Quantitative Analysis of Renal Perfusion by Arterial Spin Labeling

Kai-Hsiang Chuang, F. Kober, Min-Chi Ku

Chapter 39

Quantitative Analysis of Renal Perfusion by Arterial Spin Labeling

Kai-Hsiang Chuang, Frank Kober, and Min-Chi Ku

Abstract

The signal intensity differences measured by an arterial-spin-labelling (ASL) magnetic resonance imaging (MRI) experiment are proportional to the local perfusion, which can be quantified with kinetic modeling. Here we present a step-by-step tutorial for the data post-processing needed to calculate an ASL perfusion map. The process of developing an analysis software is described with the essential program code, which involves nonlinear fitting a tracer kinetic model to the ASL data. Key parameters for the quantification are the arterial transit time (ATT), which is the time the labeled blood takes to flow from the labeling area to the tissue, and the tissue T_1. As ATT varies with vasculature, physiology, anesthesia and pathology, it is recommended to measure it using multiple delay times. The tutorial explains how to analyze ASL data with multiple delay times and a T_1 map for quantification.

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), Kidney, Rodent, ASL, Blood flow, Perfusion

1 Introduction

The signal intensity differences measured by an arterial-spin-labelling (ASL) magnetic resonance imaging (MRI) experiment are proportional to the local perfusion, which can be quantified with kinetic modeling. To calculate the perfusion map, an additional T_1 relaxation map should be obtained from inversion recovery data acquired with the same image acquisition method (eg, Spin-Echo EPI or RARE) and geometry setting.

Creation of perfusion maps involves nonlinear fitting of a tracer kinetic model to the ASL data. Depending on the type of ASL sequence—pulsed or (pseudo)continuous—the model is slightly different. Two key parameters in the quantification are the arterial transit time (ATT), which is the time the labeled blood taken to

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_39,

flow from the labeling area to the tissue, and the tissue T_1. As ATT varies with vasculature, physiology, anesthesia, and pathology, it is recommended to measure it using multiple delay times (i.e., multiple TI in pulsed ASL, or multiple postlabeling delays in continuous ASL). In the kidney, the cortex and medulla have different T_1. T_1 could also change depending on the pathological conditions. Therefore acquiring a T_1 map is desirable. This chapter focuses on how to analyze ASL data with multiple delay times and a T_1 map for quantification.

This analysis protocol chapter is complemented by two separate chapters describing the basic concept and experimental procedure, 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

The method described in this chapter provides a detailed description for a solution in MATLAB, but can be adopted to other platforms:

1. A programming environment capable of applying fitting models, such as Python, Octave, R or MATLAB®.
2. A software to convert the data format from DICOM or vendor proprietary format (eg, Bruker 2dseq) to the NifTi format, such as dcm2nii (for DICOM; https://people.cas.sc.edu/rorden/mricron/dcm2nii.html), or Bru2Nii (for Bruker 2dseq; https://github.com/neurolabusc/Bru2Nii).
3. An image viewing software such as Fiji (www.fiji.sc) or MRIcro (https://www.mccauslandcenter.sc.edu/crnl/mricro).
4. A software for motion correction, such as SPM, FSL, or AFNI.

2.2 Source Data: Format Requirements

2.2.1 Input Requirements

To be able to calculate perfusion maps, ASL data with interleaved control and label images and multiple delay times, and a T_1 mapping acquired with inversion recovery are needed. For more details on the basic concept of ASL please refer to the chapter by Ku M-C et al. "Noninvasive Renal Perfusion Measurement Using Arterial Spin Labeling (ASL) MRI: Basic Concept." The data acquisition is described in a step-by-step manner in the chapter by Chuang K-H et al. "Renal Blood Flow Using Arterial Spin Labeling (ASL) MRI: Experimental Protocol and Principles."

2.2.2 Intensity Scaling of Multiple Delay ASL Data

If the ASL data of multiple delay times was acquired by separate scans (eg, one delay time for one scan) but not all within the same scan, there could be a difference in the intensity scaling among separate scans due to changes in receiver gain and/or internal scaling factor during image reconstruction. Such a difference

could lead to artifactual intensity among delay times leading to bias in the model fitting and inaccurate perfusion quantification.

