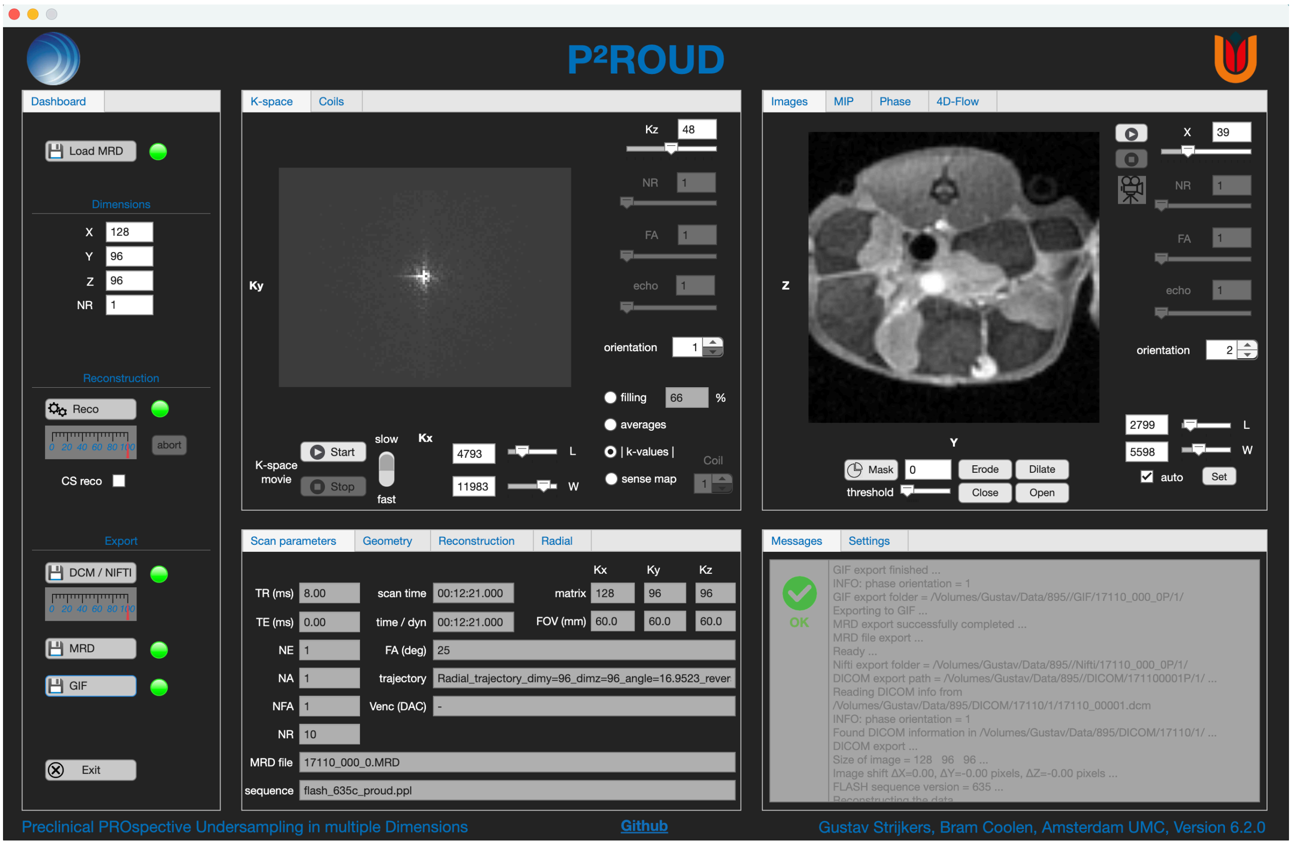


X 128
Y 96
Z 96
NR 1

Kz 48
NR 1
FA 1
echo 1
orientation 1
filling 66 %
4793 L
11983 W
Coil 1

X 39
NR 1
FA 1
echo 1
orientation 2
2799 L
5598 W
Mask 0

Scan parameters

				Kx	Ky	Kz
TR (ms)	8.00	scan time	00:12:21.000	128	96	96
TE (ms)	0.00	time / dyn	00:12:21.000	60.0	60.0	60.0
NE	1	FA (deg)	25			
NA	1	trajectory	Radial_trajectory_dimy=96_dimz=96_angle=16.9523_rever			
NFA	1	Venc (DAC)	-			
NR	10					
MRD file	17110_000_0.MRD					
sequence	flash_635c_proud.ppl					

(matrix: Kx 128, Ky 96, Kz 96; FOV (mm): 60.0, 60.0, 60.0)

```
GIF export finished ...
INFO: phase orientation = 1
GIF export folder = /Volumes/Gustav/Data/895//GIF/17110_000_0P/1/
Exporting to GIF ...
MRD export successfully completed ...
MRD file export ...
Ready ...
Nifti export folder = /Volumes/Gustav/Data/895//Nifti/17110_000_0P/1/
DICOM export path = /Volumes/Gustav/Data/895//DICOM/171100001P/1/ ...
Reading DICOM info from
/Volumes/Gustav/Data/895/DICOM/17110/1/17110_00001.dcm
INFO: phase orientation = 1
Found DICOM information in /Volumes/Gustav/Data/895/DICOM/17110/1/ ...
DICOM export ...
Size of image = 128   96   96 ...
Image shift ΔX=0.00, ΔY=-0.00 pixels, ΔZ=-0.00 pixels ...
FLASH sequence version = 635 ...
Reconstructing the data
```

