In[11]:=
```
QMRITools`$Verbose = True;
<< QMRITools`
```

```
----------------------------------------

Loading QMRITools` with version number 2.10.1

----------------------------------------

Defined packages and functions to be loaded are:

   - QMRITools`GeneralTools` with functions:

{ApplyCrop, AutoCropData, BSplineCurveFit, CheckExtension, ClearTemporaryVariables, CompilebleFunctions, CropData, CropInit, CropOutput, CropPadding,
 CutData, Data2DToVector, Data3DToVector, DataToVector, DecomposeAffineMatrix, DecomposeScaleMatrix, DevideNoZero, DynamicPartition, ExpNoZero,
 ExtractDemoData, FileSelect, FindCrop, FindMaxDimensions, GetAssetLocation, GridData, GridData3D, GyromagneticRatio, LapFilter, LLeastSquares,
 LogNoZero, MADNoZero, MakeIntFunction, MeanNoZero, MedFilter, MedianNoZero, MemoryUsage, NNLeastSquares, OutputWeights, PadDirection, PadToDimensions,
 PadValue, QMRIToolsFuncPrint, QMRIToolsFunctions, QMRIToolsPackages, QuaternionToRotationMatrix, QuaternionVectorToRotationMatrix, RescaleData,
 ReverseCrop, ReverseDimensions, RMSNoZero, RotateDimensionsLeft, RotateDimensionsRight, RotationMatrixToQuaternion, RotationMatrixToQuaternionVector,
 SaveImage, SplineKnotsNumber, SplineRegularization, Squeeze, StdFilter, StichData, SumOfSquares, TensMat, TensVec, VectorToData}

   - QMRITools`MaskingTools` with functions:

{GetMaskData, GetMaskOnly, GetMaskOutput, GrowMask, HomoginizeData, Mask, MaskClosing, MaskComponents,
 MaskData, MaskDilation, MaskFiltKernel, MaskSmoothing, MeanSignal, MergeSegmentations, NormalizeData, NormalizeMeanData,
 RemoveMaskOverlaps, RescaleSegmentation, ROIMask, SegmentMask, SmoothMask, SmoothSegmentation, SplitSegmentations, UseMask}

   - QMRITools`NiftiTools` with functions:

{CompressNii, CompressNiiFiles, CorrectNiiOrientation, DcmToNii, DeleteOutputFolder, ExportBmat, ExportBval, ExportBvec, ExportNii, ExtractNiiFiles, FlipBvec,
 GetNiiOrientation, ImportBmat, ImportBval, ImportBvalvec, ImportBvec, ImportExploreDTItens, ImportNii, ImportNiiDiff, ImportNiiDix, ImportNiiT1, ImportNiiT2,
 MakeNiiOrentationQ, MakeNiiOrentationS, NiiDataType, NiiLegacy, NiiMethod, NiiOffset, NiiScaling, NiiSliceCode, RotateGradients, UseSubfolders, UseVersion}

   - QMRITools`ElastixTools` with functions:

{AffineDirections, BsplineDirections, BsplineSpacing, DeleteTempDirectory, FindTransform, HistogramBins, HistogramBinsA, InterpolationOrderReg, InterpolationOrderRegA,
 Iterations, IterationsA, MethodReg, MethodRegA, NumberSamples, NumberSamplesA, OutputImage, OutputTransformation, PCAComponents, PrintTempDirectory,
 ReadTransformParameters, RegisterCardiacData, RegisterData, RegisterDataSplit, RegisterDataTransform, RegisterDataTransformSplit, RegisterDiffusionData,
 RegisterDiffusionDataSplit, RegistrationTarget, Resolutions, ResolutionsA, SplitMethod, TempDirectory, TransformData, UseGPU, $debugElastix}

   - QMRITools`PlottingTools` with functions:

{DropSlices, GetSliceData, GetSlicePositions, GradientPlot, ImageLegend, ImageOrientation, ListSpherePlot, MakeCheckPlot, MakeSliceImages, NormalizeIVIM, PeakNumber,
 PlotColor, PlotContour, PlotCorrection, PlotData, PlotData3D, PlotDefGrid, PlotDuty, PlotIVIM, PlotMoments, PlotSequence, PlotSpace, PositiveZ, SphereColor, SphereSize}

   - QMRITools`DixonTools` with functions:

{DixonAmplitudes, DixonBipolar, DixonClipFraction, DixonFieldStrength, DixonFilterInput, DixonFilterOutput,
 DixonFilterSize, DixonFrequencies, DixonIterations, DixonMaskThreshhold, DixonNucleus, DixonPrecessions, DixonReconstruct,
 DixonTollerance, DixonToPercent, MakeGroups, MonitorUnwrap, SimulateDixonSignal, Unwrap, UnwrapDimension, UnwrapSplit, UnwrapThresh}

   - QMRITools`IVIMTools` with functions:

{BayesianIVIMFit2, BayesianIVIMFit3, ChainSteps, CorrectPar, CorrectParMap, FConvert, FConverti, FilterMaps, FilterSize, FilterType,
 FitConstrains, FixPseudoDiff, FixPseudoDiffSD, FracCorrect, HistogramPar, IVIMCalc, IVIMComponents, IVIMConstrained, IVIMConstrains,
 IVIMCorrectData, IVIMFixed, IVIMFunction, IVIMResiduals, IVIMTensFit, MonitorIVIMCalc, OutputSamples, ThetaConv, ThetaConvi, UpdateStep}
```

- QMRITools`DenoiseTools` with functions:

{AnisoFilterData, AnisoFilterSteps, AnisoFilterTensor, AnisoItterations, AnisoKappa, AnisoKernel, AnisoStepTime, AnisoWeightType, DeNoise, DenoiseCSIdata, DenoiseDynamicSpectraData, DeNoiseIterations, DeNoiseKernel, DeNoiseMonitor, NNDeNoise, NNThreshhold, PCAClipping, PCADeNoise, PCADeNoiseFit, PCAKernel, PCANoiseSigma, PCAOutput, PCATollerance, PCAWeighting, WeightMapCalc}

- QMRITools`CardiacTools` with functions:

{AxesMethod, BackgroundValue, BloodMaskRange, BullPlotMethod, BullseyePlot, CalculateWallMap, CardiacCoordinateSystem, CardiacSegment, CentralAxes, CreateHeart, CutOffMethod, DistanceMeasure, DropSamples, ECVCalc, ExcludeSlices, GetMaskSegmentPoints, GetSegmentLines, GetSegmentSlices, GridLineSpacing, HelixAngleCalc, LCMMethod, LineStep, LinesToSegmentIndex, LineThreshold, MakeECVBloodMask, MakeLineImage, MakeMaskImage, MaskHelix, MaskToLines, MaskWallMap, OutputCheckImage, PlotSegmentMask, PlotSegments, RadialSample, RadialSamples, RowSize, SegmentAngle, SegmentLinesToMask, SegmentsPerSlice, ShowOutliers, ShowPlot, SmoothHelix, StartPoints, StartSlices, TextNumberForm, TextOffset, TextSize, TransmuralPlot}

- QMRITools`RelaxometryTools` with functions:

{CalibrateEPGT2Fit, CreateT2Dictionary, DictB1Range, DictionaryMinSearch, DictT2fRange, DictT2fValue, DictT2IncludeWater, DictT2Range, EPGCalibrate, EPGFatShift, EPGFitFat, EPGFitPoints, EPGMethod, EPGMethodCal, EPGRelaxPars, EPGSignal, EPGSmoothB1, EPGT2Fit, MonitorEPGFit, NonLinearEPGFit, OutputCalibration, ShiftPulseProfile, T1Fit, T1rhoFit, T2Fit, TriExponentialT2Fit, WaterFatShift, WaterFatShiftDirection}

- QMRITools`GradientTools` with functions:

{Bmatrix, BmatrixCalc, BmatrixConv, BmatrixInv, BmatrixRot, BmatrixToggle, CalculateMoments, ConditionCalc, ConditionNumberCalc, ConvertGrads, CorrectBmatrix, CorrectGradients, EnergyCalc, FinalGrads, FindOrder, FlipAxes, FlipGrad, FullGrad, FullSphere, GenerateGradients, GenerateGradientsGUI, GetGradientScanOrder, GetSliceNormal, GetSliceNormalDir, GradBmatrix, GradSeq, GradType, ImportGradObj, OrderSpan, OutputPlot, OutputType, OverPlusCalc, PhaseEncoding, Runs, Steps, StepSizeI, SwitchAxes, UniqueBvalPosition, UnitMulti, UseGrad, VisualOpt}

- QMRITools`TensorTools` with functions:

{ADCCalc, AngleCalc, AngleMap, ColorFAPlot, ConcatenateDiffusionData, Correct, Deriv, Distribution, DriftCorrect, ECalc, EigensysCalc, EigenvalCalc, EigenvecCalc, FACalc, FilterShape, FlipGradientOrientation, FlipTensorOrientation, FullOutput, MeanRes, MonitorCalc, NormalizeSignal, ParameterCalc, Reject, RejectMap, RemoveIsoImages, ResidualCalc, RobustFit, RobustFitParameters, RotationCorrect, SigmaCalc, SortDiffusionData, TensorCalc, TensorCorrect, WestinMeasures}

- QMRITools`JcouplingTools` with functions:

{CenterFrequency, FieldStrength, GetSpinSystem, Linewidth, LinewidthShape, MakeSpinSystem, ReadoutBandwith, ReadoutMethod, ReadoutOutput, ReadoutPhase, ReadoutSamples, SequencePulseAcquire, SequenceSpaceEcho, SequenceSpinEcho, SequenceSteam, SequenceTSE, SimAddPhase, SimEvolve, SimHamiltonian, SimNucleus, SimReadout, SimRotate, SimSignal, SimSpoil, SysTable}

- QMRITools`SpectroTools` with functions:

{ApodizationFunction, ApodizeEcho, ApodizeFid, ApodizePadEcho, ApodizePadFid, ApodizePadSpectra, ApodizeSpectra, BasisSequence, ChangeDwellTimeFid, CompareFidFitPlot, CompareSpectraFitPlot, CorrectTEFid, CorrectTESpec, CSIInterface, ExportSparSdat, FindSpectraPpmShift, FineTuneFit, FitLineShape, FitSpectra, FitSpectraResultTable, GetGyro, GetPpmRange, GetSpectraBasisFunctions, GetTimePpmRange, GetTimeRange, ImportSparSdat, InitializeFit, MakeSpectraResultPlot, PaddingFactor, PadEcho, PadFid, PadSpectra, PhaseCorrectSpectra, PhaseShiftSpectra, PlotCSIData, PlotFid, PlotSpectra, ReadjMRUI, ReadoutType, ShiftSpectra, SparID, SparName, SparOrientation, SpectraBandwith, SpectraFieldStrength, SpectraFitResult, SpectraNucleus, SpectraOutputPlots, SpectraPpmShift, SpectraSamples, SpectraSpacing, SplineSpacingFactor, TimeShiftEcho, TimeShiftFid, TimeShiftFidV}

- QMRITools`ReconstructionTools` with functions:

{AcquisitionMethod, CoilCombine, CoilSamples, CoilWeightedRecon, CoilWeightedReconCSI, DeconvolutionMethod, DeconvolveCSIdata, EchoShiftData, FourierKspace2D, FourierKspace3D, FourierKspaceCSI, FourierRescaleData, FourierShift, FourierShifted, HammingFilter, HammingFilterCSI, HammingFilterData, InverseFourierShift, InverseFourierShifted, MakeHammingFilter, MakeSense, MeanType, NoiseCorrelation, NoiseCovariance, NormalizeOutputSpectra, NormalizeSpectra, OrderKspace, OutputSense, ReadListData, RescaleRecon, SagitalTranspose, SenseRescale, ShiftedFourier, ShiftedInverseFourier, TotalType, WienerRegularization}

- QMRITools`TractographyTools` with functions:

{CombineROIs, FiberAngle, FiberLengthRange, FiberTractography, FilterTracts, FindTensorPermutation, FittingOrder, FitTract,
MakeColor, MaxSeedPoints, MaxTracts, PartTracts, PlotTracts, SeedDensityMap, SelectTractInVol, SelectTractPartInVol, SelectTracts,
SelectTractTroughPlane, SelectTractTroughVol, StepSize, StopThreshhold, TensorFilps, TensorPermutations, TracMonitor, TractDensityMap}

- QMRITools`VisteTools` with functions:

{BinaryType, DatRead, DatWrite, DTItoolExp, DTItoolExpFile, DTItoolExpInd, DTItoolExpTens, ExportVol, ImportDTI, ImportVol, LoadFiberTracts}

- QMRITools`ProcessingTools` with functions:

{B1EqualPower, B1FilterData, B1MapCalc, B1Masking, B1MaxPower, B1Output, B1Scaling, B1ShimMethod, B1Shimming, ColorValue, CombineB1,
CorrectJoinSetMotion, DataTransformation, DatTot, DatTotXLS, ErrorPlot, FiberDensityMap, FiberLengths, FindOutliers, FitData, FitFunction,
FitOutput, GetMaskMeans, Hist, Hist2, InvertDataset, JoinSets, JoinSetSplit, MaskCompartment, MeanMethod, MeanRange, MeanStd, MedCouple,
MotionCorrectSets, NormalizeOverlap, NormalizeSets, NumberTableForm, OutlierIncludeZero, OutlierIterations, OutlierMethod, OutlierOutput,
OutlierRange, OutputSNR, PaddOverlap, ParameterFit, ParameterFit2, ReferenceB1, ReverseData, ReverseSets, RotateData, RotateTensor, Scaling,
SeedDensity, SetupDataStructure, SmartMask, SmartMaskOutput, SmartMethod, SmoothSNR, SNRCalc, SNRMapCalc, SplitSets, Strictness, TableMethod}

- QMRITools`SimulationTools` with functions:

{AddNoise, BlochSeries, CalculateGfactor, CreateDiffData, FatFieldStrength, GESignal, GetPulseProfile, GfactorSimulation, GOutput, GRegularization,
MagnetizationVector, NoiseSize, NoiseType, PlotSimulation, PlotSimulationAngle, PlotSimulationAngleHist, PlotSimulationHist, PlotSimulationVec, Pulses,
ReportFits, Signal, SimAngleParameters, SimParameters, SimulateDualTR, SimulateSliceEPG, SliceRange, SliceRangeSamples, SortVecs, Tensor, TensOutput}

- QMRITools`PhysiologyTools` with functions:

{AlignRespLog, ImportPhyslog, ImportRespirect, OutputMethod, PlotPhyslog, PlotRespiract, SampleStep}

- QMRITools`CoilTools` with functions:

{CoilArrayPlot, CoilSNRCalc, CoilSurfaceVoxelSize, FindCoilPosition, LoadCoilSetup, LoadCoilTarget, MakeCoilLayout, MakeNoisePlots, MakeWeightMask, OutputCoilSurface}

- QMRITools`TaggingTools` with functions:

{AnnalyzeTagging, CalculateDispacementParameters, HistoryWeighting, MonitorTagging}

- QMRITools`ImportTools` with functions:

{BmatrixOut, BvalRead, ConvertDcm, GradRead, ReadBrukerDiff, ReadBvalue, ReadDicom,
ReadDicomDiff, ReadDicomDir, ReadDicomDirDiff, ReadGradients, ReadVoxSize, RotateGradient, ScaleCorrect, ShiftPar}

Removing all local and global definitions of:

