin in a voxel decreases both relaxation times, T_2* and T_2, which are the time constants governing the exponential signal decays due to spin

Andreas Pohlmann and Thoralf Niendorf (eds.), *Preclinical MRI of the Kidney: Methods and Protocols*, Methods in Molecular Biology, vol. 2216, https://doi.org/10.1007/978-1-0716-0978-1_36,

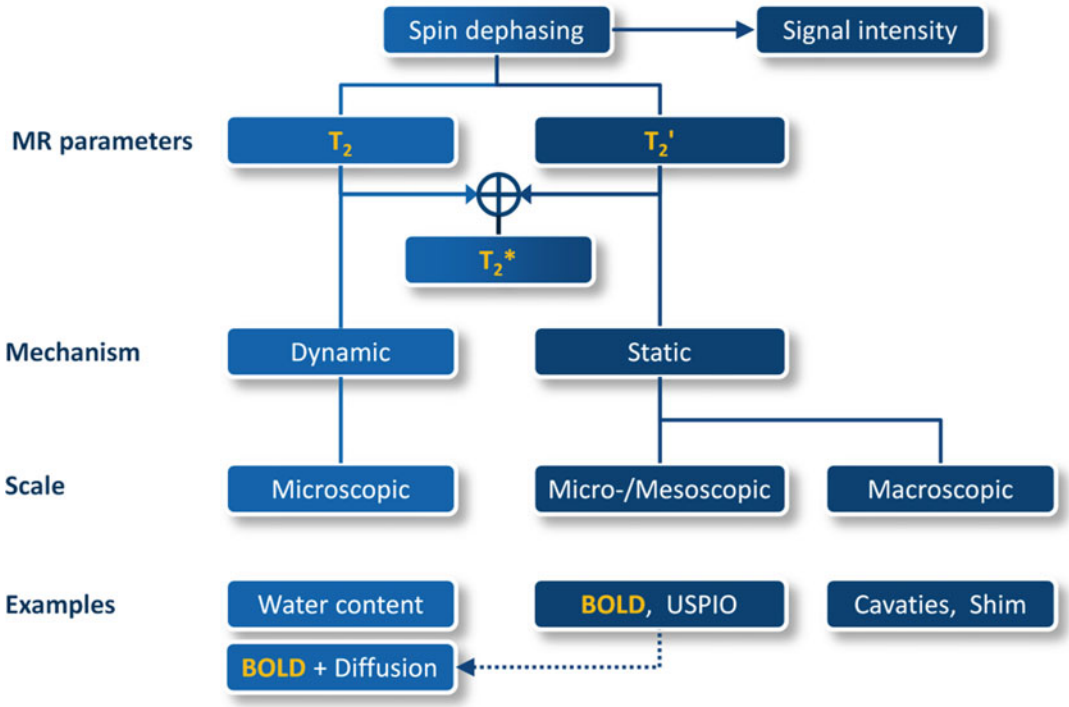

Fig. 1 Spin dephasing due to dynamic (irreversible; represented by the parameter T_2) and static (reversible; represented by the parameter T_2') sources can be quantified by calculation of T_2^* and T_2 from series of MRI images with different echo times using $1/T_2^* = 1/T_2 + 1/T_2'$. The possible sources for this dephasing range from the microscopic to the macroscopic scale and their level dictates the observed attenuation in the signal intensity. Blood oxygenation affects primarily T_2^* (often referred to as Blood Oxygenation Level Dependent, BOLD), but to a lesser extend also T_2 via diffusion effects in the proximity of blood vessels. USPIO: Ultra-small Superparamagnetic Particles of Iron Oxide, which can be used as intravascular contrast agents

dephasing in gradient-echo (GRE) and spin-echo (SE) MR measurements, respectively.

Possible sources for this dephasing are magnetic field inhomogeneities ranging from the microscopic to the macroscopic scale, which can be classified with respect to the echo time into microscopic dynamic spin–spin interactions—T_2, that is, typically ranging a time span between 1 and 100 ms (Fig. 1) and static mesoscopic interactions—T_2'. In fact, T_2^* includes the dynamic (irreversible) dephasing effects described by T_2 plus the additional effects that are due to static (reversible) dephasing effects described by T_2'. Blood oxygenation affects primarily T_2^* (often referred to as Blood Oxygenation Level Dependent, BOLD), but to a lesser extend also T_2, via water diffusion effects in the proximity of blood vessels due to local magnetic field gradients [1].

Calculation of T_2^* and T_2 requires a series of MR images with different echo times. Repeated measurements with a single echo

sequence using either a GRE or SE method and variable echo times (TE) would provide the most accurate T_2* and T_2 values but would require very long acquisition times [2]. In order to address this shortcoming fast multiecho MRI methods are commonly used for in vivo studies: multi-gradient-echo (MGE) for T_2* and multi-spin-echo (also called "MSME") for T_2. They provide relaxation times that slightly underestimate T_2* and overestimate T_2 (*see* **Note 1**).

The core of $T_2^{(*)}$ mapping consists of fitting the exponential model curve $S(\mathrm{TE}) = S_0 \exp.(-\mathrm{TE}/T_2^{(*)})$ to the signal intensities of each image pixel at increasing TE. For the calculation of such parameter maps from a series of $T_2^{(*)}$-weighted images software tools and plugins are provided by the MR system vendors but they may lack some features instrumental for precise $T_2^{(*)}$ mapping. For example, some important preprocessing steps may not be available, such as eliminating the first echo of T_2 data, removing echoes acquired at high TEs that may have too low signal-to-noise ratio (SNR) or applying Rician noise bias correction. Also, details about the used processing and curve fitting algorithms may not be available. It is also a limitation that the algorithms are fixed and cannot be modified for specific purposes. Therefore, depending on application, it may be advantageous to develop a dedicated analysis software program in-house. The analysis software is developed assuming 2D data was acquired, for 3D data (multiple slices) data set are separated into individual slices.

The pixel-wise signal fitting with a monoexponential model is known to introduce T_2 bias, and so for studies that require an extremely rigorous T_2 value (error $< 1\%$) a dictionary-based method has been suggested. In dictionary-matching methodologies, T_2 maps are calculated by matching the T_2 signal decay to precomputed Echo Modulation Curve (EMC), therefore accounting for all echo pathways. The use of dictionary-based methods has been suggested to improve T_2 accuracy, accounting for the stimulated echoes and effective B_1^+ field [3]. The use of dictionary-based methods is still fairly novel and its implementation and use is not covered in this chapter.

This chapter will describe both how to use the vendor's analysis tools and how to develop your own analysis software conceptually. An example of a MATLAB script described in this chapter together with a data sample (*T2analysis.m* and *data.mat*) can be downloaded from https://github.com/JoaoPeriquito/T2-s-Mapping.