Preclinical PROspective Undersampling in multiple Dimensions

Github

Gustav Strijkers, Bram Coolen, Amsterdam UMC, Version 6.2.0

Table of contents

1. Overview

P2ROUD = Preclinical PROspective Undersampling in multiple Dimensions

P2ROUD is a software utility and graphical user interface to reconstruct MRI acquisition from data with k-space undersampling in one or more dimensions acquired with an MR Solutions preclinical MRI system.

2. Contact

P2ROUD and this manual are written by:

Gustav Strijkers
Amsterdam UMC
Biomedical Engineering and Physics
Amsterdam, the Netherlands
g.j.strijkers@amsterdamumc.nl

and

Bram Coolen
Amsterdam UMC
Biomedical Engineering and Physics
Amsterdam, the Netherlands
b.f.coolen@amsterdamumc.nl

3. Disclaimer

The software has been extensively tested using mouse and rat data acquired with an MR Solutions preclinical 7.0T/24cm system.

However, this does not warrant the functions contained in the program will meet your requirements or that the operation of the program will be uninterrupted or error-free.

In case of questions or issues, please contact Gustav Strijkers or Bram Coolen.

4. Installation notes

Software download

Matlab source code and a Windows standalone version (using the free Matlab runtime engine) can be downloaded from github:

`https://github.com/Moby1971?tab=repositories`

Installation of the Windows standalone version

`P2ROUD_app_installer.exe`

Will install the Matlab runtime engine and the P2ROUD program.

Bart toolbox download

The advanced reconstruction options in P2ROUD require the Bart reconstruction toolbox, which can be downloaded from:

`https://mrirecon.github.io/bart/`

Bart toolbox installation in OSX with MacPorts

(1) Install Xcode from the Mac App Store

```
$ xcode-select --install
```

(2) Install MacPorts (`http://www.macports.org/`)
It is recommended to install a newer version of gcc from MacPorts

(3) Install packages

```
$ sudo port install fftw-3-single
$ sudo port install gcc12
$ sudo port install libpng
$ sudo port install openblas
$ sudo port install flock
$ sudo port install gmake
```

(4) Make

```
$ CC=gcc-mp-12 gmake
$ sudo gmake install
```

Bart toolbox installation in OSX with HomeBrew

(1) Install Xcode from the Mac App Store

```
$ xcode-select --install
```

(2) Install HomeBrew

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/
install.sh)"
```

(3) Install packages

```
$ brew install --cask gcc-arm-embedded
$ brew install libpng
$ brew install fftw
$ brew install openblas
$ brew install gmake
$ brew install llvm libomp
```

(3) Make

```
$ CC=gcc MACPORTS=0 make
$ sudo make install
```

Bart toolbox installation in Windows 11

(1) Install the Windows subsystem for Linux

Start a windows powershell as administrator and run the following command:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-
Subsystem-Linux
```

A system restart will be needed.

For more information, see:

`https://pureinfotech.com/install-wsl-windows-11/`

(2) Download Ubuntu Linux

Open PowerShell

To see a list of available Linux distributions for download, enter:

```
wsl -l -o
```

Install a Linux distribution:

```
wsl --install -d <distribution name>
```

You will be asked to create a user account the first time you start a Linux command prompt.

(3) Upgrade to WSL2 if needed

(see e.g. `https://pureinfotech.com/install-wsl-windows-11/` on how to upgrade).

(4) Upgrade the Linux distribution

```
$ sudo apt-get update
$ sudo apt-get dist-upgrade
```

(5) Install Bart prerequisites

```
$ sudo apt-get install make gcc libfftw3-dev liblapacke-dev libpng-dev gfortran
```

(6) Download Bart

```
$ wget https://github.com/mrirecon/bart/archive/v0.9.00.tar.gz
```

(7) Build Bart

```
$ tar xvfz v0.9.00.tar.gz
$ cd bart-0.9.00
$ make
$ sudo make install
```

(8) Build Bart with GPU support (highly recommended)

- Install NVIDIA and CUDA software in windows
  (https://docs.nvidia.com/cuda/wsl-user-guide/index.html)

- Install CUDA toolkit in wsl2:
  (https://developer.nvidia.com/cuda-downloads)

```
$ wget https://developer.download.nvidia.com/compute/cuda/12.8.0/
  local_installers/cuda_12.8.0_570.86.10_linux.run
$ sudo sh cuda_12.8.0_570.86.10_linux.ru
```