- QMRITools`GeneralTools`
- QMRITools`MaskingTools`
- QMRITools`NiftiTools`
- QMRITools`ElastixTools`
- QMRITools`PlottingTools`
- QMRITools`DixonTools`
- QMRITools`IVIMTools`
- QMRITools`DenoiseTools`
- QMRITools`CardiacTools`

```
    - QMRITools`RelaxometryTools`
    - QMRITools`GradientTools`
    - QMRITools`TensorTools`
    - QMRITools`JcouplingTools`
    - QMRITools`SpectroTools`
    - QMRITools`ReconstructionTools`
    - QMRITools`TractographyTools`
    - QMRITools`VisteTools`
    - QMRITools`ProcessingTools`
    - QMRITools`SimulationTools`
    - QMRITools`PhysiologyTools`
    - QMRITools`CoilTools`
    - QMRITools`TaggingTools`
    - QMRITools`ImportTools`
  ----------------------------------------
  Loading and protecting all definitions of:
    - QMRITools`GeneralTools`
    - QMRITools`MaskingTools`
    - QMRITools`NiftiTools`
    - QMRITools`ElastixTools`
    - QMRITools`PlottingTools`
    - QMRITools`DixonTools`
    - QMRITools`IVIMTools`
    - QMRITools`DenoiseTools`
    - QMRITools`CardiacTools`
    - QMRITools`RelaxometryTools`
    - QMRITools`GradientTools`
    - QMRITools`TensorTools`
    - QMRITools`JcouplingTools`
    - QMRITools`SpectroTools`
    - QMRITools`ReconstructionTools`
    - QMRITools`TractographyTools`
    - QMRITools`VisteTools`
```

- QMRITools`ProcessingTools`
- QMRITools`SimulationTools`
- QMRITools`PhysiologyTools`
- QMRITools`CoilTools`
- QMRITools`TaggingTools`
- QMRITools`ImportTools`

In[13]:=
```
Column@QMRIToolsPackages[]
```

Out[13]=

CardiacTools
CoilTools
DenoiseTools
DixonTools
ElastixTools
GeneralTools
GradientTools
ImportTools
IVIMTools
JcouplingTools
MaskingTools
NiftiTools
PhysiologyTools
PlottingTools
ProcessingTools
ReconstructionTools
RelaxometryTools
SimulationTools
SpectroTools
TaggingTools
TensorTools
TractographyTools
VisteTools

In[14]:=
```
QMRIToolsFunctions[100]
```

Functions

ADCCalc
AddNoise
AlignRespLog
AngleCalc
AngleMap
AnisoFilterData
AnisoFilterTensor
AnnalyzeTagging
ApodizeEcho
ApodizeFid
ApodizePadEcho
ApodizePadFid
ApodizePadSpectra
ApodizeSpectra
ApplyCrop
AutoCropData
B1MapCalc
B1Shimming
BayesianIVIMFit2
BayesianIVIMFit3
BlochSeries
Bmatrix
BmatrixCalc
BmatrixConv
BmatrixInv
BmatrixRot
BmatrixToggle
BSplineCurveFit
BullseyePlot
BvalRead
CalculateDispacementParameters
CalculateGfactor
CalculateMoments
CalculateWallMap
CalibrateEPGT2Fit
CardiacCoordinateSystem
CardiacSegment
CentralAxes
ChangeDwellTimeFid
CheckExtension
ClearTemporaryVariables
CoilCombine
CoilSNRCalc
CoilWeightedRecon
CoilWeightedReconCSI
ColorFAPlot
CombineB1
CombineROIs
CompareFidFitPlot
CompareSpectraFitPlot
CompilebleFunctions
EnergyCalc
EPGSignal
EPGT2Fit
ErrorPlot
ExcludeSlices
ExpNoZero
ExportBmat
ExportBval
ExportBvec
ExportNii
ExportSparSdat
ExportVol
ExtractDemoData
ExtractNiiFiles
FACalc
FConvert
FConverti
FiberDensityMap
FiberLengths
FiberTractography
FileSelect
FilterTracts
FinalGrads
FindCoilPosition
FindCrop
FindMaxDimensions
FindOrder
FindOutliers
FindSpectraPpmShift
FindTensorPermutation
FitData
FitSpectra
FitSpectraResultTable
FitTract
FlipGradientOrientation
FlipTensorOrientation
FourierKspace2D
FourierKspace3D
FourierKspaceCSI
FourierRescaleData
FourierShift
FourierShifted
FracCorrect
FullGrad
GenerateGradients
GenerateGradientsGUI
GESignal
GetAssetLocation
GetGradientScanOrder
GetGyro
GetMaskData
InvertDataset
IVIMCalc
IVIMCorrectData
IVIMFunction
IVIMResiduals
JoinSets
LapFilter
LineStep
LinesToSegmentIndex
ListSpherePlot
LLeastSquares
LoadCoilSetup
LoadCoilTarget
LoadFiberTracts
LogNoZero
MADNoZero
MakeCoilLayout
MakeColor
MakeECVBloodMask
MakeGroups
MakeHammingFilter
MakeIntFunction
MakeLineImage
MakeMaskImage
MakeNiiOrentationQ
MakeNiiOrentationS
MakeNoisePlots
MakeSense
MakeSliceImages
MakeSpectraResultPlot
MakeSpinSystem
MakeWeightMask
Mask
MaskData
MaskHelix
MaskToLines
MeanNoZero
MeanRange
MeanSignal
MeanStd
MeanType
MedCouple
MedFilter
MedianNoZero
MemoryUsage
MergeSegmentations
NNDeNoise
NNLeastSquares
NoiseCorrelation
NoiseCovariance
NonLinearEPGFit
ReadDicom
ReadDicomDiff
ReadDicomDir
ReadDicomDirDiff
ReadGradients
ReadjMRUI
ReadListData
ReadTransformParameters
ReadVoxSize
RegisterCardiacData
RegisterData
RegisterDataSplit
RegisterDataTransform
RegisterDataTransformSplit
RegisterDiffusionData
RegisterDiffusionDataSplit
RemoveIsoImages
RemoveMaskOverlaps
RescaleData
RescaleSegmentation
ResidualCalc
ReverseCrop
ReverseDimensions
RMSNoZero
ROIMask
RotateData
RotateDimensionsLeft
RotateDimensionsRight
RotateTensor
RotationMatrixToQuaternion
RotationMatrixToQuaternionVector
SagitalTranspose
SaveImage
SeedDensityMap
SegmentAngle
SegmentLinesToMask
SegmentMask
SegmentsPerSlice
SelectTractInVol
SelectTractPartInVol
SelectTracts
SelectTractTroughPlane
SelectTractTroughVol
SequencePulseAcquire
SequenceSpaceEcho
SequenceSpinEcho
SequenceSteam
SequenceTSE
SetupDataStructure
ShiftedFourier
ShiftedInverseFourier
UniqueBvalPosition
Unwrap
UnwrapSplit
VectorToData
WeightMapCalc
WestinMeasures
$debugElastix

Options

AcquisitionMethod
AffineDirections
AnisoFilterSteps
AnisoItterations
AnisoKappa
AnisoKernel
AnisoStepTime
AnisoWeightType
ApodizationFunction
AspectRatio
AxesLabel
AxesMethod
B1EqualPower
B1FilterData
B1Masking
B1MaxPower
B1Output
B1Scaling
B1ShimMethod
BackgroundValue
BasisSequence
BinaryType
BloodMaskRange
BmatrixOut
BsplineDirections
BsplineSpacing
BullPlotMethod
CenterFrequency
ChainSteps
CoilArrayPlot
CoilSamples
CoilSurfaceVoxelSize
ColorFunction
ColorValue
CompressNii
ConditionCalc
ContourStyle
ConvertDcm
CorrectPar
CropInit
CropOutput
CropPadding
CutOffMethod
DeconvolutionMethod
DeleteOutputFolder
DeleteTempDirectory
DeNoiseIterations
DeNoiseKernel
DeNoiseMonitor
DictB1Range
DictT2fRange
FlipGrad
FullOutput
FullSphere
GetMaskOnly
GetMaskOutput
GOutput
GradType
GRegularization
GridLines
GridLineSpacing
HammingFilter
HistogramBins
HistogramBinsA
HistoryWeighting
ImageLegend
ImageOrientation
ImageResolution
ImageSize
InitializeFit
InterpolationOrder
InterpolationOrderReg
InterpolationOrderRegA
Iterations
IterationsA
IVIMComponents
IVIMConstrained
IVIMConstrains
IVIMFixed
IVIMTensFit
JoinSetSplit
LCMMethod
LineThreshold
Linewidth
LinewidthShape
MagnetizationVector
MakeCheckPlot
MaskClosing
MaskCompartment
MaskComponents
MaskDilation
MaskFiltKernel
MaskSmoothing
MaskWallMap
MaxSeedPoints
MaxTracts
MeanMethod
MeanRes
Method
MethodReg
MethodRegA
MonitorCalc
PCAOutput
PCATollerance
PCAWeighting
PeakNumber
PhaseEncoding
PlotColor
PlotLabel
PlotLabels
PlotRange
PlotSpace
PlotStyle
PositiveZ
PrintTempDirectory
QMRITools`CardiacTools`Private`ReverseDirection
QMRITools`CardiacTools`Private`ReversePoints
QMRITools`ReconstructionTools`Private`ReconFilter
RadialSamples
ReadoutBandwith
ReadoutMethod
ReadoutOutput
ReadoutPhase
ReadoutSamples
ReadoutType
ReferenceB1
RegistrationTarget
Reject
RejectMap
ReportFits
RescaleRecon
Resolutions
ResolutionsA
ReverseData
ReverseSets
RobustFit
RobustFitParameters
RotateGradient
RotateGradients
RotationCorrect
RowSize
Runs
SampleStep
ScaleCorrect
Scaling
SeedDensity
SenseRescale
ShowOutliers
ShowPlot
SimNucleus
SliceRange
SliceRangeSamples
SmartMaskOutput
UseMask
UseSubfolders
UseVersion
VisualOpt
WaterFatShift
WaterFatShiftDirection
WienerRegularization
WindowTitle

In[15]:= **QMRIToolsFunctions["All", 10]**

CardiacTools

Functions

BullseyePlot	GetSegmentSlices	PlotSegments
CalculateWallMap	HelixAngleCalc	RadialSample
CardiacCoordinateSystem	LineStep	SegmentAngle
CardiacSegment	LinesToSegmentIndex	SegmentLinesToMask
CentralAxes	MakeECVBloodMask	SegmentsPerSlice
CreateHeart	MakeLineImage	TransmuralPlot
ECVCalc	MakeMaskImage	
ExcludeSlices	MaskHelix	
GetMaskSegmentPoints	MaskToLines	
GetSegmentLines	PlotSegmentMask	

Options

AxesMethod	LCMMethod	QMRITools`CardiacTools`Private`ReversePoints
BackgroundValue	LineThreshold	RowSize
BloodMaskRange	MaskWallMap	ShowOutliers
BullPlotMethod	Method	ShowPlot
ColorFunction	OutputCheckImage	SmoothHelix
CutOffMethod	PlotLabel	StartPoints
DistanceMeasure	PlotRange	StartSlices
DropSamples	PlotStyle	TextNumberForm
GridLineSpacing	RadialSamples	TextOffset
ImageSize	QMRITools`CardiacTools`Private`ReverseDirection	TextSize

CoilTools

Functions

CoilSNRCalc
FindCoilPosition
LoadCoilSetup
LoadCoilTarget
MakeCoilLayout
MakeNoisePlots
MakeWeightMask

Options

CoilArrayPlot
CoilSurfaceVoxelSize
ColorFunction
ImageSize
OutputCoilSurface
PlotRange

DenoiseTools

Functions

AnisoFilterData
AnisoFilterTensor
DeNoise
DenoiseCSIdata
DenoiseDynamicSpectraData
NNDeNoise
PCADeNoise
PCADeNoiseFit
WeightMapCalc

Options

AnisoFilterSteps
AnisoItterations
AnisoKappa
AnisoKernel
AnisoStepTime
AnisoWeightType
DeNoiseIterations
DeNoiseKernel
DeNoiseMonitor
NNThreshhold
PCAClipping
PCAKernel
PCANoiseSigma
PCAOutput
PCATollerance
PCAWeighting

DixonTools

Functions

DixonReconstruct
DixonToPercent
MakeGroups
SimulateDixonSignal
Unwrap
UnwrapSplit

Options

DixonAmplitudes
DixonBipolar
DixonClipFraction
DixonFieldStrength
DixonFilterInput
DixonFilterOutput
DixonFilterSize
DixonFrequencies
DixonIterations
DixonMaskThreshhold
DixonNucleus
DixonPrecessions
DixonTollerance
MonitorUnwrap
UnwrapDimension
UnwrapThresh

ElastixTools

Functions

ReadTransformParameters
RegisterCardiacData
RegisterData
RegisterDataSplit
RegisterDataTransform
RegisterDataTransformSplit
RegisterDiffusionData
RegisterDiffusionDataSplit
TransformData
$debugElastix

Options

AffineDirections	IterationsA	Resolutions
BsplineDirections	MethodReg	ResolutionsA
BsplineSpacing	MethodRegA	SplitMethod
DeleteTempDirectory	NumberSamples	TempDirectory
FindTransform	NumberSamplesA	UseGPU
HistogramBins	OutputImage	
HistogramBinsA	OutputTransformation	
InterpolationOrderReg	PCAComponents	
InterpolationOrderRegA	PrintTempDirectory	
Iterations	RegistrationTarget	

GeneralTools

Functions

ApplyCrop
AutoCropData
BSplineCurveFit
CheckExtension
ClearTemporaryVariables
CompilebleFunctions
CropData
CutData
Data2DToVector
Data3DToVector
DataToVector
DecomposeAffineMatrix
DecomposeScaleMatrix
DevideNoZero
DynamicPartition
ExpNoZero
ExtractDemoData
FileSelect
FindCrop
FindMaxDimensions
GetAssetLocation
GridData
GridData3D
GyromagneticRatio
LapFilter
LLeastSquares
LogNoZero
MADNoZero
MakeIntFunction
MeanNoZero
MedFilter
MedianNoZero
MemoryUsage
NNLeastSquares
PadToDimensions
QMRIToolsFuncPrint
QMRIToolsFunctions
QMRIToolsPackages
QuaternionToRotationMatrix
QuaternionVectorToRotationMatrix
RescaleData
ReverseCrop
ReverseDimensions
RMSNoZero
RotateDimensionsLeft
RotateDimensionsRight
RotationMatrixToQuaternion
RotationMatrixToQuaternionVector
SaveImage
Squeeze
StdFilter
StichData
SumOfSquares
TensMat
TensVec
VectorToData

Options

CropInit
CropOutput
CropPadding
FileType
ImageResolution
ImageSize
InterpolationOrder
OutputWeights
Padding
PadDirection
PadValue
SplineDegree
SplineKnotsNumber
SplineRegularization
WindowTitle

GradientTools

Functions

Bmatrix
BmatrixCalc
BmatrixConv
BmatrixInv
BmatrixRot
BmatrixToggle
CalculateMoments
ConditionNumberCalc
ConvertGrads
CorrectBmatrix
CorrectGradients
EnergyCalc
FinalGrads
FindOrder
FullGrad
GenerateGradients
GenerateGradientsGUI
GetGradientScanOrder
GetSliceNormal
GetSliceNormalDir
GradBmatrix
GradSeq
ImportGradObj
OverPlusCalc
UniqueBvalPosition

Options

ConditionCalc
FlipAxes
FlipGrad
FullSphere
GradType
Method
MethodReg
OrderSpan
OutputPlot
OutputType
PhaseEncoding
Runs
Steps
StepSizeI
SwitchAxes
UnitMulti
UseGrad
VisualOpt

ImportTools

Functions

BvalRead
GradRead
ReadBrukerDiff
ReadBvalue
ReadDicom
ReadDicomDiff
ReadDicomDir
ReadDicomDirDiff
ReadGradients
ReadVoxSize
ShiftPar

Options

BmatrixOut
ConvertDcm
RotateGradient
ScaleCorrect

IVIMTools

Functions

BayesianIVIMFit2
BayesianIVIMFit3
CorrectParMap
FConvert
FConverti
FracCorrect
HistogramPar
IVIMCalc
IVIMCorrectData
IVIMFunction
IVIMResiduals
ThetaConv
ThetaConvi

Options

ChainSteps
CorrectPar
FilterMaps
FilterSize
FilterType
FitConstrains
FixPseudoDiff
FixPseudoDiffSD
IVIMComponents
IVIMConstrained
IVIMConstrains
IVIMFixed
IVIMTensFit
Method
MonitorIVIMCalc
OutputSamples
Parallelize
UpdateStep

JcouplingTools

Functions

GetSpinSystem
MakeSpinSystem
SequencePulseAcquire
SequenceSpaceEcho
SequenceSpinEcho
SequenceSteam
SequenceTSE
SimAddPhase
SimEvolve
SimHamiltonian
SimReadout
SimRotate
SimSignal
SimSpoil
SysTable