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

This chapter is part of the book Pohlmann A, Niendorf T (eds) (2020) Preclinical MRI of the Kidney—Methods and Protocols. Springer, New York.

2 Materials

2.1 Software Requirements

To calculate the parametric maps using existing tools, a software such as the following is required:

2.1.1 Essential Tools

1. MR system software *ParaVision* (version 5 or higher; Bruker Biospin, Ettlingen, Germany).
2. MR system software *Syngo* (versions MR B17 or higher; Siemens Healthineers, Erlangen, Germany). The optional toolbox *MapIt* is not necessary.
3. MR system software *Ready View* (General Electric Healthcare, Milwaukee, USA).
4. MR system software from Philips (Philips Medical Systems, Best, Netherlands).

To develop custom software for calculating the parametric maps one of the following, or other equivalent, software development environments (SDE) is required:

5. *MATLAB*® including the *Curve Fitting* toolbox (The MathWorks, Natick, Massachusetts, USA; www.mathworks.com/products/MATLAB).
6. *Python* (https://www.python.org/).
7. *Octave* (http://www.gnu.org/software/octave).
8. A MATLAB tool for Rician noise bias correction can be downloaded from https://github.com/LudgerS/MRInoiseBiasCorrection. It is also compatible with Octave and includes detailed documentation to facilitate adaptation for, for example, Python.

2.1.2 Optional Tools

An image processing software for checking data quality, such as ImageJ (free Java-based image processing program developed at the National Institutes of Health and the University of Wisconsin; https://imagej.net/). We recommend and describe here the use of Fiji (https://fiji.sc/ [4]), which is ImageJ packaged with a wide range of plugins included.

2.2 Source Data: Format Requirements and Quality Check

Before the analysis it is highly recommended to check the image quality. This check should include SNR measurements, particularly for the images with longest echo times, and the assessment of geometric image distortions, motion artifacts, or susceptibility artifacts.

The steps in this section can be performed either on the scanner console using the MR vendors system viewing software, or offline using a software such as *Fiji* (recommended as a practical tool).

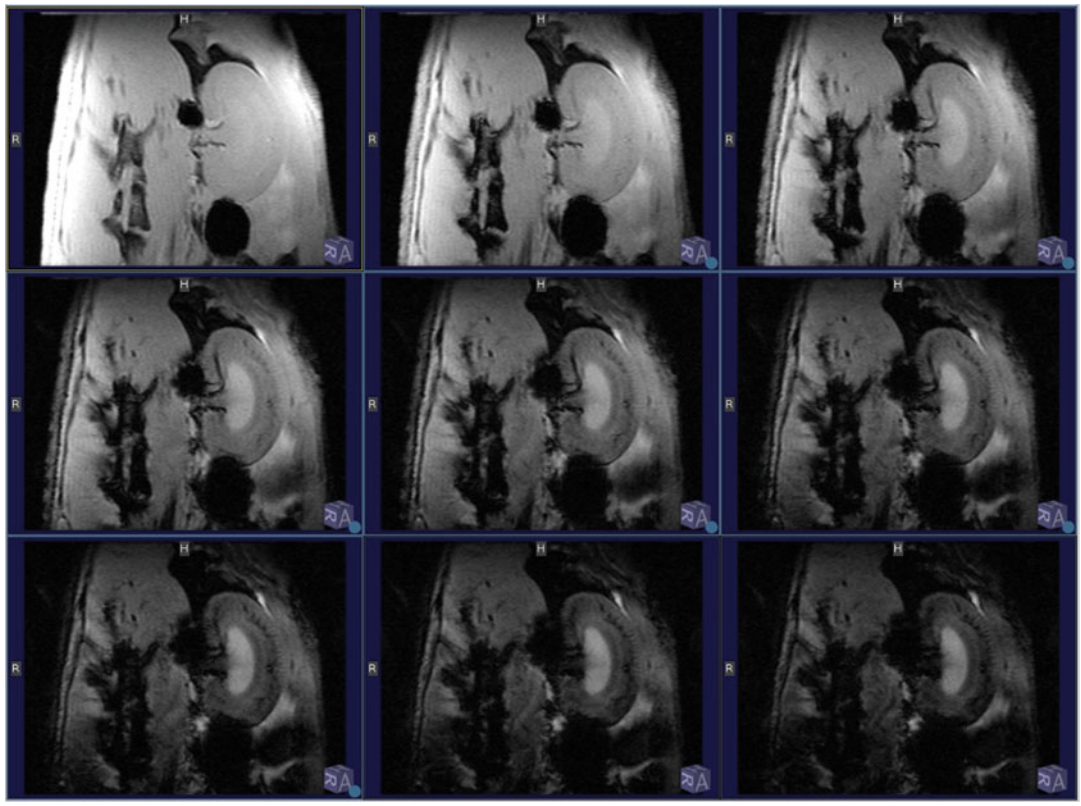

Fig. 2 Series of nine T_2*-weighted images of a healthy rat kidney acquired with a 2D MGE sequence at 9.4 T. From left to right and top to bottom, images correspond to TE = 2.14, 4.28, 6.42 ms (top row), TE = 8.56, 10.70, 12.84 ms (middle row), and TE = 14.98, 17.12, 19.26 ms (bottom row)

2.2.1 Input Requirements

As already mentioned, in order to be able to calculate $T_2^{(*)}$-maps, multiple images acquired with different echo times are needed. As an example, please refer to Fig. 2 and Fig. 3, where a series of T_2*- and T_2-weighted images suitable for mapping is presented (*see* **Note 2**). Access to specific acquisition parameters is also necessary, namely the TE of each image and in some cases the image intensity scaling parameters (offset, slope) used when storing the image data.