- Set some parameters (a bit messy, but it worked)

```
$ export CUDA=1
  export CUDA_BASE=/usr/local/cuda-12.8
  export CUDA_LIB=lib64
  export GPUARCH_FLAGS="-arch=sm_61"
  export PATH=$CUDA_BASE/bin:$PATH
  export LD_LIBRARY_PATH=$CUDA_BASE/$CUDA_LIB:$LD_LIBRARY_PATH
  export CPATH=/usr/local/cuda-12.8/targets/x86_64-linux/include:$CPATH
  export LD_LIBRARY_PATH=/usr/local/cuda-12.8/targets/x86_64-linux/lib:$LD_LIBRARY_PATH
  export PATH=/usr/local/cuda-12.8/bin:$PATH
  export CUDA_HOME=/usr/local/cuda-12.8
  export CUDA_TOOLKIT_ROOT_DIR=$CUDA_HOME
  export LD_LIBRARY_PATH="$CUDA_HOME/extras/CUPTI/lib64:$LD_LIBRARY_PATH”
  export LIBRARY_PATH=$CUDA_HOME/lib64:$LD_LIBRARY_PATH
  export LDFLAGS="-L/usr/lib/gcc/x86_64-linux-gnu/13 $LDFLAGS"
```

- Make

```
$ make -j$(nproc) CUDA=1 CUDA_BASE=$CUDA_BASE CUDA_LIB=$CUDA_LIB GPUARCH_FLAGS="$GPUARCH_FLAGS"
```

- Test

```
$ make utest_gpu
```

Required Python modules for Deep Learning noise suppression (beta)
It is highly recommended to have a GPU and CUDA for speed.

- Install conda in wsl2
  (https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html)

```
$ sudo apt-get update
$ sudo apt install python
$ sudo apt install python-is-python3
$ sudo apt install python3-pip
$ sudo apt install unzip
$ pip install tensorboard tensorboard-data-server imageio matplotlib
  matplotlib-inline opencv-python  scikit-image  pillow h5py  torch
  torchvision  torchmetrics  tqdm  pandas  scipy  opencv-python interval
  unzip nibabel  tensorboard
```

6. Running the software

Running in Matlab 2025a

The P2ROUD software can be started from its root directory from the command line.

```
>> p2roud
```

Additional licenses may be required.

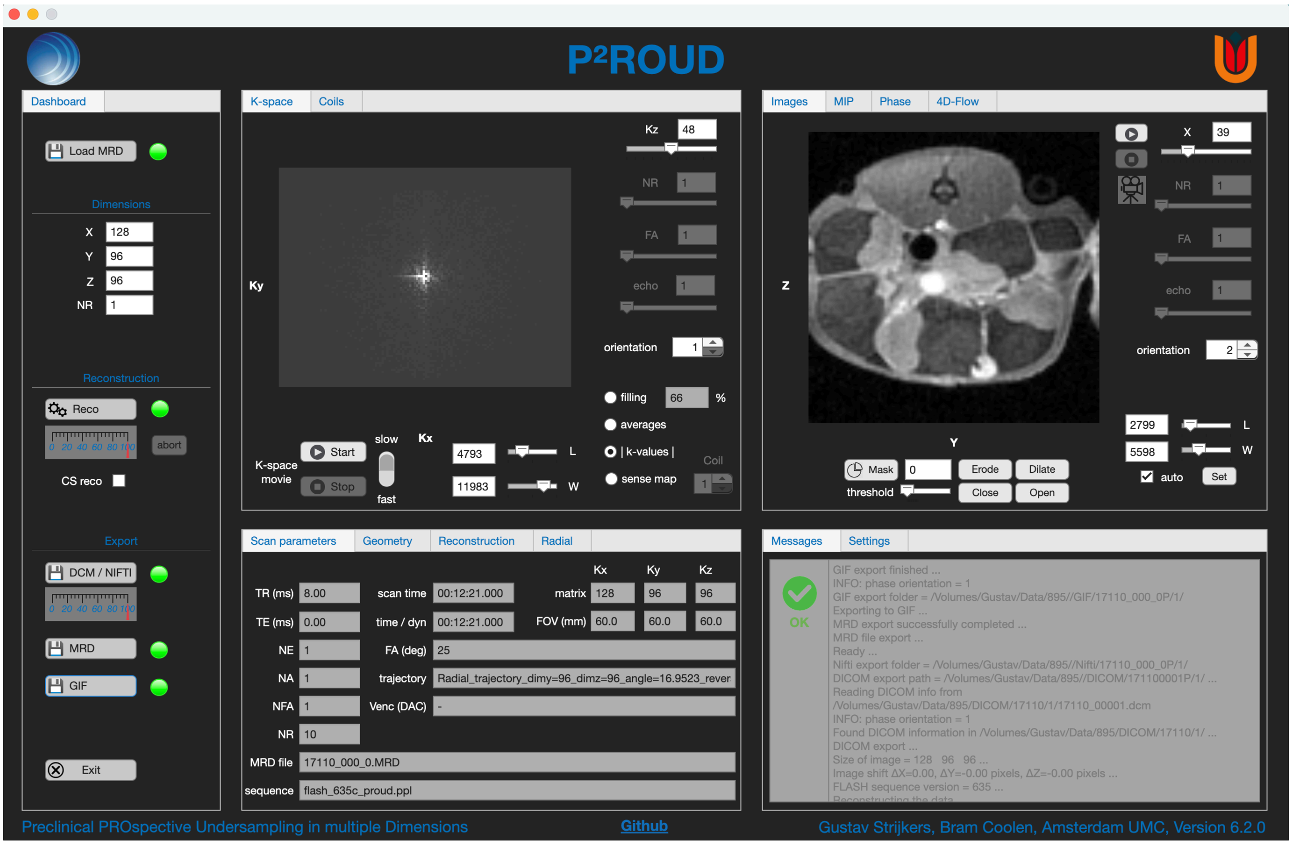
```
MATLAB
Image Processing Toolbox
Parallel Computing Toolbox
Signal Processing Toolbox
```