On Bruker MRI systems, the default intensity scaling is set to maximize the dynamic range of the output format. Therefore scans with lower signal would be magnified. The intensity scaling method in the "Reconstruction" class should be changed to "User Scaling" with a scaling factor of 1 for all the scans.

To ensure the same intensity scaling of the ASL data, the receiver gain may be extracted from the header information ("RG" variable in the "acqp" file) and used to rescale the data.

2.2.3 Format Conversion

Convert all data into 4D NifTi format. Consider magnifying the voxel size by 10 in the image header if motion correction tools designed for humans (eg, FSL or SPM) will be used (*see* **Note 1**).

3 Methods

3.1 Quality Control/ Data Exclusion

During the ASL image acquisition, movement of the body or kidney itself can occur. Artifacts may also arise from ASL labeling pulses. In order to construct accurate perfusion maps, it is important to ensure that the scan series don't show severe movement, spikes, banding artifacts, or sudden changes in SNR. Scans with poor quality or large movement should be discarded.

1. Open each dataset by a NifTi image viewer.
2. Adjust the window/level (or reduce the maximum intensity range) to visualize the image better.
3. Scroll through each time frame to visually check whether there are sudden changes in intensity or movement.

3.2 Motion Correction

ASL perfusion imaging, due to its subtraction between label and control images, is very sensitive to movement. Despite respiratory trigger, slight movement between scans could still be present. Motion artifact maybe reduced by rigid-body motion correction tools in many software packages. Since most motion correction algorithms (regardless correlation or square error based cost function) rely on intensity changes for detecting movement, the intensity changes between control and label images and between scans of different delay times (particularly for FAIR ASL) could lead to pseudo motion. Besides, the T_1 mapping data should also be coregistered to the ASL data. Therefore, it is recommended to realign images to the same reference target, (e.g., the M_0 image or an averaged image). Additionally, use "cost functions" that are less dependent on the global intensity difference, such as normalized correlation or mutual information, and avoid using least square error (*see* **Note 2**).

1. Use FSL mcflirt to do motion correction on the ASL data (ASL_DATA_TI1.nii) using the mean image as the reference target, normalized correlation as the cost function, and spline for interpolation:
   mcflirt -in ASL_DATA_TI1.nii -out rASL_DATA_TI1 -cost normcorr -meanvol -spline_final.
2. Do motion correction on another ASL data set (ASL_DATA_TI2.nii). This time using the mean image generated from the first step as the reference target:
   mcflirt -in ASL_DATA_TI2.nii -out rASL_DATA_TI2 -cost normcorr -reffile ASL_DATA_TI1_meanvol.nii -spline_final.
3. Repeat the same procedures until all the ASL data are coregistered.
4. Similarly, do motion correction on T_1 mapping data (T1MAP.nii) also using the mean image generated from the first step as the reference target:
   mcflirt -in T1MAP.nii -out rT1MAP -cost normcorr -reffile ASL_DATA_TI1_meanvol.nii -spline_final.

3.3 Quantification of M_0 and T_1

3.3.1 Model Equations

Both M_0 and T_1 are important parameters needed in ASL kinetic model. Besides, the inversion efficiency will be used for FAIR ASL quantification to correct its "labeling efficiency."

The most common method to obtain these parameters is to fit the model of an inversion recovery experiment to the signal intensity data of each pixel at various TI, $M(\mathrm{TI})$, using the following equation:

$$M(\mathrm{TI}) = \mathrm{abs}\left[M_0\left(1 - 2\alpha \mathrm{e}^{-\frac{TI}{T_1}}\right)\right], \tag{1}$$

where M_0 is the fully relaxed magnetization signal which is a scaling factor that includes many parameters such as the proton density together with the coil sensitivity and the signal gain of the system. α is the inversion efficiency of the inversion pulse ($\alpha = 1$ for a perfect inversion), and T_1 is the longitudinal relaxation time.

3.3.2 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 TIs in a vector (in Matlab: SI_vector = imgData_forFitting(xPix, yPix,:)).