Options

CenterFrequency
FieldStrength
Linewidth
LinewidthShape
ReadoutBandwith
ReadoutMethod
ReadoutOutput
ReadoutPhase
ReadoutSamples
SimNucleus

MaskingTools

Functions

GetMaskData
GrowMask
HomoginizeData
Mask
MaskData
MeanSignal
MergeSegmentations
NormalizeData
NormalizeMeanData
RemoveMaskOverlaps
RescaleSegmentation
ROIMask
SegmentMask
SmoothMask
SmoothSegmentation
SplitSegmentations

Options

GetMaskOnly
GetMaskOutput
MaskClosing
MaskComponents
MaskDilation
MaskFiltKernel
MaskSmoothing
UseMask

NiftiTools

Functions

CompressNiiFiles
CorrectNiiOrientation
DcmToNii
ExportBmat
ExportBval
ExportBvec
ExportNii
ExtractNiiFiles
GetNiiOrientation
ImportBmat
ImportBval
ImportBvalvec
ImportBvec
ImportExploreDTItens
ImportNii
ImportNiiDiff
ImportNiiDix
ImportNiiT1
ImportNiiT2
MakeNiiOrentationQ
MakeNiiOrentationS

Options

CompressNii
DeleteOutputFolder
FlipBvec
Method
NiiDataType
NiiLegacy
NiiMethod
NiiOffset
NiiScaling
NiiSliceCode
PositiveZ
RotateGradients
UseSubfolders
UseVersion

PhysiologyTools

Functions

AlignRespLog
ImportPhyslog
ImportRespirect
PlotPhyslog
PlotRespiract

Options

OutputMethod
SampleStep

PlottingTools

Functions

GetSliceData
GetSlicePositions
GradientPlot
ListSpherePlot
MakeSliceImages
PlotContour
PlotCorrection
PlotData
PlotData3D
PlotDefGrid
PlotDuty
PlotIVIM
PlotMoments
PlotSequence

Options

ColorFunction
ContourStyle
DropSlices
ImageLegend
ImageOrientation
ImageSize
MakeCheckPlot
Method
NormalizeIVIM
PeakNumber
PlotColor
PlotRange
PlotSpace
PositiveZ
SphereColor
SphereSize

ProcessingTools

Functions

B1MapCalc	FindOutliers	NumberTableForm
B1Shimming	FitData	ParameterFit
CombineB1	GetMaskMeans	ParameterFit2
CorrectJoinSetMotion	Hist	RotateData
DataTransformation	Hist2	RotateTensor
DatTot	InvertDataset	SetupDataStructure
DatTotXLS	JoinSets	SmartMask
ErrorPlot	MeanRange	SNRCalc
FiberDensityMap	MeanStd	SNRMapCalc
FiberLengths	MedCouple	SplitSets

Options

AxesLabel	FitOutput	OutlierIncludeZero	ReverseSets	TableHeadings
B1EqualPower	ImageSize	OutlierIterations	Scaling	TableMethod
B1FilterData	InterpolationOrder	OutlierMethod	SeedDensity	TableSpacing
B1Masking	JoinSetSplit	OutlierOutput	SmartMaskOutput	
B1MaxPower	MaskCompartment	OutlierRange	SmartMethod	
B1Output	MeanMethod	OutputSNR	SmoothSNR	
B1Scaling	Method	PaddOverlap	Strictness	
B1ShimMethod	MotionCorrectSets	PlotLabel	TableAlignments	
ColorValue	NormalizeOverlap	ReferenceB1	TableDepth	
FitFunction	NormalizeSets	ReverseData	TableDirections	

ReconstructionTools

Functions

CoilCombine
CoilWeightedRecon
CoilWeightedReconCSI
DeconvolveCSIdata
FourierKspace2D
FourierKspace3D
FourierKspaceCSI
FourierRescaleData
FourierShift
FourierShifted
HammingFilterCSI
HammingFilterData
InverseFourierShift
InverseFourierShifted
MakeHammingFilter
MakeSense
MeanType
NoiseCorrelation
NoiseCovariance
NormalizeSpectra
OrderKspace
ReadListData
SagitalTranspose
ShiftedFourier
ShiftedInverseFourier
TotalType

Options

AcquisitionMethod
CoilSamples
DeconvolutionMethod
EchoShiftData
HammingFilter
Method
NormalizeOutputSpectra
OutputSense
QMRITools`ReconstructionTools`Private`ReconFilter
RescaleRecon
SenseRescale
WienerRegularization

RelaxometryTools

Functions

CalibrateEPGT2Fit
CreateT2Dictionary
DictionaryMinSearch
EPGSignal
EPGT2Fit
NonLinearEPGFit
ShiftPulseProfile
T1Fit
T1rhoFit
T2Fit
TriExponentialT2Fit

Options

DictB1Range
DictT2fRange
DictT2fValue
DictT2IncludeWater
DictT2Range
EPGCalibrate
EPGFatShift
EPGFitFat
EPGFitPoints
EPGMethod
EPGMethodCal
EPGRelaxPars
EPGSmoothB1
Method
MonitorEPGFit
OutputCalibration
WaterFatShift
WaterFatShiftDirection

SimulationTools

Functions

AddNoise
BlochSeries
CalculateGfactor
CreateDiffData
GESignal
GetPulseProfile
GfactorSimulation
PlotSimulation
PlotSimulationAngle
PlotSimulationAngleHist
PlotSimulationHist
PlotSimulationVec
Pulses
Signal
SimAngleParameters
SimParameters
SimulateDualTR
SimulateSliceEPG
Tensor

Options

FatFieldStrength
GOutput
GRegularization
MagnetizationVector
NoiseSize
NoiseType
PlotRange
Reject
ReportFits
SliceRange
SliceRangeSamples
SortVecs
TensOutput

SpectroTools

Functions

ApodizeEcho
ApodizeFid
ApodizePadEcho
ApodizePadFid
ApodizePadSpectra
ApodizeSpectra
ChangeDwellTimeFid
CompareFidFitPlot
CompareSpectraFitPlot
CorrectTEFid
CorrectTESpec
CSIInterface
ExportSparSdat
FindSpectraPpmShift
FitSpectra
FitSpectraResultTable
GetGyro
GetPpmRange
GetSpectraBasisFunctions
GetTimePpmRange
GetTimeRange
ImportSparSdat
MakeSpectraResultPlot
PadEcho
PadFid
PadSpectra
PhaseCorrectSpectra
PhaseShiftSpectra
PlotCSIData
PlotFid
PlotSpectra
ReadjMRUI
ShiftSpectra
SpectraFitResult
TimeShiftEcho
TimeShiftFid
TimeShiftFidV

Options

ApodizationFunction
AspectRatio
BasisSequence
CenterFrequency
FineTuneFit
FitLineShape
GridLines
GridLineSpacing
ImageSize
InitializeFit
Method
PaddingFactor
PlotColor
PlotLabel
PlotLabels
PlotRange
ReadoutType
SparID
SparName
SparOrientation
SpectraBandwith
SpectraFieldStrength
SpectraNucleus
SpectraOutputPlots
SpectraPpmShift
SpectraSamples
SpectraSpacing
SplineSpacingFactor

TaggingTools

Functions

AnnalyzeTagging
CalculateDispacementParameters

Options

HistoryWeighting
MonitorTagging

TensorTools

Functions

ADCCalc
AngleCalc
AngleMap
ColorFAPlot
ConcatenateDiffusionData
Correct
Deriv
DriftCorrect
ECalc
EigensysCalc
EigenvalCalc
EigenvecCalc
FACalc
FlipGradientOrientation
FlipTensorOrientation
ParameterCalc
RemoveIsoImages
ResidualCalc
SigmaCalc
SortDiffusionData
TensorCalc
TensorCorrect
WestinMeasures

Options

Distribution
FilterShape
FullOutput
MeanRes
Method
MonitorCalc
NormalizeSignal
Parallelize
Reject
RejectMap
RobustFit
RobustFitParameters
RotationCorrect
UseMask

TractographyTools

Functions

CombineROIs
FiberTractography
FilterTracts
FindTensorPermutation
FitTract
MakeColor
PartTracts
PlotTracts
SeedDensityMap
SelectTractInVol
SelectTractPartInVol
SelectTracts
SelectTractTroughPlane
SelectTractTroughVol
TractDensityMap

Options

FiberAngle
FiberLengthRange
FittingOrder
ImageSize
InterpolationOrder
MaxSeedPoints
MaxTracts
Method
StepSize
StopThreshhold
TensorFilps
TensorPermutations
TracMonitor

Viste Tools

Functions

DatRead
DatWrite
DTItoolExp
DTItoolExpFile
DTItoolExpInd
DTItoolExpTens
ExportVol
ImportDTI
ImportVol
LoadFiberTracts

Options

BinaryType

In[16]:= **QMRIToolsFuncPrint[]**

CardiacTools

Functions

Symbol

BullseyePlot[data, segmask] generates a AHA–17 segement bullseye plot.

BullseyePlot[list] generates a AHA–17 segement bullseye plot of the lists (which needs to have 17 values) provide.

data is a 3D volume used for the plot.

segmask is the AHA–17 segmentation resulting form the CardiacSegment function when AHA17 is selected.

Output is a bullseye plot or a plotwindow, depending on the Method which can be "Dynamic" else it will be static.

BullseyePlot[] is based on DOI: 10.1161/hc0402.102975.

Documentation Local »

Default Definitions SyntaxInformation[BullseyePlot] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › TextOffset → 0.5 … (7 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`BullseyePlot

Symbol

CalculateWallMap[mask,vox] calculates the wall distance map and the wall derivative.

Output is {wallmap, wallDerivative}.

Documentation Local »

Default Definitions SyntaxInformation[CalculateWallMap] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options {ShowPlot → True, MaskWallMap → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`CalculateWallMap

Symbol

CardiacCoordinateSystem[mask, vox] creates the cardiac coordinate system within the mask and is used in HelixAngleCalc.

Output is a set of vectors {radvecn, norvecc, cirvec}, being the radial, normal and circular axes of each voxel respectivley.

If the option showPlot is true the output is {{radvecn, norvecc, cirvec}, plots}.

Documentation Local »

Default Definitions SyntaxInformation[CardiacCoordinateSystem] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options {ShowPlot → False, LCMMethod → WallMap, AxesMethod → Quadratic}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`CardiacCoordinateSystem

Symbol

CardiacSegment[data, mask, off] allows to segment the heart in 1, 4, 6 or AHA−17 segements for each slice 360 radial samples are generated.

data is a background image on which all overlays are projected.

mask is the mask of the left ventricle (same as used for CentralAxes) and defines the area in which the data is sampled.

off is the centerpoints generated by CentralAxes.

Output is {segmask, segang, {points, slices}}.

Documentation Local »

Default Definitions SyntaxInformation[CardiacSegment] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › StartPoints → Default … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`CardiacSegment

Symbol

CentralAxes[mask, vox] calculates the center of the lumen from a mask of the left ventricle. vox is the voxels size, {slice, x, y}.

CentralAxes[mask, maskp, vox] allows for fancy visualization of the other structures using maskp.

Output is {centerpoints, normalvecs, inout} or {centerpoints, normalvecs, inout, fit}.

Documentation Local »

Default Definitions SyntaxInformation[CentralAxes] = {ArgumentsPattern → {_, _, _., OptionsPattern[]}}

Options {ShowPlot → False, RowSize → Automatic, AxesMethod → Cubic}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`CentralAxes

Symbol

CreateHeart[] creates a simulated left ventricle shape.

CreateHeart[pars] creates a simulated left ventricle shape with predifined parameters pars.

Output is the heart shape, the voxel size and the parameters needed to generate the heart, {mask, vox, pars}.

Documentation Local »

Default Definitions SyntaxInformation[CreateHeart] = {ArgumentsPattern → {_.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`CreateHeart

Symbol

ECVCalc[T1pre, T1post, hema] calculates the ECVmap using MakeECVBloodMask.

ECVCalc[T1pre, T1post, bloodMask, hema] calculates the ECVmap using bloodMask.

The T1pre and T1post maps are assuemed to be in ms.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`ECVCalc

Symbol

ExcludeSlices[data] excludes slices that do not look like the others based on various distance measures.

Output is an array with 1 or 0 with the dimensiosn {slices, diff dirs}.

Documentation Local »

Default Definitions SyntaxInformation[ExcludeSlices] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {CutOffMethod → Auto, DistanceMeasure → 5, ShowOutliers → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`ExcludeSlices

Symbol

GetMaskSegmentPoints[mask].

Documentation Local »

Default Definitions SyntaxInformation[GetMaskSegmentPoints] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`GetMaskSegmentPoints

Symbol

GetSegmentLines[lines, lineIndex, seg].

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`GetSegmentLines

Symbol

GetSegmentSlices[mask_?ArrayQ]

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`GetSegmentSlices

Symbol

HelixAngleCalc[eigenvectors, mask, vox] calculates the helix angle matrix of cardiac data using only a left ventricle mask.

HelixAngleCalc[eigenvectors, mask, maskp, vox] calculates the helix angle matrix of cardiac data using only a left ventricle mask, and a maskp for visualization.

HelixAngleCalc[eigenvectors, mask, centerpoint, vec, inout, vox] calculates the helix angle matrix of cardiac data using only a left ventricle mask.

HelixAngleCalc[eigenvectors, mask, maskp, centerpoint, vec, inout, vox] calculates the helix angle matrix of cardiac data using a left vantricle mask and a maskp for visualization.

eigenvectors are the tensor eigenvectors calculated with EigenvecCalc.

mask is a mask of the left ventricle.

maskp is a mask used for visualization.

vox is the voxels size, {slice, x, y}.

The folowing values are calculated automaticlay Using CentralAxes but can also be provided as an input.

centerpoint is the center of each slice calculated with CentralAxes.

inout is the inner and outer radius calcualted with CentralAxes.

vec is the vector describin the central axes of the heart, calculated with CentralAxes.

Output is het fiber angle matrix FAM = {9, slice, x, y} or {FAM, plot}.

The angles are in degrees.

HelixAngleCalc[] is based on DOI: 10.1186/1532–429X–17–S1–P15.

Documentation Local »

Default Definitions SyntaxInformation[HelixAngleCalc] = {ArgumentsPattern → {_, _, _, _., OptionsPattern[]}}

Options {ShowPlot → True, LCMMethod → WallMap, AxesMethod → Quadratic}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`HelixAngleCalc

Symbol

LineStep is an option for CardiacSegment.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`LineStep

Symbol

LinesToSegmentIndex[lines, pts, seg, OptionsPattern].

Documentation Local »

Options {QMRITools`CardiacTools`Private`ReversePoints → True, QMRITools`CardiacTools`Private`ReverseDirection → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`LinesToSegmentIndex

Symbol

MakeECVBloodMask[T1pre, T1post] makes a bloodpool mask based on the T1pre and T1post images. It assumes that the hart is cropped with the blood in the center.

The T1pre and T1post maps are assuemed to be in ms.

Documentation Local »

Default Definitions SyntaxInformation[MakeECVBloodMask] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options {BloodMaskRange → {1400, {0, 700}}, OutputCheckImage → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MakeECVBloodMask

Symbol

MakeLineImage[back, segLines, pts].

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MakeLineImage

Symbol

MakeMaskImage[back, mask].

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MakeMaskImage

Symbol

MaskHelix[helix, mask] masks helix angle data, sets the background to –100 and allows for Median filter of the helix mask.

helix can be a singel map or the FAM.

Output is the masked helix angle data.

Documentation Local »

Default Definitions SyntaxInformation[MaskHelix] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options {BackgroundValue → –100, SmoothHelix → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MaskHelix

Symbol

MaskToLines[mask, vox]

MaskToLines[mask, wall, cent].

Documentation Local »

Options LineThreshold → 0.3

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MaskToLines

Symbol

PlotSegmentMask[mask, segmask, vox] plots the mask segements created by CardiacSegment.

mask is a mask the left ventricle that was used in the CardiacSegment.

segmask is the output of CardiacSegemnt.

vox is the voxels size, {slice, x, y}.

Output is a plot window.