Running the Windows standalone

The Windows standalone version can be run from the start menu or the desktop icon.

6. Basic operation

The P2ROUD program operates from a single window with 5 panels.



Dimensions: X 128, Y 96, Z 96, NR 1

K-space: Kz 48, NR 1, FA 1, echo 1, orientation 1; ○ filling 66 %, ○ averages, ● | k-values |, ○ sense map; L 4793, W 11983

Images: X 39, NR 1, FA 1, echo 1, orientation 2; L 2799, W 5598, ☑ auto, Set; Mask 0

Scan parameters:

					Kx	Ky	Kz
TR (ms)	8.00	scan time	00:12:21.000	matrix	128	96	96
TE (ms)	0.00	time / dyn	00:12:21.000	FOV (mm)	60.0	60.0	60.0
NE	1	FA (deg)	25				
NA	1	trajectory	Radial_trajectory_dimy=96_dimz=96_angle=16.9523_rever				
NFA	1	Venc (DAC)	-				
NR	10						
MRD file	17110_000_0.MRD						
sequence	flash_635c_proud.ppl						

```
GIF export finished ...
INFO: phase orientation = 1
GIF export folder = /Volumes/Gustav/Data/895//GIF/17110_000_0P/1/
Exporting to GIF ...
MRD export successfully completed ...
MRD file export ...
Ready ...
Nifti export folder = /Volumes/Gustav/Data/895//Nifti/17110_000_0P/1/
DICOM export path = /Volumes/Gustav/Data/895//DICOM/171100001P/1/ ...
Reading DICOM info from
/Volumes/Gustav/Data/895/DICOM/17110/1/17110_00001.dcm
INFO: phase orientation = 1
Found DICOM information in /Volumes/Gustav/Data/895/DICOM/17110/1/ ...
DICOM export ...
Size of image = 128   96   96 ...
Image shift ΔX=0.00, ΔY=-0.00 pixels, ΔZ=-0.00 pixels ...
FLASH sequence version = 635 ...
Reconstructing the data
```

Panel 1: Dashboard
This panel contains the task buttons and parameters that control the reconstruction process. The dashboard tasks need to be completed from top to bottom. A green light next to the task indicates that the task has been completed. When the light is yellow the task is busy. Red indicates not completed yet.

Panel 2: K-space / Coils
This panel shows the raw k-space data.

Panel 3: Scan parameters / Geometry / Reconstruction / Radial
This panel displays the relevant acquisition, geometry, reconstruction and radial reconstruction parameters.

Panel 4: Images / MIP / Phase / 4D-flow
After reconstruction is completed this panel shows the reconstructed images.

Panel 5: Messages / Settings
Program status, messages, and app settings.

Step 1: Loading data

Press 💾 Load MRD to import load the MRD raw data file.

P2ROUD in principle can reconstruct any spin-echo, fast-spin echo, and gradient-echo data (except epi).

However the program is particularly designed to reconstruct undersampled data with user-defined undersampling pattern (e.g. SENSE, random undersampling, pseudo-spiral or radial).

For data acquired with multiple receivers (phased-array RF coil) select 1 of the MRD files.

Relevant acquisition parameters will be shown in panel 3.

k-space trajectory

Currently there are 2 options to include a user-defined k-space trajectory in the FLASH (version 635e) sequence.

(1) **2D-data**

pe1-order = 3 ("LUT")

The k-space points for the 1st phase-encoding direction (views) are provided in the views-table in the sequence.

(2) **3D-data**

pe1-order = 4 ("exLUT")

The k-space points for the 1st and 2nd phase-encoding directions (views and views2) are provided in a text file (*.txt) as a list of points without comma separation.