3. As starting value for the parameter M0 use the largest SI (in Matlab: startVal_M0 = max(SI_vector)).
4. For estimating a starting value for T1, one can exploit the fact that the null point of an inversion recovery, where the SI is 0, is $TI = \ln 2 \times T_1 = 0.693 \times T_1$. Estimate the starting T_1 value by finding the TI whose SI is minimal:
5. Find the index of the smallest value in the vector (in Matlab: [min_Value,min_Index] = min(SI_vector)).
6. Get the respective TI using the index from the last step and divide by ln2 to estimate T_1 (in Matlab: startVal_T1 = TI_forFitting(min_Index)/0.693).

As for a starting value for α, one can assume that is close to 1 (in Matlab: startVal_alpha = 1).

3.3.3 Fitting Algorithm

Least squares algorithms, such as the Levenberg–Marquardt and Trust-region methods, are the most commonly used curve fitting algorithms for T_1-mapping. They work by minimizing a cost function, which describes the deviation of the fitted curve from the 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 a wrong 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_1-mapping, where such limits can easily be defined, the Trust-region method is well suited (*see* **Note 3**).

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 TIs in a vector (in Matlab: SI_vector = imgData_forFitting(xPix, yPix,:)).
3. Define the model for the function using Eq. 1 (in Matlab: T1model = fittype('abs(a*(1-2*c*exp.(−x*b)))', 'independent', 'x', 'dependent', 'y')).
4. Initialize the fitting options in Matlab: opts = fitoptions (T1model);
5. Choose the function for the fitting algorithm, that is, Trust-region or, if not available, then Levenberg–Marquardt (in Matlab: opts.Algorithm = 'Trust-Region').
6. Provide the starting values for M_0 and T_1 (in Matlab, e.g., opts. StartPoint = [startVal_M0 startVal_T1 startVal_alpha]).

7. Define lower and upper limits for the fit parameters. Use T1 [1 9000], α [0 1] and for M_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 1 0]; opts.Upper = [65536 9000 1].
8. Execute the curve fitting for each pixel data (in Matlab: [fitresult, gof] = fit(TI_forFitting, SI_vector, T1model, opts);).
9. Save the fit result for each variable in parameter maps: mapM0(xPix, yPix) = fitresult.a; mapT1(xPix, yPix) = fitresult.b; mapAlpha(xPix, yPix) = fitresult.c; rsquare(xPix, yPix) = gof.rsquare.

Besides using Matlab, inversion recovery T_1 curve fitting can be done by using ImageJ. QuickVol II (http://www.quickvol.com/launch2.html) is a powerful ImageJ plugin that supports fitting the inversion recovery data and other user defined equation. The reader should refer to the user's manual on how to use the plugin.

3.3.4 Visual Display

1. Display the parameter map, which is a matrix with floating point numbers, as an image (in Matlab: imagesc(mapT1);).
2. Remove axis labels and ensure that the axes are scaled equally, so that pixels are square and not rectangular (in Matlab: axis off; axis equal;).
3. Select the color map and display a color bar (in Matlab: colormap(jet(256)); colorbar;) (*see* **Note 4**).
4. Set the display range for the color coding, for example, for T_1 [0 2000]; (in Matlab: caxis([0 2000]);).

3.4 Quantification of Perfusion

3.4.1 Model Equation for FAIR-ASL

The pairwise subtracted perfusion-weighted signals at different TIs, ΔM(TI), can be fitted to the following kinetic function [1] (*see* Fig. 1):

$$\Delta M(\mathrm{TI}) = 2M_0\alpha f/\lambda\left[\frac{\exp\left(-\mathrm{TI}/T_{1\mathrm{app}}\right) - \exp\left(-\mathrm{TI}/T_{1\mathrm{a}}\right)}{\left(1/T_{1\mathrm{a}} - 1/T_{1\mathrm{app}}\right)}\right] \quad (2)$$

where $1/T_{1\mathrm{app}} = 1/T_1 + f/\lambda$, f is the quantified perfusion in ml/100 g/min, λ is the blood–tissue partition coefficient. In the above equation, M_0 represents the equilibrium magnetization derived from T_1 mapping, T_1 the tissue longitudinal relaxation time, and α (the inversion efficiency) are all determined from the curve fitting of the inversion recovery T_1 mapping data. λ is a value typically derived from literature which ranges between 0.52 and 0.94 have been reported in [2]. Here, $\lambda = 0.9$ ml/g can be used based on studies in the brain [3]. However, the suitable value for the kidney remains to be determined. $T_{1\mathrm{a}}$ is the longitudinal relaxation time of arterial blood, which is field strength dependent. A