Documentation Local »

Default Definitions SyntaxInformation[PlotSegmentMask] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`PlotSegmentMask

Symbol

PlotSegments[mask, data, segang] shows how the heart wil be sampled by RadialSample.

mask is a mask the left ventricle that was used in the CardiacSegment.

function and the segang is the output of the cardaic SegmentFunction.

Output is a plot window.

Documentation Local »

Default Definitions SyntaxInformation[PlotSegments] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options RadialSamples → 10

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`PlotSegments

Symbol

RadialSample[mask, data, segang] radialy samples the provided parametermap data.

The mask should be a mask of the left ventricle that was used in the CardiacSegment.

segang is the output of the cardaic SegmentFunction.

Output is {points, vals} which are orderd as indicated by the user.

Documentation Local »

Default Definitions SyntaxInformation[RadialSample] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options {RadialSamples → 10, DropSamples → 0}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`RadialSample

Symbol

SegmentAngle is an option for CardiacSegment.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`SegmentAngle

Symbol

SegmentLinesToMask[smsk, segLines].

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`SegmentLinesToMask

Symbol

SegmentsPerSlice[points, segmi].

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`SegmentsPerSlice

Symbol

TransmuralPlot[data] plots transmural profiles of the data which are created by RadialSample.

data can be a single profile or a list of profiles. In the second case the mean and standardeviations are plotted.

Output is a plot of the transmural profile.

Documentation Local »

Default Definitions SyntaxInformation[TransmuralPlot] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options › GridLineSpacing → 10 … (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`TransmuralPlot

Options

Symbol

AxesMethod is an option for HelixAngleCalc and CentralAxes and CardiacCoordinateSystem. Can be "Linear", "Quadratic", "Cubic".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`AxesMethod

Symbol

BackgroundValue is an option for MaskHelix. Sets the backgroud value (default is –100).

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`BackgroundValue

Symbol

BloodMaskRange is an option for MakeECVBloodMask.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`BloodMaskRange

Symbol

BullPlotMethod is an option for BullseyePlot. Can be "Dynamic" of "Normal".

"Dynamic" allows to change plotting parameters in Manipulation window.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`BullPlotMethod

Symbol

ColorFunction is an option for graphics functions that specifies a function to apply to determine colors of elements.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ColorFunction

Symbol

CutOffMethod is an option for ExcludeSlices. Default value is "Auto" or it can be a fixed percentage (value between 0 and .5).

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`CutOffMethod

Symbol

DistanceMeasure is an option for ExcludeSlices. Defaul value is 5. (1 ManhattanDistance, 2 SquaredEuclideanDistance, 3 EuclideanDistance, 4 Correlation, 5 SpearmanRho.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`DistanceMeasure

Symbol

DropSamples is an option for RadialSample and PlotSegments. Defines how manny samples are droped form star and end. Can be an number or set (strat, end) of numbers.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`DropSamples

Symbol

GridLineSpacing is an option of TransmuralPlot. It defines the spacing of the gridlines.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`GridLineSpacing

Symbol

ImageSize is an option that specifies the overall size of an image to display for an object.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ImageSize

Symbol

LCMMethod is an option for HelixAngleCalc and LMCSytemCalc. Can be "CentralAxes" or "WallMap".

"CentralAxes" uses wall distance calculation using projection of the centarl axes and circular approximation of the ventricle. This method is fairly fast and uses CentralAxes internaly.

"WallMap" uses wall distance interpolation and subsequential gradient calculation. Can take long for high res datasets but is most accurate. Uses CalculateWallMap internaly.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`LCMMethod

Symbol

LineThreshold is an option for CardiacSegment. Can be number between 0 and 1. Increasing the value will decrease the amount of wall sampled.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`LineThreshold

Symbol

MaskWallMap is an option for CalculateWallMap. if True or False.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MaskWallMap

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

OutputCheckImage is an option for MakeECVBloodMask.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`OutputCheckImage

Symbol

PlotLabel is an option for graphics functions that specifies an overall label for a plot.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`PlotLabel

Symbol

PlotRange is an option for graphics functions that specifies what range of coordinates to include in a plot.

Documentation Local » | Web »

Attributes {Protected, ReadProtected}

Full Name System`PlotRange

Symbol

PlotStyle is an option for plotting and related functions that specifies styles in which objects are to be drawn.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`PlotStyle

Symbol

RadialSamples is an option for RadialSample and PlotSegments. Defines how manny transmural samples are taken.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`RadialSamples

Symbol

QMRITools`CardiacTools`Private`ReverseDirection

Full Name QMRITools`CardiacTools`Private`ReverseDirection

Symbol

QMRITools`CardiacTools`Private`ReversePoints

Full Name QMRITools`CardiacTools`Private`ReversePoints

Symbol

RowSize is an option for CentralAxes. defines the number or images per showing the segmentation.

Can be "Automatic" of an integer.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`RowSize

Symbol

ShowOutliers is an option for ExcludeSlices.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`ShowOutliers

Symbol

ShowPlot is an option for CentralAxes, HelixAngleCalc and CardiacCoordinateSystem. True shows the fit of the central axes and outpu the plot as extra output.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`ShowPlot

Symbol

SmoothHelix is an option for MaskHelix, sets the kernelsize for the MedianFilter.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`SmoothHelix

Symbol

StartPoints is an option for CardiacSegment. Value is "Default" or the point list given by CardiacSegment.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`StartPoints

Symbol

StartSlices is an option for CardiacSegment. Value is "Default" or the list given by CardiacSegment.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`StartSlices

Symbol

TextNumberForm is an option for BullseyePlot. Specifies how many number and decimals to use like in NumberForm.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`TextNumberForm

Symbol

TextOffset is an option for BullseyePlot. Determines where the text is placed, can be 0 to 1.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`TextOffset

Symbol

TextSize is an option for BullseyePlot. Determines the text size.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`TextSize

CoilTools

Functions

Symbol

CoilSNRCalc[coils, noise] calculates the sensitivity weighted snr of multiple coil elements using magnitude signal and noise.

Output is {data, noise, sos, snr, sigmap, weights}.

Documentation Local »

Default Definitions SyntaxInformation[CoilSNRCalc] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`CoilSNRCalc

Symbol

FindCoilPosition[weights] finds the coil posision by locating the highest intensity location in the coil weight map, which can be obtianed by LoadCoilSetup or SumOfSquares.

Internally it uses MakeWeightMask to remove the noise of the weightmasks.

FindCoilPosition[weights, mask] limits the search region to the provided mask.

Documentation Local »

Default Definitions SyntaxInformation[FindCoilPosition] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options {OutputCoilSurface → False, CoilSurfaceVoxelSize → {1, 1, 1}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`FindCoilPosition

Symbol

LoadCoilSetup[file] load a very specific type of coil experiment, a dynmic scan with a setup of which the second dynamic is a noise measurement.

The input file is the Nii file that conatins the individualy reconstruted coil images and the noise data.

Internaly it uses CoilSNRCalc and SumOfSquares.

Output is the coil data with coil noise data and snrmap based on the SumOfSquares addition, the SOS reconstruction and the SOS weights.

{dataC, noiseC, sosC, snrC, sigmapC, weights, vox}.

Documentation Local »

Default Definitions SyntaxInformation[LoadCoilSetup] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`LoadCoilSetup

Symbol

LoadCoilTarget[file] loads a very specific typ of experiment, a dynamic scan with with the second dynmaic is a noise measuremnt.

The input file is the Nii file that conatins the scanner reconstruction and the noise data.

Internaly it uses SNRMapCalc,

Output is the reconstructed data with noise data and snrMap {dataC, noiseC, sosC, snrC, sigmapC, weights, vox}.

Documentation Local »

Default Definitions SyntaxInformation[LoadCoilTarget] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`LoadCoilTarget

Symbol

MakeCoilLayout[{name, size, number}] makes a coil grid with label name, partioned in size rows and with label number.

MakeCoilLayout[{name, size, number}, val] makes a coil grid with label name, partioned in size rows and with label the val at location number.

MakeCoilLayout[{coils..}] same but for multile coils grids. Each coil grid is defined as {name, size, number}.

MakeCoilLayout[{coils..}, val] savem but for multiple coil grids.

Documentation Local »

Options › PlotRange → Automatic … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`MakeCoilLayout

Symbol

MakeNoisePlots[noise] returns a grid of plots of the noise per channel

MakeNoisePlots[noise, {met, prt}] met can be "Grid" with prt a number or Automatic. Else all plots will be returend as a list of plots.

MakeNoisePlots[noise, {met, prt}, sub] sub defines how much the noise is subsampled, default is 40 (every 40th sample is used in plot).

Documentation Local »

Default Definitions SyntaxInformation[MakeNoisePlots] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`MakeNoisePlots

Symbol

MakeWeightMask[weights] creates a mask of homogeneous regions of weightmaps removing the noise.

Documentation Local »

Default Definitions SyntaxInformation[MakeWeightMask] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`MakeWeightMask

Options

Symbol

CoilArrayPlot is an option for MakeCoilLayout. If True and values are provided it makes an arrayplot of the coil layouts.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`CoilArrayPlot

Symbol

CoilSurfaceVoxelSize is an option for FindCoilPosition. Specifies the voxel size used for OutputCoilSurface.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`CoilSurfaceVoxelSize

Symbol

ColorFunction is an option for graphics functions that specifies a function to apply to determine colors of elements.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ColorFunction

Symbol

ImageSize is an option that specifies the overall size of an image to display for an object.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ImageSize

Symbol

OutputCoilSurface is an option for FindCoilPosition. If set true it will also output a SurfacePlot of the coil location volume.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CoilTools`OutputCoilSurface

Symbol

PlotRange is an option for graphics functions that specifies what range of coordinates to include in a plot.

Documentation Local » | Web »

Attributes {Protected, ReadProtected}

Full Name System`PlotRange

DenoiseTools

Functions

Symbol

AnisoFilterData[data] Filter the diffusion tensor data using an anisotropic filter based on the strucure tensor of the data.

Output is the smoothed data.

AnisoFilterData[] is based on DOI: 10.1016/j.jbiomech.2021.110540.

Documentation Local »

Default Definitions SyntaxInformation[AnisoFilterData] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {AnisoStepTime → 0.35, AnisoItterations → 3, AnisoKernel → {0.05, 0.1}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoFilterData

Symbol

AnisoFilterTensor[tens, diffdata] Filter the tensor tens using an anisotropic diffusion filter (Perona–Malik).

It uses the diffusion weighted data diffdata to find edges that are not visible in the tensor.

Edge weights based on the diffusion data are averaged over all normalized diffusion direction.

Output is the smoothed tensor.

AnisoFilterTensor[] is based on DOI: 10.1109/ISBI.2006.1624856.

Documentation Local »

Default Definitions SyntaxInformation[AnisoFilterTensor] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options › AnisoWeightType → 2 … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoFilterTensor

Symbol

DeNoise[data,sigma,filtersize] removes Rician noise with standard deviation "sigma" from the given dataset using a kernel with size "filtersize" a gaussian kernel.

DeNoise[data,sigma,filtersize, Kernel–>"kerneltype"] removes Rician noise with standard deviation "sigma" from the given dataset using a kernel with size "filtersize" and type "kerneltype".

Output is data denoised.

DeNoise[] is based on DOI: 10.1109/TMI.2008.920609.

Documentation Local »

Default Definitions SyntaxInformation[DeNoise] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options {DeNoiseKernel → Gaussian, DeNoiseMonitor → False, DeNoiseIterations → 1}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`DeNoise

Symbol

DenoiseCSIdata[spectra] perfroms PCA denoising of the complex values spectra, data has to be 3D and the spectral dimensions is last, {x,y,z,spectra}.

Documentation Local »

Default Definitions SyntaxInformation[DenoiseCSIdata] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {PCAKernel → 5, PCANoiseSigma → Corners}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`DenoiseCSIdata

Symbol

DenoiseDynamicSpectraData[spectra] perfroms PCA denoising of the complex values spectra, The data is given as a list of dynamicly acquired spectra {dynamic ,spectra}.

Documentation Local »

Default Definitions SyntaxInformation[DenoiseDynamicSpectraData] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`DenoiseDynamicSpectraData

Symbol

NNDeNoise[data] removes rician noise from the data using self supravized neural net.

NNDeNoise[data, mask] removes rician noise from the data with PCA using self supravized neural net withing the mask.

PCADeNoise[] is based on DOI:10.48550/arXiv.2011.01355.

Documentation Local »

Default Definitions SyntaxInformation[NNDeNoise] = {ArgumentsPattern → {_, _., _., OptionsPattern[]}}

Options NNThreshhold → 2

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`NNDeNoise

Symbol

PCADeNoise[data] removes rician noise from the data with PCA.

PCADeNoise[data, mask] removes rician noise from the data with PCA only withing the mask.

PCADeNoise[data, mask, sig] removes rician noise from the data with PCA only withing the mask using sig as prior knowledge or fixed value.

Output is de {data denoise, sigma map} by default if PCAOutput is Full then fitted {data

dnoise , {sigma fit, average sigma}, {number components, number of fitted voxesl, number of max fits}, total fit –time per 500 ittt}.

PCADeNoise[] is based on DOI: 10.1016/j.neuroimage.2016.08.016 and 10.1002/mrm.26059.

Documentation Local »

Default Definitions SyntaxInformation[PCADeNoise] = {ArgumentsPattern → {_, _., _., OptionsPattern[]}}

Options › PCAKernel → 5 … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCADeNoise

Symbol

PCADeNoiseFit[data] fits the marchencopasteur distribution to the PCA of the data using grid search.

PCADeNoiseFit[data, sig] fits the marchencopasteur distribution to the PCA of the data using sig as start value or fixed value using grid search.

Output is {simga, number of noise comp, and denoised matrix}.

Documentation Local »

Default Definitions SyntaxInformation[PCADeNoiseFit] = {ArgumentsPattern → {_, _., _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCADeNoiseFit

Symbol

WeightMapCalc[diffdata] calculates a weight map which is used in AnisoFilterTensor.

Output is a weight map of the diffdata which is high in isotropic regions and low at edges.

Documentation Local »

Default Definitions SyntaxInformation[WeightMapCalc] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {AnisoWeightType → 2, AnisoKappa → 10.}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`WeightMapCalc

Options

Symbol

AnisoFilterSteps is an option for AnisoFilterTensor and defines the amoutn of diffusin steps taken. Higher is more smoothing.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoFilterSteps

Symbol

AnisoItterations is an options for AnisoFilterData. It specifies the amount of denoising itterations.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoItterations

Symbol

AnisoKappa is an option for AnisoFilterTensor and WeightMapCalc and defines the weighting strenght, all data is normalize to 100 before filetering.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoKappa

Symbol

AnisoKernel is an options for AnisoFilterData. It defines the kernel size.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoKernel

Symbol

AnisoStepTime is an option for AnisoFilterTensor and defines the diffusion time, when small more step are needed.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoStepTime

Symbol

AnisoWeightType is an option for AnisoFilterTensor and WeightMapCalc and defines the weighting, eigher 1, the exponent of (–g/kappa) or 2, 1/(1+g/kappa).

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoWeightType

Symbol

DeNoiseIterations is and option for DeNoise. Specifies the number of the denoising iterations.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`DeNoiseIterations

Symbol

DeNoiseKernel is and option for DeNoise. Values can be "Disk", "Box" or "Gaussian".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`DeNoiseKernel

Symbol

DeNoiseMonitor is and option for DeNoise. Monitor the denoising progres.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`DeNoiseMonitor

Symbol

NNThreshhold is an options for NNDeNoise and specifies the automated back ground masking value.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`NNThreshhold

Symbol

PCAClipping is an option of PCADeNoise and can be True of False. If True the output is clipped between 0 and the max absolute value of the input data.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCAClipping

Symbol

PCAKernel is an option of PCADeNoise. It sets the kernel size.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCAKernel

Symbol

PCANoiseSigma is an option of DenoiseCSIdata and can be "Corners" or "Automatic".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCANoiseSigma

Symbol

PCAOutput is an option of PCADeNoise. If output is full the output is {datao, {output[[1]], sigmat}, {output[[2]], output[[3]], j}, timetot}.