For example:

```
-48       (views    k-point 1)
0         (views2   k-point 1)
-47       (views    k-point 2)
0         (views2   k-point 2)
-46       .
1         .
-45       .
.
.
etc...
```

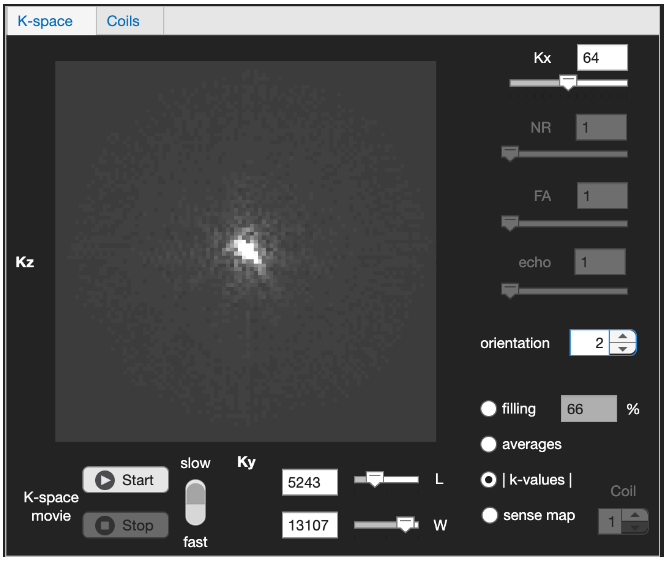
(3) **Undersampled 2D- and 3D-data**

pe1-order = 5 ("nrLUT")

In /function/trajectories matlab and python code for making trajectories can be found.

Step 2: Inspecting k-space

The raw k-space data is displayed in panel 2.



Kx 64

NR 1

FA 1

echo 1

orientation 2

○ filling 66 %

○ averages

● | k-values |

○ sense map

Coil 1

L 5243

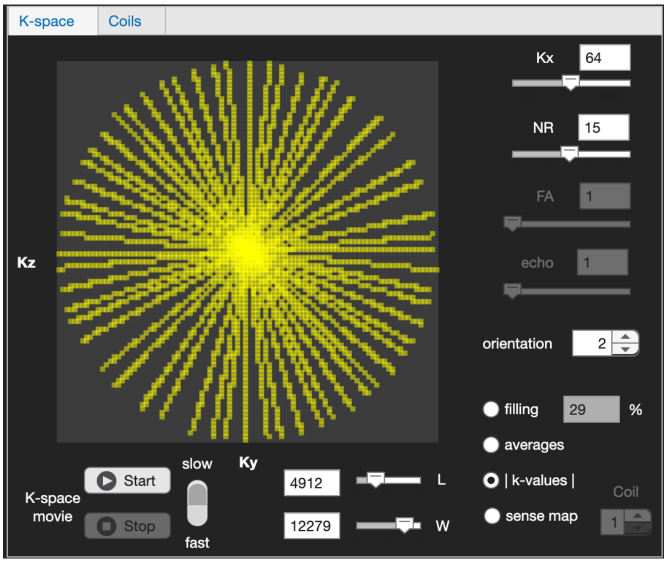
W 13107

The window displays either the acquired k-space points (**filling**), the number of averages per k-point (**averages**), or the absolute value of k-space (**|k-values|**). The percentage of k-space points acquired is indicated (66% for this dataset).

The sliders step through the 3rd k-space dimension, slices, repetitions (**NR**), flip angles (**FA**) and echoes (**Echo**), if applicable. You can toggle the orientation with the toggle buttons.

Step 3: Displaying the k-space trajectory

Press ▶ Start to start an animation of the k-space trajectory. An example for a pseudo-radial k-space trajectory is shown below



Kx 64

NR 15

FA 1

echo 1

orientation 2

○ filling 29 %

○ averages

● | k-values |

○ sense map

Coil 1

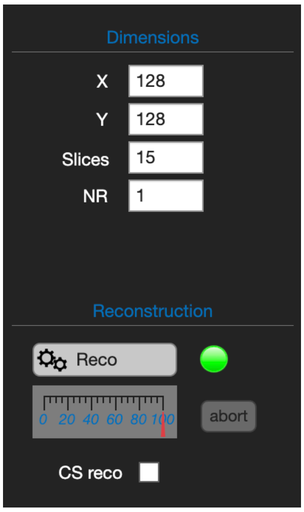
L 4912

W 12279

Press ■ Stop to stop the animation. Press slow / fast to toggle between fast and slow animation speed.

Step 4: Fourier transform reconstruction

For a first quick inspection of the reconstructed images, a Fourier transform reconstruction of the data can be done by pressing Reco .



Dimensions

X 128

Y 128

Slices 15

NR 1

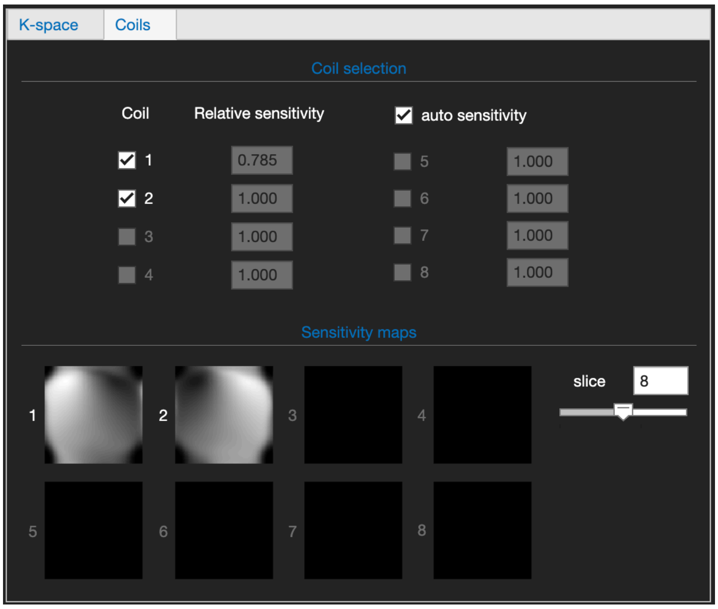
Reconstruction

CS reco □

The spatial dimensions of the reconstructed data can be increased or decreased, which will result in zero-padding or cropping of k-space.