2.2.2 Open/Import Images

When using the scanner console:

1. Open the $T_2^{(*)}$-weighted image series in the image viewer.

When using *Fiji* for DICOM images:

2. Import the DICOM image series using the DICOM import plugin.

When using *Fiji* for images in Bruker format:

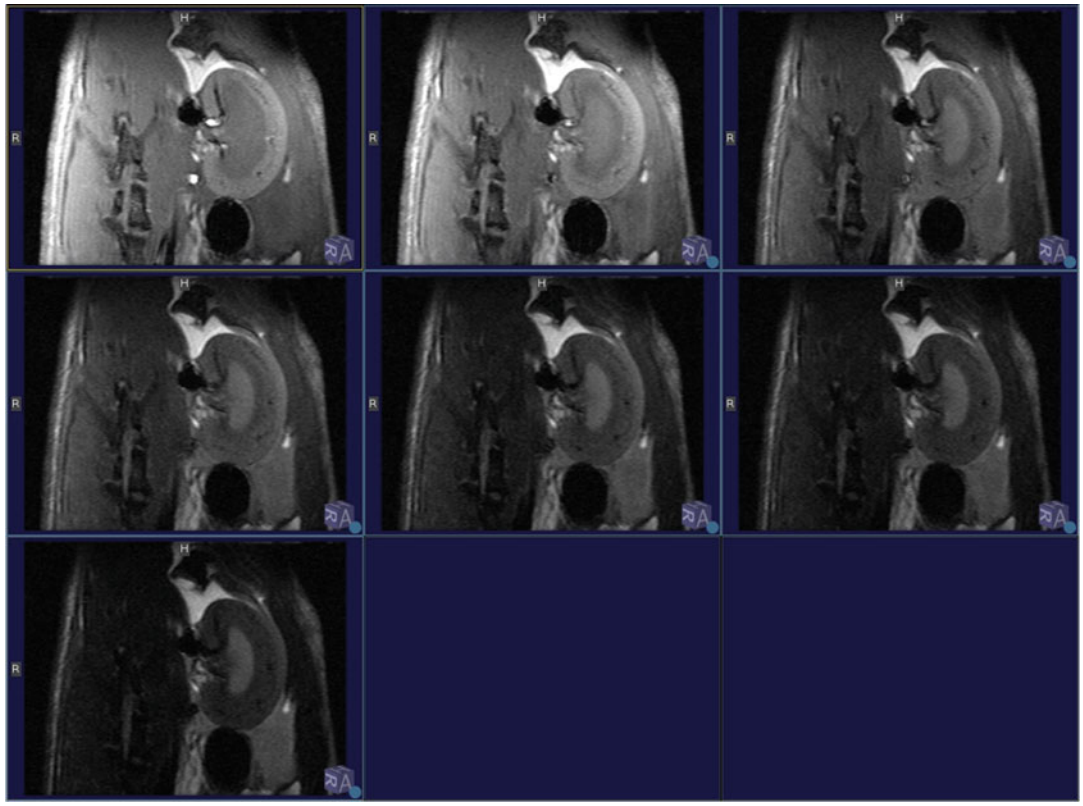

Fig. 3 Series of seven T2-weighted images of a healthy rat kidney acquired with a 2D MSME sequence at 9.4 T. Images correspond to TE = 10.0, 20.0, 30.0 ms (top row), TE = 40.0, 50.0, 60.0 ms (middle row), and TE = 70.0 ms (bottom row)

3. Browse to the data directory for the scan containing the $T_2^{(*)}$-weighted image series: /opt/PV6.0.1/data/[user_name]/[session_name]/[scan_number]/.
4. Open the "*method*" file of the scan in a text editor or by dragging it to the *Fiji* window.
5. Get the following parameters: number of slices, matrix size (e.g., 128 × 128) image type (e.g., little-endian) and byte order (e.g., 16-bit unsigned).
6. Load the image series using the RAW import function (*Import > RAW*) and providing the above parameters. The MR data file is called 2dseq and located in the subfolder "pdata/nr" (number of reconstruction) of the data directory.

2.2.3 Motion Artifacts Check

Abdominal imaging comes with complex bulk physiological motion due to the interplay of respiratory and bowel movement. If respiratory triggering of the data acquisition was used (recommended) there should only be minor motion artifacts. Check

images for artifacts from motion (nontriggered acquisition) or possible residual motion (triggered acquisition).

In the image viewer or *Fiji* go to the central slice and scroll rapidly from the first to the last echo image.

1. Check for motion between the echoes. If motion is noticeable then information close to structure boundaries may not be reliable and image registration should be used for motion correction (registration is a separate topic that is not explained in this chapter).

2.2.4 Susceptibility Artifacts Check on T_2*-Weighted Images

Kidney regions adjacent to bowels or in close proximity to skin/fat/muscle boundaries or air cavities are particularly challenging to T_2* imaging and prone to loss of anatomical integrity due to geometric distortions and signal loss created by susceptibility artifacts induced by the air-filled bowels, cavities and tissue interfaces surrounding the kidneys.

1. Scroll to the images acquired at the later echoes, where susceptibility artifacts are more severe.
2. Check areas affected by susceptibility artifacts, that is, areas in the kidney with unusual and severe hypointensities (typically in the form of spherical dark "shadows"; Fig. 4).
3. Make a record of areas affected by such artifacts (notes, screenshots). They must be excluded from the analysis or at least be considered during interpretation of the result.

2.2.5 Signal-to-Noise Ratio Check

Before the analysis it is highly recommended to define an SNR acceptance threshold (*see* **Note 3**).

1. Draw a region-of-interest (ROI) over the inner medulla in the first echo image (in *Fiji* use the "Freehand Selection" tool) (*see* **Note 4**).

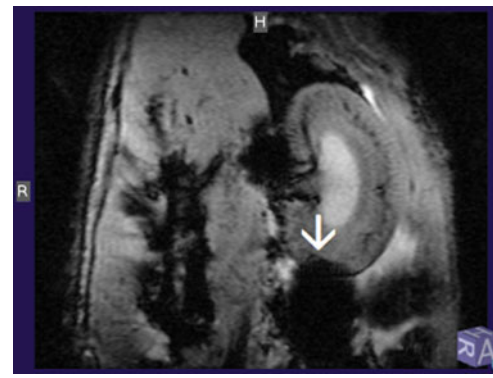

Fig. 4 T_2*-weighted image of a rat kidney that shows a susceptibility artifact at the caudal end (white arrow); image acquired with TE = 10.70 ms at B_0 = 9.4 T

2. Measure the mean signal intensity of the ROI (in *Fiji*: *Analyze* > *Set Measurements*; select mean and standard deviation; *Analyze* > *Measure*).
3. Draw an ROI over a region in the background that contains only noise without any signal or artifacts (*see* **Note 5**).
4. Measure the standard deviation of the signal within this background ROI.
5. Divide this value by 0.655 to obtain the noise standard deviation. This is necessary because the background noise in MR magnitude images differs from noise in signal regions (*see* **Note 6** if you use a multichannel RF array).
6. Calculate the SNR by dividing the mean signal of the inner medulla ROI by the noise standard deviation.
7. If the SNR is significantly below your predefined acceptance threshold, this indicates that there might be a technical problem or that the animal is not positioned correctly relative to the receiver RF coils. For more information on adequate SNR thresholds refer to **Note 7**.