Else the output is {datao, sigmat}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCAOutput

Symbol

PCATollerance is an option of PCADeNoise and shuld be an integer > 0. Default value is 0. When increased the denoise method removes less noise.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCATollerance

Symbol

PCAWeighting is an option of PCADeNoise and can be True of False. Default

value is False. When True the weights of the per voxel result are calculated based on the number of non noise components.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCAWeighting

DixonTools

Functions

Symbol

DixonReconstruct[real, imag, echo] reconstruxt Dixon data with initital guess b0 = 0 and T2star = 0.

DixonReconstruct[real, imag, echo, b0] reconstructs Dixon data with intitial guess T2star = 0.

DixonReconstruct[real, imag, echo, b0, t2] reconstructs Dixon data.

real is the real data in radials.

imag is the imaginary data in radians.

B0 can be estimated from two phase images using Unwrap.

T2 can be estimated from multiple echos using T2fit.

Output is {{watF,fatF},{watSig,fatSig},{inphase,outphase},{B0,T2star},itterations}.

The fractions are between 0 and 1, the B0 field map is in Hz and the T2start map is in ms.

DixonReconstruct[] is based on DOI: 10.1002/mrm.20624 and 10.1002/mrm.21737 (10.1002/nbm.3766).

Documentation Local »

Default Definitions SyntaxInformation[DixonReconstruct] = {ArgumentsPattern → {_, _, _, _., _., OptionsPattern[]}}

Options › DixonPrecessions → −1 … (13 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonReconstruct

Symbol

DixonToPercent[water, fat] converts the dixon water and fat data to percent maps.

Output is {waterFraction, fatFraction}.

The values of water and fat are arbitraty units and the ouput fractions are between 0 and 1.

Documentation Local »

Default Definitions SyntaxInformation[DixonToPercent] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonToPercent

Symbol

QMRITools`DixonTools`MakeGroups

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`MakeGroups

Symbol

SimulateDixonSignal[echo, fr, B0, T2] simulates an Dixon gradient echo sequence with echotimes.

Echotimes echo in ms, fat fraction fr between 0 and 1, field of resonance B0 in Hz and relaxation T2 in ms.

Documentation Local »

Default Definitions SyntaxInformation[SimulateDixonSignal] = {ArgumentsPattern → {_, _, _, _, OptionsPattern[]}}

Options › DixonNucleus → 1H … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`SimulateDixonSignal

Symbol

Unwrap[data] unwraps the given dataset. The data should be between –Pi and Pi.

Unwrap[] is based on DOI: 10.1364/AO.46.006623 and 10.1364/AO.41.007437.

Documentation Local »

Default Definitions SyntaxInformation[Unwrap] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {MonitorUnwrap → True, UnwrapDimension → 2D, UnwrapThresh → 0.5}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`Unwrap

Symbol

UnwrapSplit[phase, data] unwarps the give phase dataset but splits the data into left

and right using SplitData based in the data and performs the unwrapping seperately. The data should be between –Pi and Pi.

UnwrapSplit[] is based on DOI: 10.1364/AO.46.006623 and 10.1364/AO.41.007437.

Documentation Local »

Default Definitions SyntaxInformation[UnwrapSplit] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options {MonitorUnwrap → True, UnwrapDimension → 2D, UnwrapThresh → 0.5}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`UnwrapSplit

Options

Symbol

DixonAmplitudes is an options for DixonReconstruct. Defines the relative amplitudes of the fat peaks being used.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonAmplitudes

Symbol

DixonBipolar is an option for DixonReconstruct. If set to true it assumes alternating readout directions.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonBipolar

Symbol

DixonClipFraction is an option for DixonReconstruct. If set true the fat fraction is clipped between 0 and 1.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonClipFraction

Symbol

DixonFieldStrength is an options for DixonReconstruct. Defines the fieldstrengths in Tesla on which the data was acquired.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonFieldStrength

Symbol

DixonFilterInput is an options for DixonReconstruct. If True the input b0 and T2star values are smoothed using a gaussian kernel.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonFilterInput

Symbol

DixonFilterOutput is an options for DixonReconstruct. If True the out b0 and T2star values are smoothed Median filter and lowpassfiltering after which the water and fat maps are recomputed.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonFilterOutput

Symbol

DixonFilterSize is an options for DixonReconstruct. Defines the number of voxel with which the input b0 and T2star values are smoothed.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonFilterSize

Symbol

DixonFrequencies is an options for DixonReconstruct. Defines the frequencies in ppm of the fat peaks being used.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonFrequencies

Symbol

DixonIterations is an options for DixonReconstruct. Defines the maximum itterations the fit can use.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonIterations

Symbol

DixonMaskThreshhold is an options for DixonReconstruct. Defines at which threshhold the dixon reconstruction considers a voxel to be background noise. Defualt values is 0.05.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonMaskThreshhold

Symbol

DixonNucleus is an option for DixonReconstruct. Defines the nucleus for which the reconstruction is performed.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonNucleus

Symbol

DixonPrecessions is an options for DixonReconstruct. Defines the rotation of the signal {−1,1} default is −1.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonPrecessions

Symbol

DixonTollerance is an options for DixonReconstruct. Defines at which change per itteration of b0 and R2star the ittarative methods stops. Default value is 0.1.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonTollerance

Symbol

MonitorUnwrap is an option for Unwrap. Monitor the unwrapping progress.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`MonitorUnwrap

Symbol

UnwrapDimension is an option for Unwrap. Can be "2D" or "3D". 2D is for unwarpping 2D images or

unwrapping the individual images from a 3D dataset (does not unwrap in the slice direction). 3D unwraps a 3D dataset in all dimensions.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`UnwrapDimension

Symbol

UnwrapThresh is an option for Unwrap. Is a value between 0.6 and 0.9, and defines when to unwrap, the higher the value the less unwrapping will be done.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`UnwrapThresh

ElastixTools

Functions

Symbol

ReadTransformParameters[directory] reads the tranfomation parameters generated by

RegisterData. The directory should be the TempDirectory were the registration is stored. DeleteTempDirectory should be False.

Output is the affine transformation vector per volume.

Documentation Local »

Default Definitions SyntaxInformation[ReadTransformParameters] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`ReadTransformParameters

Symbol

RegisterCardiacData[data] registers the data using a 2D algorithm. data can be 3D or 4D.

RegisterCardiacData[{data,vox}] registers the data series using the given voxel size.

RegisterCardiacData[{data,mask}] registers the data series only using data whithin the mask.

RegisterCardiacData[{data,mask,vox}] registers the data series using the given voxel size only using data within the mask.

Output is the registered data.

Documentation Local »

Default Definitions SyntaxInformation[RegisterCardiacData] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options › RegistrationTarget → Mean … (17 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterCardiacData

Symbol

RegisterData[data] registers the data series. If data is 3D it performs multiple 2D registration, if data is 4D it performs multipe 3D registration. The target is the first image orvolume in the series.

RegisterData[{data, vox}] registers the data series using the given voxel size.

RegisterData[{data, mask}] registers the data series only using data whithin the mask.

RegisterData[{data, mask, vox}] registers the data series using the given voxel size only using data within the mask.

RegisterData[target, moving] registers the moving data to the target data. target can be 2D or 3D. moving can be the same of one dimension higher than the target.

RegisterData[{target, mask, vox},{moving, mask, vox}] registers the data using the given voxel size only using data within the mask.

RegisterData[{target, vox}, moving] registers the data using the given voxel size.

RegisterData[target, {moving, vox}] registers the data using the given voxel size.

RegisterData[{target, vox}, {moving, vox}] registers the data using the given voxel size.

RegisterData[{target, mask}, moving] registers the data series only using data whithin the mask.

RegisterData[target, {moving, mask}] registers the data series only using data whithin the mask.

RegisterData[{target, mask}, moving] registers the data series only using data whithin the mask.

RegisterData[{target, mask}, {moving, mask}] registers the data series only using data whithin the mask.

RegisterData[target, {moving, mask, vox}] registers the data series using the given voxel size only using data within the mask.

RegisterData[{target, mask}, {moving, mask, vox}] registers the data series using the given voxel size only using data within the mask.

RegisterData[{target, vox}, {moving, mask, vox}] registers the data series using the given voxel size only using data within the mask.

RegisterData[{target, mask, vox}, moving] registers the data series using the given voxel size only using data within the mask.

RegisterData[{target, mask, vox}, {moving, mask}] registers the data series using the given voxel size only using data within the mask.

RegisterData[{target, mask, vox}, {moving, vox}] registers the data series using the given voxel size only using data within the mask.

RegisterData[{target, mask}, {moving, vox}] registers the data series using the given voxel size only using data within the mask.

RegisterData[{target, vox}, {moving, mask}] registers the data series using the given voxel size only using data within the mask.

Output is the registered data with the dimensions of the moving data.

If OutputTransformation is True it also outputs the translation, rotation scale and skew of all images or volumes.

RegisterData[] is based on DOI: 10.1109/TMI.2009.2035616 and 10.3389/fninf.2013.00050.

Documentation Local »

Default Definitions SyntaxInformation[RegisterData] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › Iterations → 250 … (16 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterData

Symbol

RegisterDataSplit[target, moving] is identical to RegisterData data however left and right side of the data are registered seperately.

Splitting the data is done using the function CutData and merged wit Stich data.

Output is the registered data.

Documentation Local »

Default Definitions SyntaxInformation[RegisterDataSplit] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › Iterations → 250 … (17 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterDataSplit

Symbol

RegisterDataTransform[target, moving, {moving2nd, vox}] performs the registration exactly as RegisterData. target and moving are the inputs for Registerdata, which can be {data,mask,vox}.

After the registeration is done the moving2nd data is deformed acording to the output of the registrtion of moving.

moving2nd can have the same dimensions of moving or one dimension higher (e.g. 3D and 3D or 3D and 4D).

Output is {registered moving, deformed moving2nd}.

Documentation Local »

Default Definitions SyntaxInformation[RegisterDataTransform] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › Iterations → 250 … (16 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterDataTransform

Symbol

RegisterDataTransformSplit[target, moving, {moving2nd, vox}] is idenditcal to RegisterDataTransform with the same functionality as RegisterDataSplit.

This means the data is split in two using the function CutData and merged wit Stich data.

Output is {registered moving, deformed moving2nd}.

Documentation Local »

Default Definitions SyntaxInformation[RegisterDataTransformSplit] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › Iterations → 250 … (17 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterDataTransformSplit

Symbol

RegisterDiffusionData[{dtidata, vox}] registers a diffusion dataset. dtidata should be 4D {slice, diff, x, y}. vox is the voxelsize of the data.

RegisterDiffusionData[{dtidata, dtimask, vox}] registers the data series using the given voxel size only using data within the mask.

RegisterDiffusionData[{dtidata ,vox}, {anatdata, voxa}] registers a diffusion dataset. The diffusion data is also registered to the anatdata.

RegisterDiffusionData[{dtidata, dtimask, vox}, {anatdata, voxa}] registers the data series using the given voxel size only using data within the mask.

RegisterDiffusionData[{dtidata,vox}, {anatdata, anatmask, voxa}] registers the data series using the given voxel size only using data within the mask.

RegisterDiffusionData[{dtidata, dtimask, vox}, {anatdata, anatmask, voxa}] registers the data series using the given voxel size only using data within the mask.

Output is the registered dtidata and, if anatdata is given, the registered dtidata in

anatomical space. If OutputTransformation is True it also outputs the translation, rotation scale and skew of all images or volumes.

Documentation Local »

Default Definitions SyntaxInformation[RegisterDiffusionData] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › Iterations → 250 … (23 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterDiffusionData

Symbol

RegisterDiffusionDataSplit[dtidata, vox] is identical to Register diffusion data however left and right side of the data are registered seperately.

RegisterDiffusionDataSplit[{dtidata, vox}, {anatdata, voxa}] is identical to Register diffusion data however left and right side of the data are registered seperately.

RegisterDiffusionDataSplit[{dtidata, dtimask, vox}, {anatdata, anatmask, voxa}] is identical to Register diffusion data however left and right side of the data are registered seperately.

Splitting the data is done using the function CutData and merged wit Stich data.

Output is the registered data.

Documentation Local »

Default Definitions Options[RegisterDiffusionDataSplit] := Options[RegisterDiffusionData]

SyntaxInformation[RegisterDiffusionDataSplit] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterDiffusionDataSplit

Symbol

TransformData[{data,vox}] deforms the data according to the last output of register data.

The directory should be the TempDirectory were the registration is stored. DeleteTempDirectory should be False.

Documentation Local »

Default Definitions SyntaxInformation[TransformData] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options › TempDirectory → Default … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`TransformData

Symbol

If set true Elastix commands will be printed.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`$debugElastix

Options

Symbol

AffineDirections is an option for RegisterData ad RegisterDiffusionData.

It gives the directions in which data can be moved when registering diffusion data to anatomical space.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`AffineDirections

Symbol

BsplineDirections is an option for RegisterData ad RegisterDiffusionData.

It gives the direction in which the bsplines are allowed to move when registering diffusion data to anatomical space.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`BsplineDirections

Symbol

BsplineSpacing is an options for RegisterData, RegisterDiffusionData, RegisterCardiacData and RegisterDataTransform.

It specifies the spacing of the bsplines if the method is "bspline".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`BsplineSpacing

Symbol

DeleteTempDirectory an options for RegisterData, RegisterDiffusionData, RegisterCardiacData and RegisterDataTransform.

It specifies if the temp directory should be deleted after the registration is finisched.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`DeleteTempDirectory

Symbol

FindTransform is an option for TransformData and RegisterTransformData. It specifies where to find the transformfile.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`FindTransform

Symbol

HistogramBins is an options for RegisterData, RegisterDiffusionData, and RegisterDataTransform.

It specifies the number of bins of the joined histogram used by the registration functions.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`HistogramBins

Symbol

HistogramBinsA is an option for RegisterDiffusionData.

It specifies the number of bins of the joined histogram used when registering diffusion data to anatomical space.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`HistogramBinsA

Symbol

InterpolationOrderReg is an options for RegisterData, RegisterDiffusionData, and RegisterDataTransform.

It specifies the interpolation order used in the registration functions.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`InterpolationOrderReg

Symbol

InterpolationOrderRegA is an option for RegisterDiffusionData.

It specifies the interpolation order used in the registration functions when registering diffusion data to anatomical space.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`InterpolationOrderRegA

Symbol

Iterations is an options for RegisterData, RegisterDiffusionData, and RegisterDataTransform.

It specifies the number of iterations used by the registration functions.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`Iterations

Symbol

IterationsA is an option for RegisterDiffusionData.

It specifies the number of iterations used when registering diffusion data to anatomical space.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`IterationsA

Symbol

MethodReg is an options for RegisterData, RegisterDiffusionData, RegisterCardiacData and RegisterDataTransform.

It spefifies which registration method to use.

Mehtods can be be "translation", "rigid", "affine", "bspline", "rigidDTI", "affineDTI", "PCAtranslation", "PCArigid", "PCAaffine", or "PCAbspline".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`MethodReg

Symbol

MethodRegA is an option for RegisterDiffusionData.

It spefifies which registration method to use when registering diffusion data to anatomical space. Mehtods can be be "rigid","affine" or "bspline".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`MethodRegA

Symbol

NumberSamples is an options for RegisterData, RegisterDiffusionData, and RegisterDataTransform.

It specifies the number of random samples that are taken each iteration used by the registration functions.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`NumberSamples

Symbol

NumberSamplesA is an option for RegisterDiffusionData.

It specifies the number of random samples that are taken each iteration when registering diffusion data to anatomical space.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`NumberSamplesA

Symbol

OutputImage is an options for RegisterData, RegisterDiffusionData, and RegisterDataTransform.

It specifies if the result image should be writen in the TempDirectory as nii file.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`OutputImage

Symbol

OutputTransformation is an option for RegisterData ad RegisterDiffusionData.

It specifies if the tranformation paramters (translation, rotation, scale and skew) should be given as output in the registration functions.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`OutputTransformation

Symbol

PCAComponents is an option for RegisterData. It speciefies how many PCA components are used if method is set to "PCA".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`PCAComponents

Symbol

PrintTempDirectory is an options for RegisterData, RegisterDiffusionData, RegisterCardiacData and RegisterDataTransform.