Step 5: Coils panel

For acquisitions performed with a phased-array coil the **Coils** panel gives access to the different channels. Here coils can be checked/unchecked to be considered/omitted in the reconstruction. Relative coil sensitivities can be changed or automatically determined by checking the Auto sensitivity checkbox.



Coil selection

☑ auto sensitivity

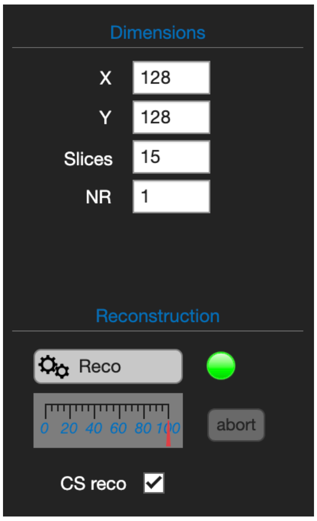
Coil	Relative sensitivity	Coil	Relative sensitivity
☑ 1	0.785	□ 5	1.000
☑ 2	1.000	□ 6	1.000
□ 3	1.000	□ 7	1.000
□ 4	1.000	□ 8	1.000

Sensitivity maps

slice 8

Step 6: Reconstruction

Checking the **CS reco** box gives access to the Compressed Sensing (CS) reconstruction parameters.



Dimensions

X 128

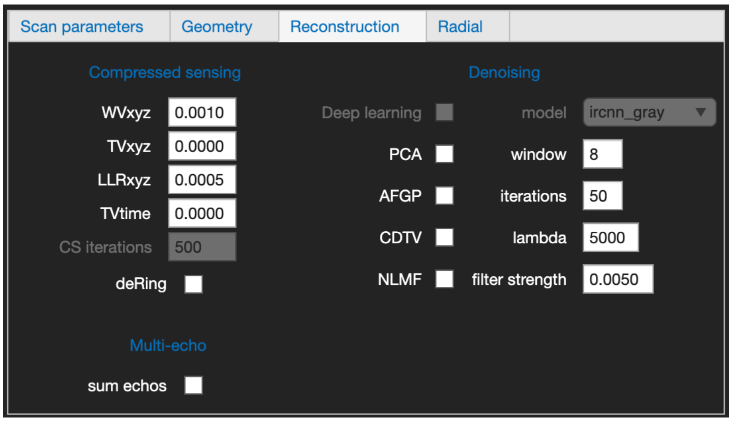
Y 128

Slices 15

NR 1

Reconstruction

CS reco ☑



Compressed sensing

WVxyz	0.0010
TVxyz	0.0000
LLRxyz	0.0005
TVtime	0.0000
CS iterations	500
deRing	□

Multi-echo

sum echos □

Denoising

Deep learning	□	model	ircnn_gray
PCA	□	window	8
AFGP	□	iterations	50
CDTV	□	lambda	5000
NLMF	□	filter strength	0.0050

The CS regularization parameters are:

(1) **WVxyz**
Wavelet regularization parameter in the spatial (x, y, and z) dimensions.

(2) **TVxyz**
Total variation constraint in the spatial dimensions.

(3) **LRxyz**
Locally Low Rank constraint in the spatial dimensions.

(4) **TVtime**
Total variation constraint in the dynamics (repetitions / **NR**) dimension.

The spatial dimensions of the reconstructed data can be increased or decreased. The number of repetitions (**NR**) can also be changed to undersample or average data in the dynamics dimension.

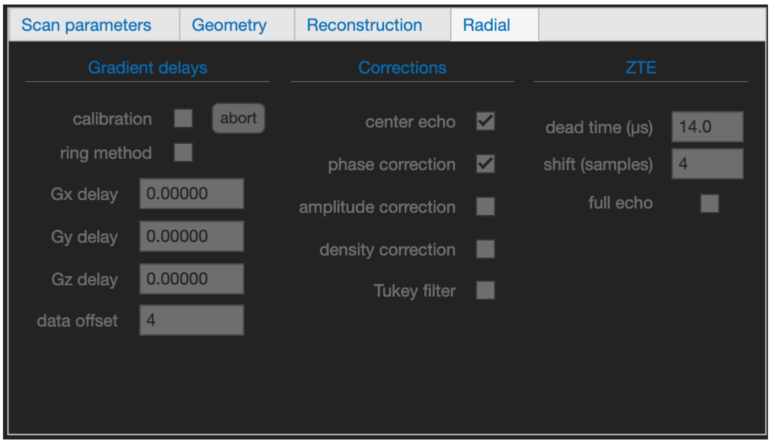
Pressing Reco will start the CS reconstruction with the BART toolbox.