3 Methods

Tools for calculating $T_2^{(*)}$-maps are provided by most MRI vendors, but, as mentioned in the introduction, we recommend creating your own analysis tool to have the maximum level of control over the processing steps and freedom to adapt them to your specific needs. Hence, this section will describe both how to use vendor's analysis tools and how to develop your own analysis software.

In both cases, mapping of T_2 and T_2*, it may be required to exclude some data from the analysis, such as data points with insufficient SNR in the tail of the exponential decay. In the case of T_2-mapping from data acquired with a MSME method, it is also recommended to discard the first echo (*see* **Note 8**).

3.1 Data Exclusion and Model Fitting (Siemens Syngo)

When using the Siemens scanner console this is currently not possible, but one can manually select the echo images that should be included when computing the fit. Using this feature, one can exclude entire images at high TEs. We recommend using a cut-off for echo images with low SNR. Please exclude all echo images with SNR < 2.6 in the ROI of interest (*see* **Note 7**). When performing T_2-mapping, the first echo should also be excluded from the data analysis (*see* **Note 8**).

1. Load image series in the *Viewing Task Card*.
2. Using the *ROI* tool to draw an ROI on the image with the largest TE over the region of interest, and register the mean signal.

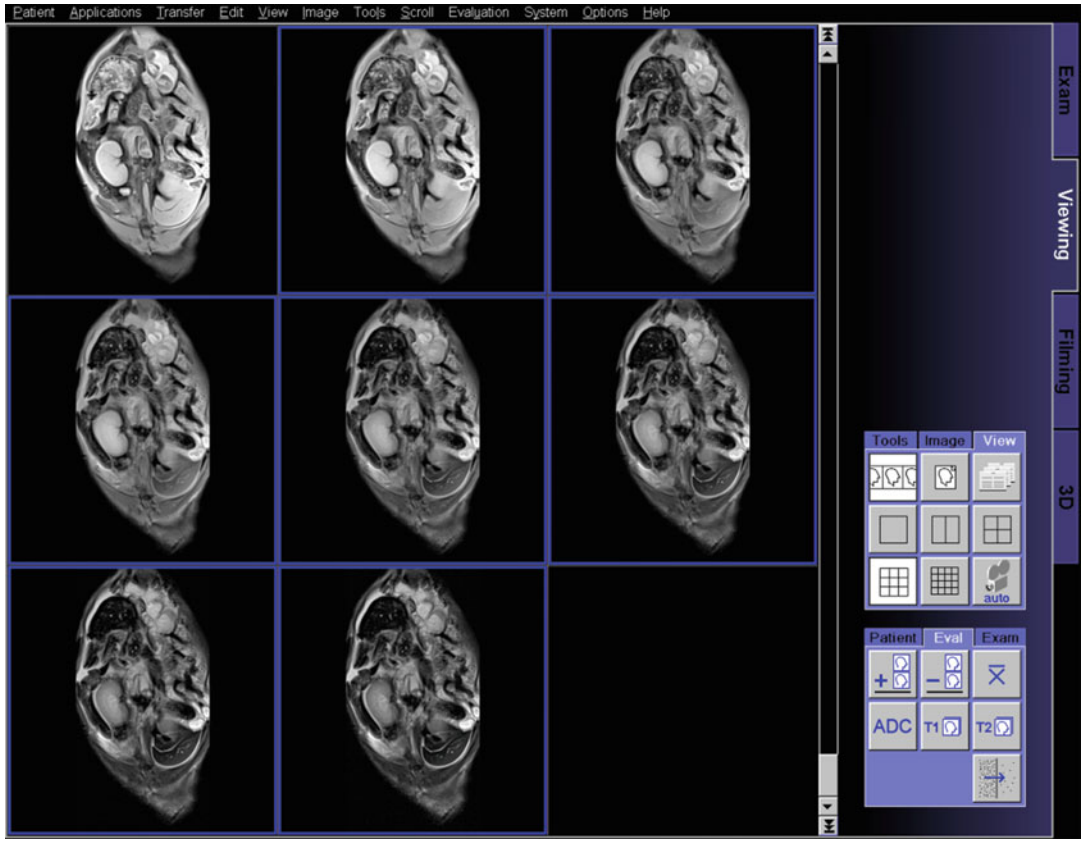

Fig. 5 Selection of echo images for $T_2^{(*)}$-mapping of a rat kidney in Siemens *Syngo*. Here T_2-mapping was performed, for which the first echo image must be excluded. Then start the $T_2^{(*)}$-map calculation with the *T2* button in the *Eval* tab in the *Control Area* panel on the bottom right

3. On the same image, use the *ROI* tool draw an ROI on the background, and register the standard deviation of the signal. Divide this value by 0.655 to obtain the noise standard deviation (*see* **Note 6** if you use a multichannel RF array).
4. Calculate the SNR by dividing the mean signal ROI by the noise standard deviation. If the SNR is below 2.6 this image should be excluded on step "8" (*see* **Note 7**).
5. Select a 3 × 3 or 4 × 4 layout in the *View* tab of the *Control Area* (right panel) in order to see all acquired echo images.
6. Select all echo images for T_2*-mapping or all echo images except the first for T_2-mapping (Fig. 5).
7. In the *Control Area* choose *Eval* > T_2 > *OK*. Two new images will appear in the viewer, the $T_2^{(*)}$-map and the S_0-map.
8. Exclude all echo images that have a SNR < 2.6 (*see* **Note 7**).

Fig. 6 T_2-map of a rat kidney calculated using the *T2-analysis* tool of Siemens *Syngo* from data acquired with a clinical 3 T scanner

9. In the *Control Area* choose *Eval* > T_2 > *OK*. A $T_2{}^{(*)}$-map will appear in the viewer. Use this new map for further analysis/quantification (Fig. 6).

3.2 Model Fitting (General Electric $T_2{}^{(*)}$ Map Ready View)

On a GE system a similar tool is used for mapping of T_2 and T_2*. The *Ready View T2 Map* protocol post processes data sets acquired using the *Cartigram* (T_2 Map) application. The *T2 Map* layout is comprised of four viewports: upper-left Map viewport: source image, upper-right viewport: curve displaying signal intensity (vertical axis) and the echo number (horizontal axis). A data point for each echo is plotted when an ROI is deposited on any of the three images. Lower left viewport: *T2 Map* Preset-1 parametric image. Lower right viewport: *T2 Map* Preset-2 parametric map.

Use these steps to post-process data sets acquired using the *Cartigram* application. The $T_2{}^{(*)}$ relaxation time color map is

coded to capture $T_2^{(*)}$ values from the TE range of the acquired images. Blue and green reflect the longer $T_2^{(*)}$ values, yellow the intermediate $T_2^{(*)}$ values, and red and orange the shorter $T_2^{(*)}$ values. The functional $T_2^{(*)}$ map units are ms.