It spefifies if the location of the temp directory should be deplayed.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`PrintTempDirectory

Symbol

RegistrationTarget is an option for RegisterDiffusionData and RegisterCardiacData. Specifies which target to uses for registration if using "rigid", "affine" or "bspline" as MethodReg.

If the MethodReg is "PCA" based it does not need a target and this options does nothing.

Values can be "First", "Mean" or "Median".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegistrationTarget

Symbol

Resolutions is an options for RegisterData, RegisterDiffusionData, and RegisterDataTransform.

It specifies the number of scale space resolutions used by the registration functions.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`Resolutions

Symbol

ResolutionsA is an option for RegisterDiffusionData.

It specifies the number of scale space resolutions used when registering diffusion data to anatomical space.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`ResolutionsA

Symbol

SplitMethod is an option for RegisterDataSplit and RegisterDataTransformSplit. values can be "mean", "moving", "target".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`SplitMethod

Symbol

TempDirectory is an options for RegisterData, RegisterDiffusionData, RegisterCardiacData and RegisterDataTransform.

It specifies the temprary directory used to perform and output the registration.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`TempDirectory

Symbol

UseGPU is an option for RegisterData. The value is {bool, gpu} where bool is True or False, and gpu is the gpu ID which is an integer or Automatic.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`UseGPU

GeneralTools

Functions

Symbol

ApplyCrop[data,crop] aplies the corpped region obtained form CropData to the data.

ApplyCrop[data,crop,{voxorig,voxnew}] aplies the corpped region obtained form CropData to the data.

Documentation Local »

Default Definitions SyntaxInformation[ApplyCrop] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`ApplyCrop

Symbol

AutoCropData[data] crops the data by removing all background zeros.

AutoCropData[data,pad] crops the data by removing all background zeros with padding of pad.

Documentation Local »

Default Definitions SyntaxInformation[AutoCropData] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options CropPadding → 5

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`AutoCropData

Symbol

BSplineCurveFit[points] fits a bspline to the points. Output is a list of same size as points.

Documentation Local »

Default Definitions SyntaxInformation[BSplineCurveFit] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {SplineDegree → 2, SplineKnotsNumber → 50, SplineRegularization → 0}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`BSplineCurveFit

Symbol

CheckExtension[filename, extension] checks if file has correct extention. Removes .gz or add ext if not present.

Default Definitions SyntaxInformation[CheckExtension] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`CheckExtension

Symbol

ClearTemporaryVariables [] Clear temporary variables.

Documentation Local »

Default Definitions SyntaxInformation[ClearTemporaryVariables] = {ArgumentsPattern → {_.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`ClearTemporaryVariables

Symbol

CompilebleFunctions[] generates a formatted table of all compilable functions generated by Compile`CompilerFunctions.

Documentation Local »

Default Definitions SyntaxInformation[CompilebleFunctions] = {ArgumentsPattern → {}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`CompilebleFunctions

Symbol

CropData[data] creates a dialog window to crop the data (assumes voxsize (1,1,1)).

CropData[data,vox] creates a dialog window to crop the data.

Documentation Local »

Default Definitions SyntaxInformation[CropData] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options {CropOutput → All, CropInit → Automatic}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`CropData

Symbol

CutData[data] splits the data in two equal sets left and right.

Documentation Local »

Default Definitions SyntaxInformation[CutData] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`CutData

Symbol

QMRITools`GeneralTools`Data2DToVector

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`Data2DToVector

Symbol

QMRITools`GeneralTools`Data3DToVector

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`Data3DToVector

Symbol

DataToVector[data] converst the non zero data to vector.

DataToVector[data, mask] converst the data within the mask to vector.

the data can be reconstructed using VectorToData.

output is the vecotrized data and a list contining the original data dimensions and a list with the data coordinates. {vec, {dim,pos}}.

Documentation Local »

Default Definitions SyntaxInformation[DataToVector] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`DataToVector

Symbol

DecomposeAffineMatrix[S] decomposes the scale matrix in S1, S2 and S3.

Documentation Local »

Default Definitions SyntaxInformation[DecomposeAffineMatrix] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`DecomposeAffineMatrix

Symbol

DecomposeScaleMatrix[mat] decomposes the affine matirx in T, R, S and Q.

Documentation Local »

Default Definitions SyntaxInformation[DecomposeScaleMatrix] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`DecomposeScaleMatrix

Symbol

DevideNoZero[a, b] devides a/b but when b=0 the result is 0. a can be a number or vector.

Documentation Local »

Default Definitions SyntaxInformation[DevideNoZero] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`DevideNoZero

Symbol

DynamicPartition[data, {part}] patitions the data into parts which is a list of integers. The remainders is los.

DynamicPartition[data,part,last] patitions the data into parts which is a list of integers. The remainders is partitioned into equal parts defined by last.

If last is All, the remainders is just one partition.

Documentation Local »

Default Definitions SyntaxInformation[DynamicPartition] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`DynamicPartition

Symbol

ExpNoZero[val] return the Exp of the val which can be anny dimonsion array. if val=0 the output is 0.

Documentation Local »

Default Definitions SyntaxInformation[ExpNoZero] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`ExpNoZero

Symbol

ExtractDemoData[] Extracts the demo data archilve.

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`ExtractDemoData

Symbol

FileSelect[action] creates a systemdialog wicht returs file/foldername action can be "FileOpen", "FileSave" or "Directory".

FileSelect[action, {type}] same but allows the definition of filetypes for "FileOpen" and "FileSave" e.g. "jpg" or "pdf".

Documentation Local »

Default Definitions SyntaxInformation[FileSelect] = {ArgumentsPattern → {_, _., _., OptionsPattern[]}}

Options WindowTitle → Automatic

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`FileSelect

Symbol

FindCrop[data] finds the crop values of the data by removing all zeros surrounding the data.

Documentation Local »

Default Definitions SyntaxInformation[FindCrop] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options CropPadding → 5

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`FindCrop

Symbol

FindMaxDimensions[{data1, data2, ..}] finds the maximal dimensions of all datasets. Each dataset is 3D.

Documentation Local »

Default Definitions SyntaxInformation[FindMaxDimensions] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`FindMaxDimensions

Symbol

GetAssetLocation[name] Gets the location of the executable assets of the package for the highest installed version.

Current assests are "Elastix", "Transformix" and "DcmToNii".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`GetAssetLocation

Symbol

GridData[{data1,data2,...}, part] makes a grid of multiple datasets with part sets on each row.

Documentation Local »

Default Definitions SyntaxInformation[GridData] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options Padding → None

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`GridData

Symbol

GridData3D[{data1,data2,...}, part] same as grid data, but only works on 4D data where the data is gridded in axial, coronal and sagital.

Documentation Local »

Default Definitions SyntaxInformation[GridData3D] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`GridData3D

Symbol

GyromagneticRatio[] gives the gyromagnetic ratio for "1H" in MHz/T.

GyromagneticRatio[nucle] gives the gyromagnetir ratio for the nuclei, e.g. "31P" of "1H".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`GyromagneticRatio

Symbol

LapFilter[data] Laplacian filter of data with kernel size 0.8.

LapFilter[data, ker] Laplacian filter of data with kernel ker.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`LapFilter

Symbol

LLeastSquares[A, y] = performs a Linear Linear Least Squares fit.

It uses a compiled version of the Pseudo inverse of A.

Documentation Local »

Default Definitions SyntaxInformation[LLeastSquares] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`LLeastSquares

Symbol

LogNoZero[val] return the log of the val which can be anny dimonsion array. if val=0 the output is 0.

Documentation Local »

Default Definitions SyntaxInformation[LogNoZero] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`LogNoZero

Symbol

MADNoZero[vec] return the MAD error of the vec which can be anny dimonsion array. if vec={0...} the output is 0. Zeros are ignored.

Documentation Local »

Default Definitions SyntaxInformation[MADNoZero] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MADNoZero

Symbol

MakeIntFunction[data,int]

MakeIntFunction[data, vox ,int]

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MakeIntFunction

Symbol

MeanNoZero[data] calculates the mean of the data ignoring the zeros.

Documentation Local »

Default Definitions SyntaxInformation[MeanNoZero] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MeanNoZero

Symbol

MedFilter[data] Median filter of data with kernel size 1.

MedFilter[data, ker] Madian filter of data with kernel ker.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MedFilter

Symbol

MedianNoZero[data] calculates the Median of the data ignoring the zeros.

Documentation Local »

Default Definitions SyntaxInformation[MedianNoZero] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MedianNoZero

Symbol

MemoryUsage[] gives a table of which definitions use up memory.

MemoryUsage[n] gives a table of which definitions use up memory, where n is the amout of definitions to show.

Documentation Local »

Default Definitions SyntaxInformation[MemoryUsage] = {ArgumentsPattern → {_.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MemoryUsage

Symbol

NNLeastSquares[A, y] performs a Non Negative Linear Least Squares fit.

finds an x that solves the linear least−squares problem for the matrix equation A.x==y.

output is the solution x.

Documentation Local »

Default Definitions SyntaxInformation[NNLeastSquares] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`NNLeastSquares

Symbol

PadToDimensions[data, dim] pads the data to dimensions dim.

Documentation Local »

Default Definitions SyntaxInformation[PadToDimensions] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options {PadValue → 0., PadDirection → Center}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`PadToDimensions

Symbol

QMRIToolsFuncPrint[] gives a list of all the QMRITools functions with their usage infomation.

Documentation Local »

Default Definitions SyntaxInformation[QMRIToolsFuncPrint] = {ArgumentsPattern → {_.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`QMRIToolsFuncPrint

Symbol

QMRIToolsFunctions[] give list of all the QMRITools packages, functions and options.

QMRIToolsFunctions[p] print a table with length p of all the QMRITools functions and options.

QMRIToolsFunctions["toobox"] gives a list of all the functions and options in toolbox.

QMRIToolsFunctions["toobox", p] gives a table op length p of all the functions and options in toolbox. If toolbox is "All" it will list all toolboxes.

Documentation Local »

Default Definitions SyntaxInformation[QMRIToolsFunctions] = {ArgumentsPattern → {_., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`QMRIToolsFunctions

Symbol

QMRIToolsPackages[] give list of all the QMRITools pacakges.

Documentation Local »

Default Definitions SyntaxInformation[QMRIToolsPackages] = {ArgumentsPattern → {}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`QMRIToolsPackages

Symbol

QuaternionToRotationMatrix[{a, b,c,d}] converts quarternion to rotation matrix R.

Documentation Local »

Default Definitions SyntaxInformation[QuaternionToRotationMatrix] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`QuaternionToRotationMatrix

Symbol

QuaternionVectorToRotationMatrix [{b,c,d}] converts quarternion to rotation matrix R.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`QuaternionVectorToRotationMatrix

Symbol

RescaleData[data,dim] rescales image/data to given dimensions.

RescaleData[data,{vox1, vox2}] rescales image/data from size vox1 to size vox2.

Documentation Local »

Default Definitions SyntaxInformation[RescaleData] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options InterpolationOrder → 3

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`RescaleData

Symbol

ReverseCrop[data,dim,crop] reverses the crop on the cropped data with crop values crop to the original size dim.

ReverseCrop[data,dim,crop,{voxorig,voxnew}] reverses the crop on the cropped data with crop values crop to the original size dim.

Documentation Local »

Default Definitions SyntaxInformation[ReverseCrop] = {ArgumentsPattern → {_, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`ReverseCrop

Symbol

ReverseDimensions[data] reverses the dimensions of the data.

Documentation Local »

Default Definitions SyntaxInformation[ReverseDimensions] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`ReverseDimensions

Symbol

RMSNoZero[vec] return the RMS error of the vec which can be anny dimonsion array. if vec={0...} the output is 0. Zeros are ignored.

Documentation Local »

Default Definitions SyntaxInformation[RMSNoZero] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`RMSNoZero

Symbol

RotateDimensionsLeft[data] rotates the dimensions of the data one to the left.

RotateDimensionsLeft[data, i] rotates the dimensions of the data i to the left.

Documentation Local »

Default Definitions SyntaxInformation[RotateDimensionsLeft] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`RotateDimensionsLeft

Symbol

RotateDimensionsRight[data] rotates the dimensions of the data one to the right.

RotateDimensionsRight[data, i] rotates the dimensions of the data i to the right.

Documentation Local »

Default Definitions SyntaxInformation[RotateDimensionsRight] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`RotateDimensionsRight

Symbol

RotationMatrixToQuaternion [R] converts rotation matrix to quarternions {a, b,c,d}.

Documentation Local »

Default Definitions SyntaxInformation[RotationMatrixToQuaternion] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`RotationMatrixToQuaternion

Symbol

RotationMatrixToQuaternionVector [R] converts rotation matrix to quarternions {b,c,d}.

Documentation Local »

Default Definitions SyntaxInformation[RotationMatrixToQuaternionVector] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`RotationMatrixToQuaternionVector

Symbol

SaveImage[image] exports graph to image, ImageSize, FileType and ImageResolution can be given as options.

SaveImage[image, "filename"] exports graph to image with "filname", ImageSize, FileType and ImageResolution can be given as options.

Documentation Local »

Default Definitions SyntaxInformation[SaveImage] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options {ImageSize → 6000, FileType → .jpg, ImageResolution → 300}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`SaveImage

Symbol

Squeeze[data] Removes the singleton dimensions from data.

Documentation Local »

Default Definitions SyntaxInformation[Squeeze] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`Squeeze

Symbol

StdFilter[data] StandardDeviation filter of data using gaussian kernel 2.

StdFilter[data, ker] StandardDeviation filter of data using kernel with size ker.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`StdFilter

Symbol

StichData[datal,datar] joins left and right part of the data generated by CutData.

Documentation Local »

Default Definitions SyntaxInformation[StichData] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`StichData

Symbol

SumOfSquares[{data1, data2, ... datan}] calculates the sum of squares of the datasets.

Output is the SoS and the weights, or just the SoS.

Documentation Local »

Default Definitions SyntaxInformation[SumOfSquares] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options OutputWeights → True

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`SumOfSquares

Symbol

TensMat[tensor] transforms tensor form vector format {xx,yy,zz,xy,xz,yz} to matrix format {{xx,xy,xz},{xy,yy,yz},{xz,yz,zz}}.

Documentation Local »

Default Definitions SyntaxInformation[TensMat] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`TensMat

Symbol

TensVec [tensor] transforms tensor form matrix format {{xx,xy,xz},{xy,yy,yz},{xz,yz,zz}} to vector format {xx,yy,zz,xy,xz,yz}.

Documentation Local »

Default Definitions SyntaxInformation[TensVec] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`TensVec

Symbol

VectorToData [vec, {dim,pos}] converts the vectroized data from DataToVector back to its original Dimensoins.

Documentation Local »

Default Definitions SyntaxInformation[VectorToData] = {ArgumentsPattern → {_, {_, _}}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`VectorToData

Options

Symbol

CropInit is an option for CropData. By default the crop is not initialized bu can be with {{xmin,xmax},{ymin,ymax},{zmin,zmax}}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`CropInit

Symbol

CropOutput is an option for CropData, can be "All","Data" or "Crop".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`CropOutput

Symbol

CropPadding is an option for AutoCropData or FindCrop. It specifies how much padding to use around the data.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`CropPadding

Symbol

FileType["*file*"] gives the type of a file, typically File, Directory, or None.

Documentation Local » | Web »

Attributes {Protected, ReadProtected}

Full Name System`FileType

Symbol

ImageResolution is an option for Export, Rasterize, and related functions that specifies at what resolution bitmap images should be rendered.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ImageResolution

Symbol

ImageSize is an option that specifies the overall size of an image to display for an object.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ImageSize

Symbol

InterpolationOrder is an option for Interpolation, as well as ListLinePlot, ListPlot3D, ListContourPlot, and related functions, that specifies what order of interpolation to use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`InterpolationOrder

Symbol

OutputWeights is an option for SumOfSqares. If True it also output the SoS weights.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`OutputWeights

Symbol

Padding is an option to various array and image operations that specifies what padding to use when extending beyond the original data specified.