If the BART toolbox is not available CS reconstruction will be done using a CS algorithm implemented in Matlab, which can be very slow for large datasets.

Several (experimental) denoising algorithms are available.
DeRing applies a Gibbs ringing suppression filter.

Reconstruction of radial scans

The *Radial* tab gives access to advanced reconstruction parameters for radial (2D / 3D) scans.



calibration □

ring method □

Gx delay 0.00000

Gy delay 0.00000

Gz delay 0.00000

data offset 4

center echo ☑

phase correction ☑

amplitude correction □

density correction □

Tukey filter □

dead time (µs) 14.0

shift (samples) 4

full echo □

Tick the *calibration* checkbox for automatic adjustment of the gradient delays. When the *ring method* checkbox is ticked, the RING method described by Rosenzweig et al. is used [Rosenzweig, S., Holme, H. C. M. & Uecker, M. Simple auto-calibrated gradient delay estimation from few spokes using Radial Intersections (RING). Magn Reson Med 81, 1898–1906 (2019)]. When *ring method* is unchecked, a recursive image-based gradient delay optimization is performed. The latter method is much slower.

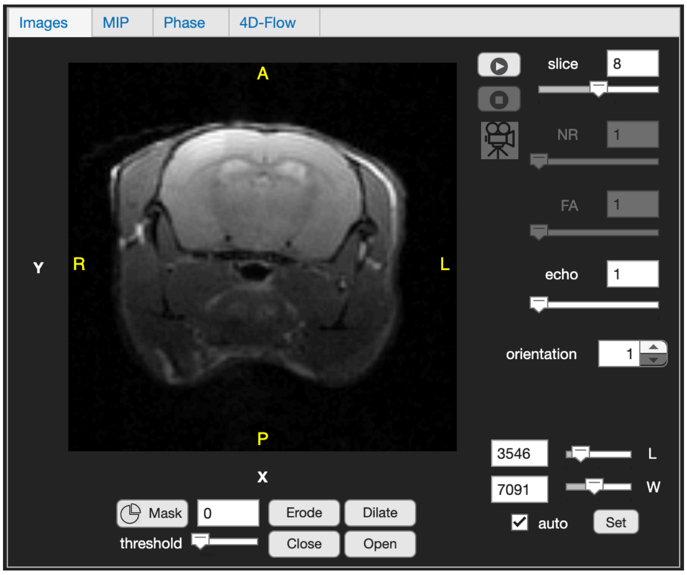
With *center echo* checked, the echoes are centered before reconstruction.

Phase correction performs a phase correction of the echoes before reconstruction.

Density correction performs a 2D or 3D density correction in k-space to account for non-uniform sampling. This setting only applies to recontructions with *Bart*.

Step 7: Viewing the reconstructed images

After reconstruction the MR images are displayed in panel 4.



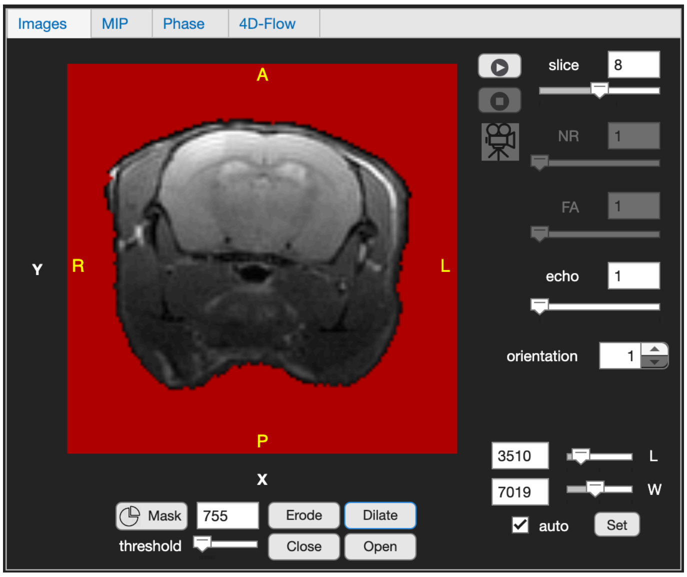
slice 8
NR 1
FA 1
echo 1
orientation 1
3546
7091
☑ auto
Mask 0

The image orientation can be changed by pressing the **Orientation** toggle buttons.

Image intensity can be adjusted with the **Level** and **Window** sliders. Checking **Auto level** will result in automatic intensity adjustment.

Press Start to start a movie of images as function of the dimension / parameter shown next to the camera icon .

Images can be segmented (on the basis of a threshold) by pressing Mask .