1. Open *ReadyView* and start the *T2map* protocol.
2. Adjust the *W/L* (window width and level) and magnification—to adjust the *W/L*, middle-click and drag over the image. To adjust the magnification factor, place the cursor over the red *DFOV* and middle-click and drag right to left.
3. Locate the desired images to view—press the Up and Down arrows to move through the images to locate the image with the area of interest. Or click and drag the red slice Location annotation. Press the Right and Left arrows to select the desired echo or click and drag the red echo annotation.
4. Open the *T2 Map* Settings screen.
5. Adjust the settings on the *T2 Map Settings* screen. Typically, the Preset Color Level sliders for both the $T_2^{(*)}$ map and parametric image are within a 10 and 90 ms range and the Threshold is 20. Typically, the Confidence Level is 0.05. A default smoothing kernel of 2 is recommended.
6. Click Compute.
7. If necessary, use the *Clip Min & Max* values. By default (25–75).
8. In the measurements panel, select the ROI icon and draw the desired region of interest. An annotation with the area, mean value and standard deviation of the ROI will appear.

3.3 Model Fitting (Philips)

1. When acquiring a $T_2^{(*)}$ (mTSE/MGE) dataset, select the *Post-proc* tab and choose the option $T2^{(*)}$ or $R2^{(*)}$ ($1/T_2^{(*)}$) under the field *Calculated images* (Fig. 7).
2. Insert the maximum expected $T_2^{(*)}$ or its reciprocal ($1/T_2^{(*)}$) under the field *T2 clip value* or its reciprocal $R2^{(*)}$ *clip value* to specify the maximum calculated $T_2^{(*)}/R_2^{(*)}$ value for the SE sequence. Larger $T_2^{(*)}/R_2^{(*)}$ values are clipped to this value.
3. By default, T_2/T_2* reconstruction includes the first echo and performs a MLE fitting of the signal curve. Research customers can modify these parameters under the *Scan Control Parameters* panel select *Reconstruction* and choose the option *Include first echo for (SE) T2,R2 calc (yes/no)*, and reconstruction method (*MLE* [5, 6] or Least squares (*RLSQ*)) for $T_2^{(*)}$, $R_2^{(*)}$. If available we recommend to use *Include first echo = no* and the default *MLE* calculation method (*see* **Note 9**)—Warning: changing these options affects every scan until reset of the system (Fig. 8). After the acquisition, drag and drop the images in the viewer.

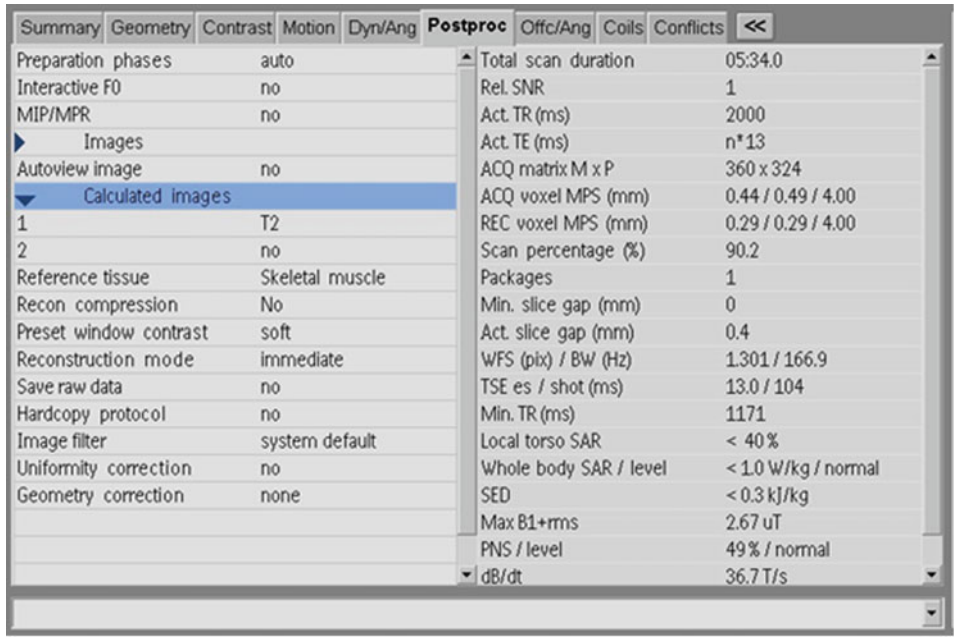

Fig. 7 Philips acquisition software *Postproc* tab. Here T_2-mapping can be performed by choosing the option *T2* or *R2* ($1/T_2$) under the field *Calculated images*

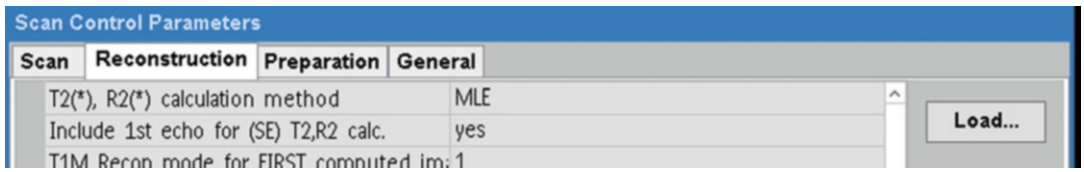

Fig. 8 Philips acquisition software *Scan Control Parameters* panel—only accessible for research customers. Here under the *Reconstruction* tab select *no* on the option *Include first echo for (SE) T2/R2 calc.* to exclude the first echo image on T_2 mapping and select *MLE* for *T2(*), R2(*) calculation method* (*see* **Note 9**)

3.4 Data Preparation and Model Fitting (Custom Program)

The following custom program for $T_2^{(*)}$ relaxometry contains code examples following MATLAB syntax. These were tested to be compatible with Octave and should be easily reproducible in Python or an equivalent SDE.

3.4.1 Data Import and Exclusion of First TE (Only for T_2)

1. Import your scan with the images obtained for different TEs.

   Bruker data: in MATLAB use the import function provided by Bruker's *pvtools*—to obtain Bruker's *pvtools* send an e-mail to Bruker mri-software-support@bruker.com.

   DICOM data: in MATLAB use the dicomread function—https://mathworks.com/help/images/ref/dicomread.html; in Python use dcmread from the pydicom package—https://pydicom.github.io/). As input you need to provide a string containing the full path of the DICOM file.