Documentation Local » | Web »

Attributes {Protected, ReadProtected}

Full Name System`Padding

Symbol

PadDirection is an option for PadToDimensions. It specifies the direction of padding, "Center", "Left" or "Right".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`PadDirection

Symbol

PadValue is an option for PadToDimensions. It specifies the value of the padding.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`PadValue

Symbol

SplineDegree is an option for spline functions and graphics primitives that specifies the degree of polynomial basis to use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`SplineDegree

Symbol

SplineKnotsNumber is an option for BSplineCurveFit and defines how many knots the bspline has.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`SplineKnotsNumber

Symbol

SplineRegularization is an option for BSplineCurveFit and defines the amount of regularization for the linear fit.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`SplineRegularization

Symbol

WindowTitle is an option that specifies the title to give for a window.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`WindowTitle

GradientTools

Functions

Symbol

Bmatrix[bvec, grad] creates bmatrix form grad and bvec in form {−bxx, −byy, −bzz, −bxy, −bxz, −byz ,1}.

Bmatrix[{bvec, grad}] creates bmatrix form grad and bvec in form {bxx, byy, bzz, bxy, bxz, byz}.

Documentation Local »

Default Definitions SyntaxInformation[Bmatrix] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options Method → DTI

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`Bmatrix

Symbol

BmatrixCalc["folder", grads] calculates the true bmatrix from the exported sequence parameters from the philips scanner that are stored in "folder" for each of the gradient directions grads.

Documentation Local »

Default Definitions SyntaxInformation[BmatrixCalc] = {ArgumentsPattern → {_, _, _., OptionsPattern[]}}

Options › UseGrad → {1, 1, {1, 1}, 1, 1}… (8 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`BmatrixCalc

Symbol

BmatrixConv[bm] converts the bmatrix form 7 to 6 or from 6 to 7.

Documentation Local »

Default Definitions SyntaxInformation[BmatrixConv] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`BmatrixConv

Symbol

BmatrixInv[bm] generates a bvecotr and gradiens directions form a given bmatrx.

BmatrixInv[bm, bvi] generates a bvecotr and gradiens directions form a given bmatrx using the given bvalues bvi.

Documentation Local »

Default Definitions SyntaxInformation[BmatrixInv] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`BmatrixInv

Symbol

BmatrixRot[bmat, rotmat] Rotates the B–matrix.

Documentation Local »

Default Definitions SyntaxInformation[BmatrixRot] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`BmatrixRot

Symbol

BmatrixToggle[bmat, axes, flip], axes can be any order of {"x","y","z"}. flip should be {1,1,1},{1,1,−1},{1,−1,1} or {−1,1,1}.

Documentation Local »

Default Definitions SyntaxInformation[BmatrixToggle] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`BmatrixToggle

Symbol

CalculateMoments[{Gt, hw, te}, t] calculates the 0th to 3th order moments of the sequence created by GradSeq. Output is {{Gt, M0, M1, M2, M3}, vals}.

Documentation Local »

Default Definitions SyntaxInformation[CalculateMoments] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`CalculateMoments

Symbol

ConditionNumberCalc[grads] calcualtes the condition number of the gradient set.

Documentation Local »

Default Definitions SyntaxInformation[ConditionNumberCalc] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`ConditionNumberCalc

Symbol

ConvertGrads[grad, bv] converts the gradients to txt format, which is needed for FinalGrads.

Documentation Local »

Default Definitions SyntaxInformation[ConvertGrads] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`ConvertGrads

Symbol

CorrectBmatrix[bmat, transformation] corrects the bmatrix bmat with the tranformation parameters from RegisterData or RegisterDiffusionData.

Output is the corrected bmatrix.

Documentation Local »

Default Definitions SyntaxInformation[CorrectBmatrix] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options MethodReg → Full

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`CorrectBmatrix

Symbol

CorrectGradients[grad, transformation] corrects the gradient directions grad with the tranformation parameters from RegisterData or RegisterDiffusionData.

Output is the corrected gradient vector.

Documentation Local »

Default Definitions SyntaxInformation[CorrectGradients] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options MethodReg → Rotation

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`CorrectGradients

Symbol

EnergyCalc[grads] calcualtes the total Energy of the gradient set.

Documentation Local »

Default Definitions SyntaxInformation[EnergyCalc] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`EnergyCalc

Symbol

FinalGrads[grtxt,{int,intn},{rand,order}] finalizes the gradient txt file.

grtxt is the output from the function ConvertGrads, which convert the grad to txt format.

int is True or False, if set to True it interleaves b=0 gradients every intn directions.

rand indicates if the gradients need to be randomized, for this it uses the order which is the output of FindOrder.

Documentation Local »

Default Definitions SyntaxInformation[FinalGrads] = {ArgumentsPattern → {_, {_, _, _.}, {_, _}}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`FinalGrads

Symbol

FindOrder[grad,bv] finds the optimal order of the gradient directions which minimizes the duty cycle.

The output is needed for FinalGrads.

grad is a list of gradient sets and bv is a list of b–values with the same number as the list of gradient sets.

Documentation Local »

Default Definitions SyntaxInformation[FindOrder] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options OrderSpan → Auto

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`FindOrder

Symbol

FullGrad is an option for Grad. Default is True. When true the gradient directions wil be loaded with the first gradient {0,0,0}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`FullGrad

Symbol

GenerateGradients[numb] optimizes a set with numb gradients, numb mus be an integer.

GenerateGradients[{numb, fixed}] optimizes a set with numb gradients, numb must ba an integer and fixed a list of 3D coordiantes e.g. {{0,0,1},{0,1,0}}. The fixed gradients will not be moved.

GenerateGradients[{numb1, numb2 ...}, alpha] optimizes a multi shel gradient set with numb gradients per shel. If alpha is set to 0.5 equal importance is given

tothe optimal distribution of each shell en the enitre set. if alpha is 0 only the sub shels will be optimized, if alpha is set to 1 only the global set wil be optimized.

GenerateGradients[] is based on DOI: 10.1002/mrm.26259 and 10.1002/(SICI)1522–2594(199909)42:3<515::AID–MRM14>3.0.CO;2–Q.

Documentation Local »

Default Definitions SyntaxInformation[GenerateGradients] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › Steps → 1000 ... (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`GenerateGradients

Symbol

GenerateGradientsGUI[] runs the GenerateGradients function in GUI with output for the philips system.

GenerateGradientsGUI[] is based on DOI: 10.1002/mrm.26259 and 10.1002/(SICI)1522–2594(199909)42:3<515::AID–MRM14>3.0.CO;2–Q.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`GenerateGradientsGUI

Symbol

GetGradientScanOrder[grad, bval] determines the scanorder based on the txt file provided to the scanner as input.

GetGradientScanOrder[file, grad, bval] determines the scanorder based on the txt file provided to the scanner as input.

Documentation Local »

Default Definitions SyntaxInformation[GetGradientScanOrder] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`GetGradientScanOrder

Symbol

GetSliceNormal[file] imports the slice normal from a dicom image.

Documentation Local »

Default Definitions SyntaxInformation[GetSliceNormal] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`GetSliceNormal

Symbol

GetSliceNormalDir[file] imports the slice normal from a enhanced dicom image.

Documentation Local »

Default Definitions SyntaxInformation[GetSliceNormalDir] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`GetSliceNormalDir

Symbol

GradBmatrix[Gt, hw, te, t] Calculates the true bmatrix from the sequence created by GradSeq.

Documentation Local »

Default Definitions SyntaxInformation[GradBmatrix] = {ArgumentsPattern → {_, _, _, _, OptionsPattern[]}}

Options {OutputPlot → False, Method → Analytical, StepSizeI → 0.025}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`GradBmatrix

Symbol

GradSeq[pars, t, grad] Creates a sequence from the gradient pars imported by ImportGradObj.

Documentation Local »

Default Definitions SyntaxInformation[GradSeq] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › UseGrad → {0, 1, {1, 0}, 1}… (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`GradSeq

Symbol

ImportGradObj[folder] Imports the gradient par files exported from the philips scanner.

Documentation Local »

Default Definitions SyntaxInformation[ImportGradObj] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`ImportGradObj

Symbol

OverPlusCalc[grads] determines the minimal overplus factor of of the gradient set.

Documentation Local »

Default Definitions SyntaxInformation[OverPlusCalc] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`OverPlusCalc

Symbol

UniqueBvalPosition[bval] generates a list of all the unique bvalues and their positions.

UniqueBvalPosition[bval, num] generates a list of all the unique bvalues and their positions that are present in the dataset equal or more than num times.

Documentation Local »

Default Definitions SyntaxInformation[UniqueBvalPosition] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`UniqueBvalPosition

Options

Symbol

ConditionCalc is an option for GenerateGradients if set to true GenerateGradients will also give the condition number evolution of the system.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`ConditionCalc

Symbol

FlipAxes is an option for GradSeq. Defaul value is {{1,1,1},{1,1,1}}. First three values are for diffusion gradients last three are for the acquisition gradients.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`FlipAxes

Symbol

FlipGrad is an option for GradSeq. When FlipGrad is true the gr180 is fliped.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`FlipGrad

Symbol

FullSphere is an option for GenerateGradients. If set True the gradients will be optimized on a full sphere rather than half a sphere.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`FullSphere

Symbol

GradType is an option GenerateGradients. It specifies what type of gradient set wil be produced, "Normal" or "OverPlus".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`GradType

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

MethodReg is an options for RegisterData, RegisterDiffusionData, RegisterCardiacData and RegisterDataTransform.

It spefifies which registration method to use.

Mehtods can be be "translation", "rigid", "affine", "bspline", "rigidDTI", "affineDTI", "PCAtranslation", "PCArigid", "PCAaffine", or "PCAbspline".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`MethodReg

Symbol

OrderSpan is an options for FindOrder.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`OrderSpan

Symbol

OutputPlot is an option for GradBmatrix. It specifies if the plots of the gradients should also be exported.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`OutputPlot

Symbol

OutputType is an option for BmatrixCalc. Values can be "Matrix" of "Gradients".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`OutputType

Symbol

PhaseEncoding is an options of GradSeq. Values can be "A", "P", "R" and "L".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`PhaseEncoding

Symbol

Runs is an option for GenerateGradients. Set how often the minimalization function is run. The best solution of all runs is the output. Default value is 1.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`Runs

Symbol

Steps is an option GenerateGrads and is the number of step that is used.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`Steps

Symbol

StepSizeI is an option for GradBmatrix. Specifies the integration stepsize is Method –> "Numerical" is used.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`StepSizeI

Symbol

SwitchAxes is an option for GradSeq. Defaul value is {{1,2,3},{1,2,3}}. First three values are for diffusion gradients last three are for the acquisition gradients.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`SwitchAxes

Symbol

UnitMulti is an option for GradSeq. Defaul value is 10^–3. Defines the scaling of the gradient strength.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`UnitMulti

Symbol

UseGrad is an option for GradSeq. The default value is {0, 1, {1, 0}, 1} where {grex, gr180, {grepi1, grepi2}, grdiff, grflow}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`UseGrad

Symbol

VisualOpt is an option for GenerateGradients. Show the minimalization proces of eacht calculation step. Default is False.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`VisualOpt

ImportTools

Functions

Symbol

BvalRead[file] imports the bvalue from a .dcm file. file must be a string.

Documentation Local »

Default Definitions SyntaxInformation[BvalRead] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`BvalRead

Symbol

GradRead[filename] imports the diffusion gradient direction from a .dcm file.

filename must be a string.

Documentation Local »

Default Definitions SyntaxInformation[GradRead] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options ConvertDcm → True

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`GradRead

Symbol

ReadBrukerDiff[""] imports the bruker diffusion data selected by the input dialog.

ReadBrukerDiff["file"] imports the bruker diffusion data from "file", file must be location of 2dseq.

Documentation Local »

Default Definitions SyntaxInformation[ReadBrukerDiff] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options BmatrixOut → True

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ReadBrukerDiff

Symbol

ReadBvalue[folder,nr] imports the gradient directions from the dicom header of the first nr of files in de given folder.

folder must be a string, nr must be a int. Uses BvalRead.

Documentation Local »

Default Definitions SyntaxInformation[ReadBvalue] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ReadBvalue

Symbol

ReadDicom[folder] imports all dicom files from the given folder.

ReadDicom[{file1, file2,...}] imports all the given filenames.

ReadDicom[folder, {file1, file2,...}] imports all the given filenames from the given folder.

ReadDicom[folder, partsize] imports all dicom files from the given folder and partions them in given partsize.

ReadDicom[{file1, file2, ...}, partsize] imports all the given filenames and partions them in given partsize.

ReadDicom[folder, {file1, file2, ...}, partsize] imports all the given filenames from the given folder and partions them in given partsize.

Documentation Local »

Default Definitions SyntaxInformation[ReadDicom] = {ArgumentsPattern → {_, _., _., OptionsPattern[]}}

Options ScaleCorrect → False

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ReadDicom

Symbol

ReadDicomDiff[folder, part] imports all dicom files from the given folder and the corresponding diffusion parameters.

part is the number of diffusion images per slice including the unweighted images.

Documentation Local »

Default Definitions SyntaxInformation[ReadDicomDiff] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options ScaleCorrect → False

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ReadDicomDiff

Symbol

ReadDicomDir[file] reads the image data from a dicom directory.

Documentation Local »

Default Definitions SyntaxInformation[ReadDicomDir] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ReadDicomDir

Symbol

ReadDicomDirDiff[file] reads the image data and relevant diffuison parameters from a dicom directory.

Documentation Local »

Default Definitions SyntaxInformation[ReadDicomDirDiff] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options RotateGradient → True

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ReadDicomDirDiff

Symbol

ReadGradients[folder, nr] imports the diffusion gradient directions from the dicom header of the first nr of files in de given folder.

folder must be a string, nr must be a int. Uses GradRead.

Documentation Local »

Default Definitions SyntaxInformation[ReadGradients] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ReadGradients

Symbol

ReadVoxSize[filename] imports the voxelsize from a .dcm file. filename must be a string.

Imports the pixel and slice spacing from the dicom header. Output is a list containg the voxels size {slice thickness, x, y}.

Documentation Local »

Default Definitions SyntaxInformation[ReadVoxSize] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ReadVoxSize

Symbol

ShiftPar[B0file.dcm,DTIfile.dcm] imports the parameters from the dicom headeand and calculates the needed values to preform B0 field map correction.

Needs a B0 dicom file and a diffusion dicom file.

Documentation Local »

Default Definitions SyntaxInformation[ShiftPar] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ShiftPar

Options

Symbol

BmatrixOut is a option for ImportBrukerData if True the bmatrix is given, if false the gradients and bvec are given.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`BmatrixOut

Symbol

ConvertDcm is an option for GradRead.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ConvertDcm

Symbol

RotateGradient is an option for ReadDicomDirDiff. If False it will also output the gradient direction as stored in the dicom header.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`RotateGradient

Symbol

ScaleCorrect is an option for ReadDicom, ReadDicomDiff, ReadDicomDir and ReadDicomDirDiff. The dicom image values are corrected for rescale slope, scale slope and rescale intercept.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ImportTools`ScaleCorrect

IVIMTools

Functions

Symbol

BayesianIVIMFit2[data, bval, init, mask] performs bayesian IVIM fit of data.

data is the data which should be {slice, Ndiff, x, y}.

bval is the bvector whould be length Ndiff.

init is the initalization of the bayesian fit which comes from IVIMCalc, (without S0 using 2 compartments).

mask is the region in which the bayesian fit is performed.

output is {f1, dc, pdc1}. The fraction is defined between 0 and 1, the dc, pdc1 is in mm^2/s.

Documentation Local »

Default Definitions SyntaxInformation[BayesianIVIMFit2] = {ArgumentsPattern → {_, _, _, _, OptionsPattern[]}}

Options › ChainSteps → {20 000, 1000, 10} … (7 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`BayesianIVIMFit2

Symbol

BayesianIVIMFit3[data, bval, init, mask] performs bayesian IVIM fit of data.

data is the data which should be {slice, Ndiff, x, y}.

bval is the bvector whould be length Ndiff.

init is the initalization of the bayesian fit which comes from IVIMCalC, (without S0 using 3 compartments).

mask is the region in which the bayesian fit is performed.

output is {f1, f2, dc, pdc1, pdc2}. The fractions f1 and f2 are defined between 0 and 1, the dc, pdc1 and pdc1 is in mm^2/s.