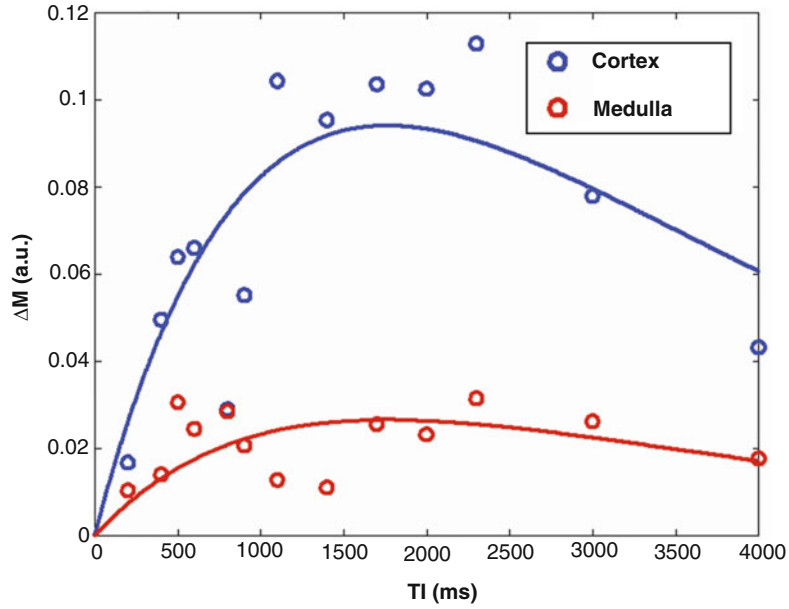

Fig. 1 Fitting perfusion-weighted signal from ROIs in the renal cortex and medulla to the FAIR-ASL kinetic model described in Eq. 2

value of 2.21 s can be used at 7 T [4]. And the blood T_1 at other field strength can be calculated using the equation in [4] (*see* **Note 5**).

3.4.2 Pairwise Subtraction

The first step is to generate the ΔM image of each TI, and then concatenate the results of different TI together to form a 4D time-series data:

1. Use the following FSL command to do pairwise subtraction:

```
asl_file --data = rASL_DATA_TI1 --ntis = 1 --iaf = tc --diff
--out = rASL_DeltaM_TI1 --mean = rASL_DeltaM_TI1_mean
```

Note that the --iaf option is to specify the order of label–control pair where tc means the label (tag) image is acquired first and then followed by control image. Use --iaf = ct if the data is acquired with the opposite order.

2. Concatenate the mean difference images of all the TIs together:

```
fslmerge -t rASL_DeltaM_mean rASL_DeltaM_TI1_mean rASL_DeltaM_TI2_mean ... rASL_DeltaM_TIn_mean
```

   where n is the number of TI.

3. If the data is acquired with varying TIs in one scan, the command could be:

```
asl_file --data = rASL_DATA --ntis = n --iaf = tc --diff --out = rASL_DeltaM --mean = rASL_DeltaM_mean
```

   The resulted rASL_DeltaM_mean image will then be used in the following fitting process.

3.4.3 Starting Values

Deriving the starting value of perfusion from the SI of each pixel is not trivial. Therefore one can just use a literature value (e.g., 300 ml/100 g/min).

1. The starting value for perfusion (startVal_f = 300).

3.4.4 Fitting Algorithm

Similar to T_1 fitting, the Trust-region method can be used.