2. As previously explained, when performing T_2-mapping using a multiecho sequence, the first echo must be excluded from data analysis (*see* **Note 8**). For example, that could mean that for a series of T_2 weighted data acquired at TE = 10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 70.0 ms one would only be able to fit the model curve to the TE = 20.0, 30.0, 40.0, 50.0, 60.0 ms data points: store your data in a different variable excluding the first TE (`imgData_forAnalysis = imgDataT2w(:,:,2:end); TE_forAnalysis = TE(2:end))`.

3.4.2 Rician Noise Bias Correction

MR signal intensities are overestimated in low SNR magnitude images. This bias is due to the Rician distribution of noisy MR magnitude data and can be corrected in post-processing. We can understand the processing step as multiplication with a signal-level dependent correction factor, which approaches 1 as higher SNRs are reached. To incorporate Rician noise bias correction into our custom program, we first need to determine the noise level and then utilize a MATLAB/Octave tool provided for this book under https://github.com/LudgerS/MRInoiseBiasCorrection:

1. Add the downloaded noise correction tool to the search path.

   ```
   Addpath(genpath('...\MRInoiseBiasCOrrection)).
   ```

2. Compute unbiased data by executing the following:

   ```
   imgData_corrected=correctNoiseBias(imgData_forA-
   nalysis,sigma,1)
   ```

   If you work with data from a multichannel RF array, change the last argument of the function to the number of receive elements.

3.4.3 Fitting Model

The most common method is to iteratively fit the model of the $T_2^{(*)}$ decay to the signal intensity (SI) data of each pixel using the following equation:

$$S(\mathrm{TE}) = S_0 . \exp\left(-\frac{\mathrm{TE}}{T_2^{(*)}}\right) \quad (1)$$

where S_0 is a scaling factor that includes many parameters such as the proton density together with the signal gain of the system.

3.4.4 Starting Values

An important step of the curve fitting process is the choice of suitable starting values for each parameter that will be determined by the fitting algorithm (NB: different starting values may lead to different results!). Here we describe how to derive starting values from the SI of each pixel.

1. Step through all pixels in the images using for loops for the pixel coordinates x and y.

2. Store the SI of that pixel at all TEs in a vector (in MATLAB: `SI_vector = imgData_forFitting(xPix, yPix,:)`).
3. As starting value for the parameter S_0 use the first and largest SI (`startVal_S0 = SI_vector(1)`).

For estimating a starting value for $T_2^{(*)}$, one can exploit the fact that at TE = $T_2^{(*)}$ the SI has fallen to 37% of its value at TE = 0. Estimate the starting value by determining the TE whose SI is closest to 37% of the largest SI value:

4. Normalize the SI vector to a maximum value of 1 (divide by the largest SI value).
5. Subtract 0.63 from the SI vector.
6. Get the absolute values of the SI vector (resulting in only positive values).
7. Find the index of the smallest value in the vector (in MATLAB: `[min_Value,min_Index] = min(SI_norm_abs_37pct)`).
8. Get the respective echo from your echoes using the index from the last step (`startVal_T2=TE_forFitting(min_Index)`).

3.4.5 Fitting Algorithm

Least squares algorithms, such as the Levenberg–Marquardt [7] and Trust-region methods [8], are the most commonly used curve fitting algorithms for $T_2^{(*)}$-mapping. They work by minimizing a cost function, which describes the deviation of the fitted curve from the corresponding data points. With starting values near the optimal solution they quickly converge, but with starting values far away from the solution, the Levenberg–Marquardt algorithm will slow down significantly. Also, there is the risk that it may converge to a local minimum (rather than the global minimum) and hence produce an erroneous result.

In contrast, the Trust-region method (a further development of the Levenberg–Marquardt algorithm) will quickly converge, even with suboptimal starting values, and it will always find the global minimum. However, it does require the definition of lower and upper limits for the parameters to be fitted. Hence, for $T_2^{(*)}$-mapping, where such limits can easily be defined, the Trust-region method is well suited (*see* **Note 10**)

1. Create for loops for pixel coordinates *x* and *y* to step through all pixels of the image.
2. Store the SI of that pixel at all TEs in a vector (in MATLAB: `SI_vector = imgData_forFitting(xPix, yPix,:)`).
3. Define the model for the function using Equation1 (in MATLAB: `T2model = fittype('a*exp.(-x*b)', 'independent', 'x', 'dependent', 'y')`).

4. Choose the function for the fitting algorithm, that is, Trust-region or, if not available, then Levenberg–Marquardt (in MATLAB: `opts.Algorithm = 'Trust-Region'`).
5. Provide the starting values for S_0 and $T_2^{(*)}$ (in MATLAB, e.g., `opts.StartPoint = [startVal_S0 startVal_T2]`).
6. Define lower and upper limits for the fit parameters. Use for T_2* [0.1 150], for T_2 [1 1000], and for S_0 the possible range of SI in the image data (this depends on the system; for 16-bit integer it may be [1 65536]). In MATLAB, for example, `opts.Lower = [1 0.1]; opts.Upper = [65,536 150].`
7. Execute the curve fitting for each pixel data (in MATLAB: `[fitresult, gof] = fit(TE_forFitting, SI_vector, T2model, opts);`).
8. Save the fit result for each variable in parameter maps: `mapS0(xPix, yPix) = fitresult.a; mapT2(xPix, yPix) = fitresult.b; rsquare(xPix, yPix) = gof.rsquare.`

3.4.6 Visual Display

1. Display the parameter map, which is a matrix with floating point numbers, as an image (in MATLAB: `imagesc(mapT2);`).
2. Remove axis labels and ensure that the axes are scaled such that the aspect ratio of the image is identical to that of the acquired FOV (For the case of square pixels in MATLAB: `axis off; axis equal;`).
3. Select the color map and display a color bar (in MATLAB: colormap(jet(256)); `colorbar;`) (*see* **Note 11**).
4. Set the display range for the color coding, for example, for T_2* [0 25]; and for T_2 [0 80]; (in MATLAB: `caxis([0 25]);`). An example is shown in Fig. 9.

3.5 Quantification

Quantitative values can be obtained from the parameter maps using manually drawn ROI for the different morphological regions (cortex, outer medulla, and inner medulla). Because manual ROI drawing can introduce unwanted additional variability or bias, we recommend the use of semiautomated methods, such as the concentric objects technique [9] or the morphology-based ROI placement technique [10]. For further details and step-by-step protocols for these two techniques please refer to the chapter by Riazy L et al. "Subsegmentation of the Kidney in Experimental MR Images Using Morphology-Based Regions-of-Interest or Multiple-Layer Concentric Objects."