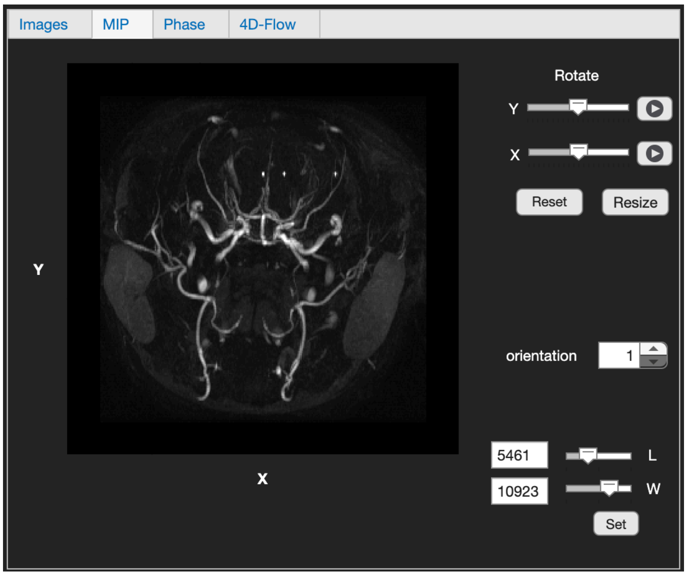


slice 8
NR 1
FA 1
echo 1
orientation 1
3510
7019
☑ auto
Mask 755

Use the threshold slider or morphological operations to fine-tune the segmentation.

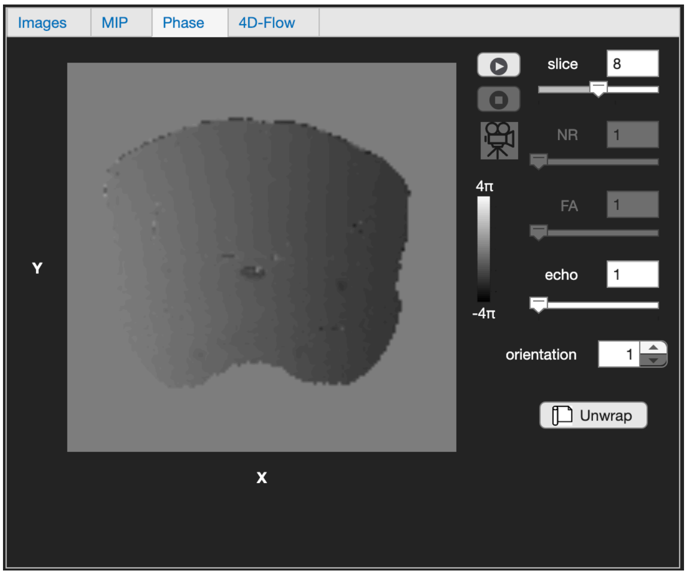
Maximum intensity projection (MIP)

In the *MIP* tab a maximum intensity projection is shown. The MIP can be rotated and animated using the available sliders and buttons.



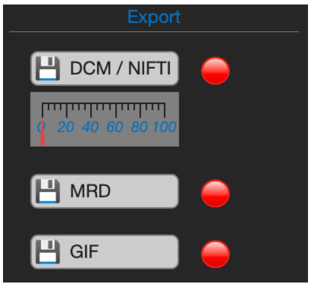
Phase images

The phase tab shows the phase images.



Phase can be unwrapped by pressing the Unwrap button.

Step 8: Image export



There are several ways to export the images.

(1) **DCM / NIFTI**

Exports the data in Dicom format for further processing in 3rd party software. The program searches for the Dicom information. If this information is not found, tags will be generated by the program itself. In the latter case the correct image position and orientation information are lost.

After Dicom exports data will be converted into NIFTI format for further processing in 3rd party software.

(3) **MRD**

The data will be stored as a new MRD data file in the same directory as the input MRD file. If the program finds an input rpr file, also a new rpr file will be generated. Additionally, when the program runs on the scanner computer, data will be reconstructed by the scanner software and new SUR files will be placed in the data directory for viewing and planning of next scans in the MR Solutions preclinical software.

(4) **GIF**

The data will be exported as (animated) gif files.

Step 9: Exit

Press ⊗ Exit to shut down the program.

Credits

- BART toolbox and compressed sensing algorithms

  https://mrirecon.github.io/bart/

  Michael Lustig, David Donoho, John M. Pauly, Sparse MRI: The application of compressed sensing for rapid MR imaging, Magn Reson Med 58, 1182-1195 (2007).

- Automatic gradient delay estimation algorithm

  Rosenzweig, S., Holme, H. C. M. & Uecker, M., Simple auto-calibrated gradient delay estimation from few spokes using Radial Intersections (RING). Magn Reson Med 81, 1898–1906 (2019)

- Gibbs ringing suppression

  Kellner, E, Dhital B., Kiselev VG and Reisert, M., Gibbs-ringing artifact removal based on local subvoxel-shifts. Magn Reson Med, 76(5), 1574-1581.

- NUFFT

  https://web.eecs.umich.edu/~fessler/

- Phase unwrapping

  M. A. Herráez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat, "Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path", Applied Optics, Vol. 41, Issue 35, pp. 7437-7444 (2002),

  H. Abdul-Rahman, M. Gdeisat, D. Burton, M. Lalor, "Fast three-dimensional phase-unwrapping algorithm based on sorting by reliability following a non-continuous path", Proc. SPIE 5856, Optical Measurement Systems for Industrial Inspection IV, 32 (2005).