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 TIs in a vector (in Matlab: SI_vector = imgData_forFitting(xPix, yPix,:)).
3. Change T_1 value from millisecond to second (in Matlab: mapT1(xPix,yPix) = mapT1(xPix,yPix)/1000).
4. The model for the function using Eq. 2 (in Matlab: FAIRmodel = fittype('mapM0(xPix,yPix) * mapAlpha(xPix,yPix) * a * (exp(−x * (1/mapT1(xPix,yPix) + a)) − exp.(−x/0.221))/(1/0.221 − (1/mapT1(xPix,yPix) + a)) ', 'independent', 'x', 'dependent', 'y')).
5. Initialize the fitting options in Matlab: opts = fitoptions (FAIRmodel);
6. Choose the function for the fitting algorithm, that is, Trust-region or, if not available, then Levenberg–Marquardt (in Matlab: opts.Algorithm = 'Trust-Region').
7. Provide the starting values for perfusion (in Matlab, e.g., opts. StartPoint = [startVal_f]).
8. Define lower and upper limits for the fit parameters. Assuming perfusion will be below 900 ml/100 g/min, use for f [0 900/6000]. In Matlab, for example, opts.Lower = [0]; opts. Upper = [0.15] (*see* **Note 6**).

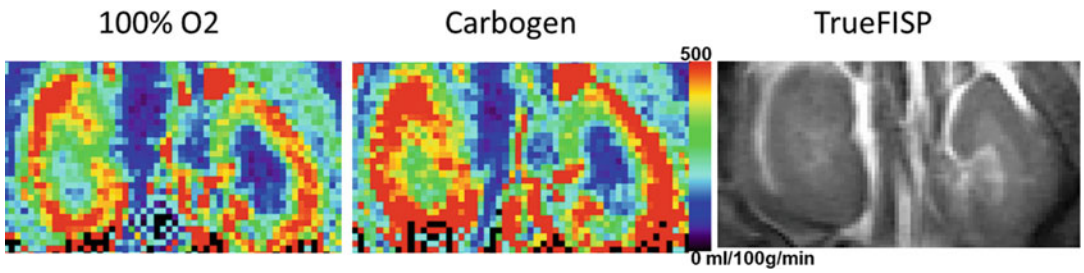

Fig. 2 Quantitative perfusion maps show lower perfusion under hyperoxia than that under carbogen (5% CO_2 + 95% O_2). Anatomical image was acquired with TrueFISP showing contrast between the cortex (dark) and medulla

9. Execute the curve fitting for each pixel data (in Matlab: [fitresult, gof] = fit(TI_forFitting, SI_vector, FAIRmodel, opts);).
10. Save the fit result in parameter maps and rescale by $2/\lambda$ and unit conversion factor: mapF(xPix, yPix) = fitresult.a * 0.9/2 * 6000; rsquare(xPix, yPix) = gof.rsquare (*see* **Note 6**).

3.4.5 Visual Display

1. Display the parameter map, which is a matrix with floating point numbers, as an image (in Matlab: imagesc(mapF);) (*see* Fig. 2).
2. Remove axis labels and ensure that the axes are scaled equally, so that pixels are square and not rectangular (in Matlab: axis off; axis equal;).
3. Select the color map and display a color bar (in Matlab: colormap(jet(256)); colorbar;) (*see* **Note 4**).
4. Set the display range for the color coding, for example, for perfusion [0 500] (in Matlab: caxis([0 500]);).

3.5 Regional Analysis

To obtain quantitative medulla and cortex perfusion, the kidney can be manually or automatically segmented based on the T_1 map or other kinds of structural images.

3.6 Results Validation

3.6.1 Comparison with Tissue Values Under Hypercapnia or Hyperoxia

Further validation of imaging derived results can be performed using datasets acquired under physiological manipulations that alter blood flow, such as hypercapnia or hyperoxia. As CO_2 is a potent vasodilator, the blood flow under hypercapnia condition should be higher than that in normal gas condition. Hyperoxia is expected to lead to the opposite result with O_2 acting like a vasoconstrictor (*see* Fig. 2).

3.6.2 Comparison with Reference Values from the Literature

Obtained values for both cortex and medulla tissue segmentation can be compared against reference values, shown in Table 1. It should be noted that blood flow is also modulated by anesthesia. For example, the commonly used isoflurane is a vasodilator and can increase blood flow depending on dosage. Therefore, the

Table 1
Comparison of renal blood flow measured by ASL-MRI in rodents

Cortex (ml/100 g/min)	Medulla (ml/100 g/min)	Animal strain, anesthesia and gas	Reference
338–452	123–240	C57BL/6 mouse, 1–1.5% isoflurane in 100% O_2	[5]
550–750	140–230	C57BL/6J mouse, 1.5% isoflurane (gas unknown)	[6]
344–575	ND	C57BL/6N mouse, isoflurane (dose and gas unknown)	[7]
288.4 ± 51.3	ND	Spraque-Dawley rat, 1.5% isoflurane (gas unknown)	[8]
750 ± 80	ND	Fisher 344 rat, thiobutabarbitural, 69-135 mg/kg (gas unknown)	[9]

comparison with literature values should be done with comparable anesthesia dosage.