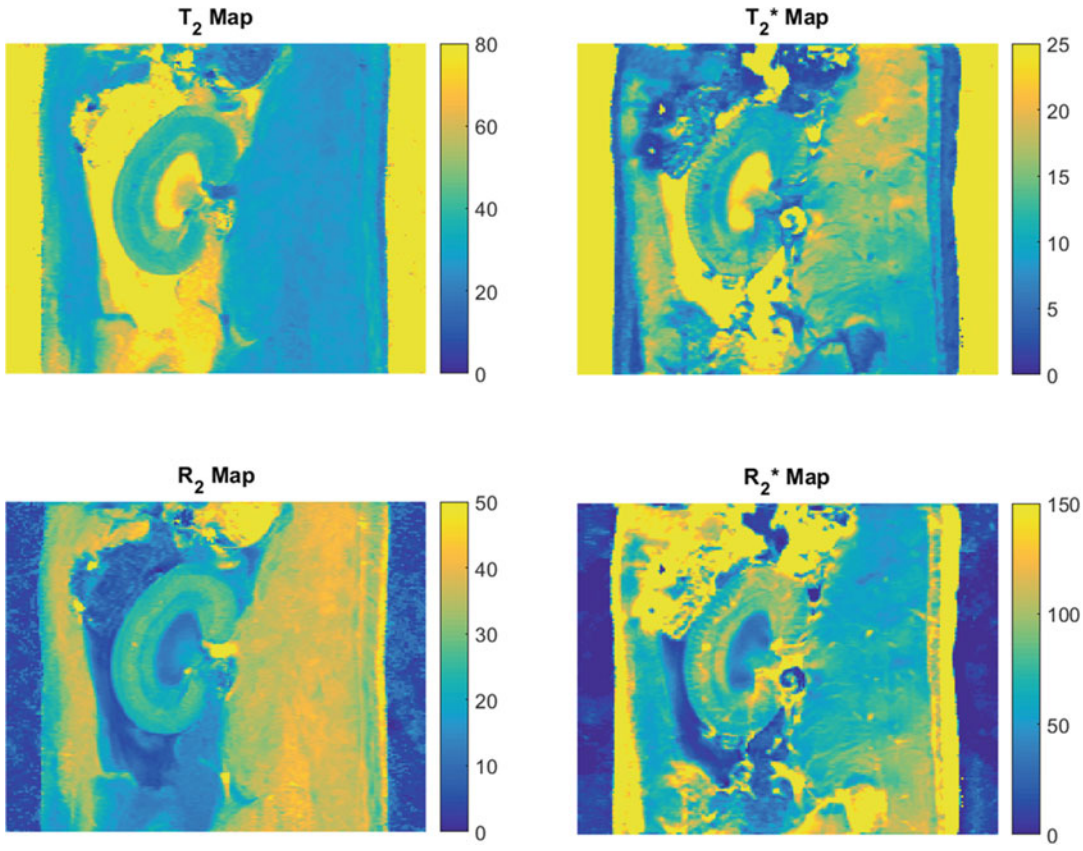

Fig. 9 $T_2^{(*)}$ in ms and $R_2^{(*)}$ in ms^{-1} maps of an healthy rat kidney calculated using a custom program developed in MATLAB (displayed using colormap *parula*) from data acquired with a preclinical 9.4 T scanner

3.6 Methods for Result Validation

3.6.1 Evaluation of Analysis Errors and Variability Using Synthetic Data

A simple approach to validate the data analysis procedure is to generate synthetic image data and then evaluate how close the results produced by the analysis are to the known true value. A series of artificial images could be created (using an in-house software program) that mimic a multiecho MR experiment. Each image could represent a chosen TE for which the signal intensity in a virtual phantom (could simply be a circle in the center of the image) has been calculated using the known exponential equation that describes the $T_2^{(*)}$ relaxation. Alternatively, more sophisticated simulation approaches as previously described in [3] could be used, accounting for the timings and characteristics of the used RF pulses so as to evaluate the effect of using the simpler monoexponential decay model. Having generated the images, Rician noise is added. This synthetic data could then be analyzed, evaluating both accuracy (how close the resulting $T_2^{(*)}$ is to the true $T_2^{(*)}$) and precision (evaluating dispersion over multiple instances with the same level of SNR).

Table 1
Typical values of T_2 and T_2^* for each specific renal tissue for a healthy rat at 9.4 T [11] and 3 T [12]

Map	Inner medulla	Outer medulla	Cortex
9.4 T [8]			
R_2 [1 s^{-1}]	11.0–14.6	20.0–23.2	23.0–27.3
R_2* [1 s^{-1}]	32.1–47.2	73.9–101.1	64.8–93.6
3.0 T [9]			
R_2 [1 s^{-1}]	5.5–7.0	9.6–12.5	13.8–16.3
R_2* [1 s^{-1}]	12.8–19.8	28.3–30.3	29.5–37.5

3.6.2 Comparison with Reference Values from the Literature

The obtained values can be compared with those reported for healthy animals in the literature. Table 1 provides ranges of T_2* and T_2 for different magnetic field strengths based on current literature.

4 Notes

1. With increasing echo number additional diffusion weighting is added due to the repeated magnetic field gradients used for spatial encoding (NB: this effect is much more pronounced in small animal systems, which use much stronger gradients than clinical systems). A greater bias is introduced by the superposition of stimulated echoes in multi-spin-echo imaging, which leads to significant overestimation of T_2. It also demands the exclusion of the first echo (pure spin echo) from the analysis, because its magnitude is usually much smaller than that of the following echoes, which are combinations of spin-echoes and stimulated echoes. Although these biases depend on the acquisition parameters, they are fixed and reproducible for each protocol and hence acceptable in studies focusing on relative differences/changes, where precision is far more important than accuracy. If needed, more accurate T_2 values can be obtained from multiecho data using sophisticated post-processing, such as dictionary-matching methodologies [3].
2. Make sure your data is ordered from the lowest to the highest TE as in Figs. 2 and 3.
3. When establishing the MR technique, define an SNR acceptance threshold. The aim is to have at least three (preferably five or more) number of echoes with an SNR > 5. This threshold will depend on the expected $T_2^{(*)}$ values, which in turn depend on parameters like the magnetic field strength, shim quality, and tissue properties (pathology). Example: for a rat imaged at 9.4 T using a 4-element rat heart array receiver surface RF coil

together with a volume resonator for excitation in combination with interventions leading to strong hypoxia an SNR > 60 was needed for the image with the lowest TE. Always draw the ROI over the same region of the kidney, that is, cortex, outer medulla, or inner medulla (we use the inner medulla, which is the brightest region of the kidney). If reaching a sufficiently high SNR is a problem, consider creating ROIs and performing the $T_2^{(*)}$-fitting on the average values of these ROIs in order to boost the SNR (SNR scales with the square root of the number of pixels). Another option is to modify the acquisition protocol, reducing the attained spatial resolution to gain SNR.