4 Notes

1. Processing is typically performed in software packages such as MATLAB (The MathWorks, MA) or open source platforms as R programming language (http://r.org). The data can either be processed as DICOM images or if available as the raw data format from the scanner. As some motion correction tools (e.g., FSL, SPM) works on NiFti format, the data may need to be converted to that format during the processing. At the end of the processing pipeline, finalizing the data in the DICOM format would allow for further analysis and comparison to other image data on clinical viewers.
2. To correct for respiratory induced motion, resulted ASL images can be coregistered using Elastix (open source software, http://elastix.isi.uu.nl/) and a rigid registration algorithm.
3. An approach that is frequently used to increase the speed of the curve fitting, is to fit a linear model to the log of the SI data. This only requires a few simple modifications to the protocol described here. We recommend using exponential fitting (unless speed is very important), because the log-scaling leads to the noise error at each TI being scaled differently and this will impact on the fitted curve. If you do want to use linear fitting it would be good to initially perform both, exponential and linear fitting, on a few data sets and compare the results so as to better appreciate the impact of this choice.
4. Pseudocolor representations can be extremely useful for analyzing T_1-maps or perfusion 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 emphasized 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.

5. The transit time is ignored in the above equation as the delivery of spins is almost instantaneous with a small gap between the imaging and inversion slices used in FAIR ASL. Whether the transit time is negligible can be verified by adding the transit time into the kinetic model.
6. To get the correct unit, the T_1, T_{1a} and T_{1app} should be expressed in "second" and the perfusion value, f, will need to be multiplied by 6000 to convert to per 100 g/min.

Acknowledgments

This chapter is based upon work from COST Action PARENCHIMA, 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. Pell GS, Thomas DL, Lythgoe MF, Calamante F, Howseman AM, Gadian DG et al (1999) Implementation of quantitative FAIR perfusion imaging with a short repetition time in time-course studies. Magn Reson Med 41 (4):829–840
2. Kudomi N, Koivuviita N, Liukko KE, Oikonen VJ, Tolvanen T, Iida H et al (2009) Parametric renal blood flow imaging using [15O]H2O and PET. Eur J Nucl Med Mol Imaging 36 (4):683–691
3. Herscovitch P, Raichle ME (1985) What is the correct value for the brain–blood partition coefficient for water? J Cereb Blood Flow Metab 5 (1):65–69
4. Dobre MC, Ugurbil K, Marjanska M (2007) Determination of blood longitudinal relaxation time (T1) at high magnetic field strengths. Magn Reson Imaging 25(5):733–735
5. Rajendran R, Lew SK, Yong CX, Tan J, Wang DJ, Chuang KH (2013) Quantitative mouse renal perfusion using arterial spin labeling. NMR Biomed 26(10):1225–1232

6. Duhamel G, Prevost V, Girard OM, Callot V, Cozzone PJ (2014) High-resolution mouse kidney perfusion imaging by pseudo-continuous arterial spin labeling at 11.75T. Magn Reson Med 71(3):1186–1196
7. Hueper K, Gutberlet M, Rong S, Hartung D, Mengel M, Lu X et al (2014) Acute kidney injury: arterial spin labeling to monitor renal perfusion impairment in mice-comparison with histopathologic results and renal function. Radiology 270(1):117–124
8. Romero CA, Cabral G, Knight RA, Ding G, Peterson EL, Carretero OA (2018) Noninvasive measurement of renal blood flow by magnetic resonance imaging in rats. Am J Physiol Renal Physiol 314(1):F99–F106
9. Wang JJ, Hendrich KS, Jackson EK, Ildstad ST, Williams DS, Ho C (1998) Perfusion quantitation in transplanted rat kidney by MRI with arterial spin labeling. Kidney Int 53 (6):1783–1791