4. As a good practice you should always save your ROIs for later reference. After creating an ROI click on Analyze > Tools > ROI Manager > Add. Then rename your ROI to a meaningful name (using the button "Rename on the ROI Manager).
5. When you create an ROI over a region with no signal make sure no artifacts are present. Click on Image > Adjust > Brightness and Contrast and drag the scroll bar "Minimum" to the left.
6. Sum-of-squares reconstruction of multichannel data changes the distribution of noisy MR magnitude data. Substitute the factor 0.655 by 0.682 or 0.695 for two channel and four channel data, respectively [13].
7. The Rician distribution of MR magnitude images introduces a signal level-dependent positive bias. The relative error introduced by this bias increases with decreasing SNR. For single receive element coils, an error of 10% is reached at SNR = 2.6. For two-channel or four-channel coils, the corresponding threshold is at SNR = 4.2 or 6.4, respectively, assuming sum-of-squares reconstruction. Under pathophysiological conditions $T_2^{(*)}$ can become significantly shorter and images acquired at high TEs may have a rather low SNR or may even reach noise level. This could result in a poor model curve fit and overestimate the calculated $T_2^{(*)}$. Therefore, data points at high TEs below a predefined SNR threshold should be excluded from the data analysis. Data exclusion can be done either during the acquisition (by not acquiring some echoes) or performed prior to data analysis. The latter is advantageous because the SNR cutoff to exclude data from the fitting can also be adjusted for each pixel individually.
8. When a multiecho sequence is used for T_2-mapping, the first echo must be excluded from the data analysis. The reason is that the first echo is a pure spin echo, unlike all following echoes, which are superpositions of spin-echoes and stimulated echoes. As a result, the first echo has significantly smaller amplitude than the second echo and fitting the model curve to all data points would produce a poor curve fit and an inaccurate T_2 value [10].

9. MLE option refers to "Maximum-likelihood estimator" that takes into account the Rician noise. The noise level is measured as part of the scan. This approach should be robust against noise floor estimation bias and therefore there is no need to exclude later echoes from the analysis [5, 6].
10. An approach to increase the speed of the curve fitting is using a fast mono exponential fit [14]. The MATLAB implementation of the following fit can be downloaded from https://github.com/JoaoPeriquito/T2-s-Mapping under the name *fastExpoFit.m*. The result of a simulation comparing the recommended fast mono exponential fit and the trust-region can be found at the same location under the name *fastExpoFit_Simulation.png*.
11. Pseudocolor representations can be extremely useful for analyzing $T_2^{(*)}$-maps, since they generally enhance the perception of differences within the value range of the parameter. But one needs to be careful when choosing a color map. Parameter maps displayed as images in pseudocolor can potentially be misleading. Small differences in the underlying values may be artificially emphasised by a change in color hue, or a significant parameter difference that can easily be seen in a gray-scale image may be flattened or hidden if there is little change of hue or brightness over a certain range of the color-scale. It is recommended to always use the same color map and scale to improve comparability.

Acknowledgments

The authors wish to thank Dr. Torben Schneider and Dr. Anabea Solana for the contribute of the $T_2^{(*)}$ mapping protocol for Philips and GE scanner, respectively. Portuguese Foundation for Science and Technology (FCT—UID/EEA/50009/2019), and POR Lisboa 2020 (LISBOA-01-0145-FEDER-029686).
This work was funded in part (Thoralf Niendorf, Andreas Pohlmann, Joao Periquito) by the German Research Foundation (Gefoerdert durch die Deutsche Forschungsgemeinschaft (DFG), Projektnummer 394046635, SFB 1365, RENOPROTECTION. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project number 394046635, SFB 1365, R ENOPROTECTION).
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. Niendorf T, Pohlmann A et al (2015) How bold is blood oxygenation level-dependent (BOLD) magnetic resonance imaging of the kidney? Opportunities, challenges and future directions. Acta Physiol (Oxf) 213(1):19–38. https://doi.org/10.1111/apha.12393
2. Pohlmann A et al (2014) Detailing the relation between renal T2* and renal tissue pO2 using an integrated approach of parametric magnetic resonance imaging and invasive physiological measurements. Investig Radiol 49:547–560
3. Ben-Eliezer N, Sodickson DK, Block KT (2015) Rapid and accurate T2 mapping from multi-spin-echo data using Bloch-simulation-based reconstruction. Magn Reson Med 73:809–817
4. Schindelin J, Arganda-Carreras I, Frise E et al (2012) Fiji: an open-source platform for biological-image analysis. Nat Methods 9 (7):676–682. https://doi.org/10.1038/nmeth.2019
5. Senegas J, Knopp T, Dahnke H (2007) Dealing with spatially varying noise in T2* mapping with SENSE. Proceedings of the International Society of Magnetic Resonance in Medicine, Berlin
6. Hedström E et al (2017) Automatic T2* determination for quantification of iron load in heart and liver: a comparison between automatic inline maximum likelihood estimate and the truncation and offset methods. Clin Physiol Funct Imaging 37(3):299–304
7. Moré JJ (1978) The Levenberg-Marquardt algorithm: implementation and theory. Numerical analysis. Springer, Berlin, pp 105–116
8. Byrd RH, Schnabel RB, Shultz GA (1987) A trust region algorithm for nonlinearly constrained optimization. SIAM J Numer Anal 24(5):1152–1170
9. Piskunowicz M et al (2015) A new technique with high reproducibility to estimate renal oxygenation using BOLD-MRI in chronic kidney disease. Magn Reson Imaging 33(3):253–261
10. Pohlmann A, Cantow K, Huelnhagenm T, Grosenick D, dos Santos Periquito J et al (2017) Experimental MRI monitoring of renal blood volume fraction variations en route to renal magnetic resonance oximetry. Tomography 3(4):188–200
11. Pohlmann A et al (2013) High temporal resolution parametric MRI monitoring of the initial ischemia/reperfusion phase in experimental acute kidney injury. PLoS One 8(2):e57411
12. Thacker J, Zhang JL, Franklin T, Prasad P (2017) BOLD quantified renal pO2 is sensitive to pharmacological challenges in rats. Magn Reson Med 78(1):297–302
13. Constantinides CD, Atalar E, McVeigh ER (1997) Signal-to-noise measurements in magnitude images from NMR phased arrays. Magn Reson Imaging 38(5):852–857
14. Pei M et al (2015) Algorithm for fast monoexponential fitting based on auto-regression on linear operations (ARLO) of data. Magn Reson Med 73:843–850