Documentation Local »

Default Definitions SyntaxInformation[BayesianIVIMFit3] = {ArgumentsPattern → {_, _, _, _, OptionsPattern[]}}

Options › ChainSteps → {20 000, 1000, 10} … (7 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`BayesianIVIMFit3

Symbol

CorrectParMap[par, constraints, mask] removes the IVIM parameters outside the constraints within the mask.

par is {f1, dc, pdc1} or {f1, f2, dc, pdc1, pdc2}.

constraints are the lower and upper constraints for each parameters {{min, max},...}.

mask has the same dimensions as the parameter maps.

output are the corrected paremeter maps.

Documentation Local »

Default Definitions SyntaxInformation[CorrectParMap] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`CorrectParMap

Symbol

FConvert[F] convers the fraction F from log space.

Documentation Local »

Default Definitions SyntaxInformation[FConvert] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FConvert

Symbol

FConverti[f] converts the fraction f to log space.

Documentation Local »

Default Definitions SyntaxInformation[FConverti] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FConverti

Symbol

FracCorrect[fraction, time] corrects the signal fraction calculated with the IVIM model for tissue relaxation and acquisition parameters.

After correction the signal fraction can be regarded as volume fraction.

FracCorrect[{fraction1, fraction2}, time] corrects the signal fraction1 and fraction2 from a 3 compartement IVIM model.

time is {{te, tr}, {t2t, t21}, {t1t, t11}} or {{te, tr}, {t2t, t21, t22}, {t1t, t11, t12}}.

where t2t and t1t are "tissue" relaxation times and t11 t12, t21 and t22 the "fluid" relaxation times.

The te and tr as well as the relaxation times T2 and T1 can be defines in any time unit as long as they are consistant for all, e.g. all in ms.

output is the corrected fraction maps.

Documentation Local »

Default Definitions SyntaxInformation[FracCorrect] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FracCorrect

Symbol

HistogramPar[data, {constraints, Nbins}, style, color, range] plots histograms of IVIM solution.

HistogramPar[data, {constraints, Nbins, mu, conv}, components, color, range] plots histograms of IVIM solution.

data is {f1, dc, pdc1} or {f1, f2, dc, pdc1, pdc2}.

constraints are the ranges of the x–axes for the plots.

Nbins are the number of histogram bins.

style is the plot type, can be 1, 2, or 3.

color is the color of the histogram.

range are the ranges of the y–axes.

output is a row of histograms.

Documentation Local »

Default Definitions SyntaxInformation[HistogramPar] = {ArgumentsPattern → {_, _, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`HistogramPar

Symbol

IVIMCalc[data, binp, init] calculates the IVIM fit.

data should be 1D ,2D, 3D or 4D.

binp should be full bmatrix which can be calculated from the bvecs en bvals using Bmatrix with the bvalues in s/mm^2.

init should are the initialization parameters for 2 components this is {S0, f, D, Dp} for 3 componentes this is {S0, f1, f2, D, Dp1, Dp2}.

The fraction is defined between 0 and 1, the D, Dp, Dp1 and Dp2 is in mm^2/s.

output is {S0, f1, D, pD1} or {S0, f1, f2, D, pD1, pD2}.

Documentation Local »

Default Definitions SyntaxInformation[IVIMCalc] = {ArgumentsPattern → {_, _, _, _., OptionsPattern[]}}

Options › Method → Automatic … (8 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMCalc

Symbol

IVIMCorrectData[data, {S0, f, pdc}, bval] removes the ivim signal from the data.

data is the original data.

{S0, f, pdc} are the solution to a 2 compartment IVIM fit using IVIMCalc or BayesianIVIMFit2.

bval are the bvalues.

The fraction is defined between 0 and 1, the pdc is in mm^2/s.

output is the corrected data.

Documentation Local »

Default Definitions SyntaxInformation[IVIMCorrectData] = {ArgumentsPattern → {_, {_, _, _}, _, OptionsPattern[]}}

Options {FilterMaps → True, FilterType → Median, FilterSize → 1}

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMCorrectData

Symbol

IVIMFunction[] gives the IVIM function with 2 comps.

IVIMFunction[components] gives the IVIM function.

IVIMFunction[components, type] gives the IVIM function.

type can be "Normal" or "Exp".

componenets can be 2 or 3.

output is the function with b, S0, f1, f2, D, pD1, pD2 as parameters. The fraction is defined between 0 and 1, the D, Dp, Dp1 and Dp2 is in mm^2/s.

Documentation Local »

Default Definitions SyntaxInformation[IVIMFunction] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMFunction

Symbol

IVIMResiduals[data, binp, pars] calculates the root mean square residuals of an IVIM fit ussing IVIMCalc, BayesianIVIMFit2 or BayesianIVIMFit3.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMResiduals

Symbol

ThetaConv[{F1, Fc, pDc}] converts the parameters from Log space to normal space. Is used in BayesianIVIMFit2 and BayesianIVIMFit3.

ThetaConv[{F1, F2, Dc, pDc1}] converts the parameters from Log space to normal space. Is used in BayesianIVIMFit2 and BayesianIVIMFit3.

ThetaConv[{F1, F2, Dc, pDc1, pDc2}] converts the parameters from Log space to normal space. Is used in BayesianIVIMFit2 and BayesianIVIMFit3.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`ThetaConv

Symbol

ThetaConvi[{f, dc, pdc}] converts the parameters from Normal space to Log space. Is used in BayesianIVIMFit2 and BayesianIVIMFit3.

ThetaConvi[{f1, f2, dc, pdc1}] converts the parameters from Normal space to Log space. Is used in BayesianIVIMFit2 and BayesianIVIMFit3.

ThetaConvi[{f1, f2, dc, pdc1, pdc2}] converts the parameters from Normal space to Log space. Is used in BayesianIVIMFit2 and BayesianIVIMFit3.

Documentation Local »

Default Definitions SyntaxInformation[ThetaConvi] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`ThetaConvi

Options

Symbol

ChainSteps is an option for BayesianIVIMFit2 and BayesianIVIMFit3. It determines how long the algorithm runs.

three values must be given {itterations, burn steps, sample density}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`ChainSteps

Symbol

CorrectPar is an option for BayesianIVIMFit2 and BayesianIVIMFit3. If True it removes the values outside the contraints using CorrectParMap.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`CorrectPar

Symbol

FilterMaps is an option for IVIMCorrectData. If True the IVIM parameter maps are filtered before signal correction.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FilterMaps

Symbol

FilterSize is an option for IVIMCorrectData. If FilterMaps is True it gives the kernel size.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FilterSize

Symbol

FilterType is an option for IVIMCorrectData. If FilterMaps is True it tells which filter to use. can be "Median" of "Gausian".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FilterType

Symbol

FitConstrains is an option for BayesianIVIMFit2 and BayesianIVIMFit3. Gives the contraints of the parameters.

The values are used for displaying the histograms and for the initialization if CorrectPar is True.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FitConstrains

Symbol

FixPseudoDiff is an option for BayesianIVIMFit2 and BayesianIVIMFit3. If the pDc1 and pD2 were fixed in IVIMCalc this value should be True.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FixPseudoDiff

Symbol

FixPseudoDiffSD is an option for BayesianIVIMFit2 and BayesianIVIMFit3. Gives the standard deviation of pDc1 and pD2 if FixPseudoDiff is True.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FixPseudoDiffSD

Symbol

IVIMComponents is an option for IVIMCalc. Default value is 2, the tissue and the blood component. can also be set to 3.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMComponents

Symbol

IVIMConstrained is an option for IVIMCalc. When set True the fit wil be constrained to the values given in IVIMConstrains.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMConstrained

Symbol

IVIMConstrains is an option for IVIMCalc.

Default values are: {{0.8, 1.2}, {0, 1}, {0.0005, 0.0035}, {0.005, 0.5}, {0.002, 0.015}}.

Where {{S0 in percentage},{fractions},{tissue diffusion},{blood compartment Dp},{third compartment}}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMConstrains

Symbol

IVIMFixed is an option for IVIMCalc and the default value is False.

When set True the pseudo diffusion wil be fixed to the parameter given as init.

When set to "One" only the fast component of a 3 compartment fit is fixed.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMFixed

Symbol

IVIMTensFit is an option for IVIMCalc. When set True the tissue diffusion component wil be calculated as a tensor.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMTensFit

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

MonitorIVIMCalc is an option for IVIMCalc. When true the proceses of the calculation is shown.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`MonitorIVIMCalc

Symbol

OutputSamples is an option for BayesianIVIMFit2 and BayesianIVIMFit3. If set True the full marcov chain is given as an additionaln output.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`OutputSamples

Symbol

Parallelize[*expr*] evaluates *expr* using automatic parallelization.

Definitions

```
Parallelize[Parallel`Evaluate`Private`expr_, Parallel`Evaluate`Private`opts : OptionsPattern[]] :=
 Module[{Parallel`Evaluate`Private`res, Parallel`Evaluate`Private`ehead, Parallel`Evaluate`Private`fopts}, Parallel`Protected`tryRelaunch[];
  If[Parallel`Protected`$seQ, Message[Parallelize::nopar];
   Return[Parallel`Evaluate`Private`expr]];
  Parallel`Evaluate`Private`fopts = Sequence @@ {Method → OptionValue[Method],
     DistributedContexts → Parallel`Protected`DistOptCheck[Parallelize, OptionValue[DistributedContexts]], ProgressReporting → OptionValue[ProgressReporting]};
  Parallel`Evaluate`Private`ehead = Replace[Head[Unevaluated[Parallel`Evaluate`Private`expr], Hold],
    {Hold[Parallel`Evaluate`Private`s_Symbol] :→ Parallel`Evaluate`Private`s, _ :→ $Failed}];
  With[{Parallel`Evaluate`Private`handler = Lookup[Parallel`Evaluate`$ParallelExtensions, Parallel`Evaluate`Private`ehead, $Failed]},
   If[Parallel`Evaluate`Private`handler =!= $Failed, Return[Parallel`Evaluate`Private`handler[Parallel`Evaluate`Private`expr, Parallel`Evaluate`Private`fopts]]]];
  Parallel`Evaluate`Private`res = Parallel`Evaluate`Private`tryCombine[Parallel`Evaluate`Private`expr, Parallel`Evaluate`Private`fopts];
  If[Head[Parallel`Evaluate`Private`res] =!= Parallel`Evaluate`Private`tryCombine, Return[Parallel`Evaluate`Private`res]];
  Block[{Parallel`Evaluate`Private`$seqWarning = False},
   Parallel`Evaluate`Private`res = Parallel`Evaluate`Private`wrapAround[Parallel`Evaluate`Private`expr, Parallel`Evaluate`Private`fopts];
   If[Head[Parallel`Evaluate`Private`res] =!= Parallel`Evaluate`Private`wrapAround, Return[Parallel`Evaluate`Private`res]];];
  Parallel`Evaluate`Private`res = Parallel`Evaluate`Private`silentFail[Parallel`Evaluate`Private`expr];
  If[Head[Parallel`Evaluate`Private`res] =!= Parallel`Evaluate`Private`silentFail, Return[Parallel`Evaluate`Private`res]];
  If[Parallel`Evaluate`Private`$seqWarning, Message[Parallelize::nopar1, Parallel`Evaluate`Private`expr]];
  Parallel`Evaluate`Private`expr]
```

Documentation Local » | Web »

Options {DistributedContexts :→ $Context, Method → Automatic, ProgressReporting :→ $ProgressReporting}

Attributes {HoldFirst, Protected}

Full Name System`Parallelize

Symbol

UpdateStep is an option for BayesianIVIMFit2 and BayesianIVIMFit3. It determines how often the parameters are updated. Is optimized during the first 500 burn steps.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`UpdateStep

JcouplingTools

Functions

Symbol

GetSpinSystem[name] get a spinsystem that can be used in SimHamiltonian. Current implementes systems are "glu", "lac", "gaba", "fatGly", "fatAll", "fatEnd", "fatDouble", "fatStart", and "fatMet".

Documentation Local »

Default Definitions SyntaxInformation[GetSpinSystem] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options CenterFrequency → 0

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`GetSpinSystem

Symbol

MakeSpinSystem[name, freqs, jcoup] makes a spin system for jcoupling simulations. The with name is defined by the freqs of the nuclei and the jcoup values {{n1, nx}, j} between nuclei.

MakeSpinSystem[{name,labs}, freqs, jcoup] same but each nuclei has a specific name, e.g.{"ATP", {"γ","α","β"}.

MakeSpinSystem[name, freqs, jcoup, scales] same but each nuclei has a scale, default scales are 1.

MakeSpinSystem[{name,labs}, freqs, jcoup, scales] same as alle before.

Documentation Local »

Default Definitions SyntaxInformation[MakeSpinSystem] = {ArgumentsPattern → {_, _., _., _., OptionsPattern[]}}

Options CenterFrequency → 0

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`MakeSpinSystem

Symbol

SequencePulseAcquire[din, H] performs a pulsaquire experiment of the spin system din given the hamiltonian H with a 90 Degree pulse.

SequencePulseAcquire[din, H, b1] performs a pulsaquire experiment of the spin system din given the hamiltonian H with a 90 Degree pulse and b1.

The output is a new spinsystem dout.

Documentation Local »

Default Definitions SyntaxInformation[SequencePulseAcquire] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SequencePulseAcquire

Symbol

SequenceSpaceEcho[din, H, t1, t2, necho, b1] performs a multi echo spin echo

experiment with a 90 degree spin echo, with t1 the time between the 90 degree RF pulse and the first 180 degree RF pulse,

t2 the time betwteen a 180 degree RF pulse and the following readout (and 2xt1 the time between two consecutive 180 degree RF pulses.

Further defines necho the number of 180 degree RF pulses, din the spin system given the hamiltonian H using b1.

The t1 and t2 are defined in ms, and b1 of 100% is defines as 1.

The output is a new spinsystem dout.

Documentation Local »

Default Definitions SyntaxInformation[SequenceSpaceEcho] = {ArgumentPattern → {_, _, _, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SequenceSpaceEcho

Symbol

SequenceSpinEcho[din, H, te] performs a spin echo experiment with echo time te of the spin system din given the hamiltonian H with a 90 and 180 Degree pulse.

SequenceSpinEcho[din, H, te, b1] performs a spin echo experiment with echo time te of the spin system din given the hamiltonian H with a 90 and 180 Degree pulse and b1.

The te is defined in ms and the b1 of 100% is defined as 1.

The output is a new spinsystem dout.

Documentation Local »

Default Definitions SyntaxInformation[SequenceSpinEcho] = {ArgumentsPattern → {_, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SequenceSpinEcho

Symbol

SequenceSteam[din, H, {te, tm}] performs a stimulated echo experiment with echo time te and mixing time tm of the spin system din given the hamiltonian H with 3 90 Degree pulses.

The te and tm are defined in ms.

The output is a new spinsystem dout.

Documentation Local »

Default Definitions SyntaxInformation[SequenceSteam] = {ArgumentsPattern → {_, _, {_, _}}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SequenceSteam

Symbol

SequenceTSE[din ,H, {te, necho}, {ex, ref}] performs a multi echo spin echo experiment with echo

time te with necho echos of the spin system din given the hamiltonian H using ex Degree exitation and ref Degree refocus pulses.

SequenceTSE[din ,H, {te, necho}, {ex, ref}, b1] performs a multi echo spin echo experiment with echo

time te with necho echos of the spin system din given the hamiltonian H using ex Degree exitation and ref Degree refocus pulses and b1.

The te is defined in ms, the ex and ref are defined in degree and b1 of 100% is defined as 1.

The output is a new spinsystem dout.

Documentation Local »

Default Definitions SyntaxInformation[SequenceTSE] = {ArgumentsPattern → {_, _, {_, _, _}, {_, _}, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SequenceTSE

Symbol

SimAddPhase[din ,H ,phase] adds phase to the spin system din given the hamiltonian H.

din and H are generated by SimHamiltonian.

The phase is defined in degree.

The output is a new spinsystem dout.

Documentation Local »

Default Definitions SyntaxInformation[SimAddPhase] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SimAddPhase

Symbol

SimEvolve[din,H,t] evolves the spin system din given the hamiltonian H over a time t. din and H are generated by SimHamiltonian.

The output is a new spinsystem dout.

Documentation Local »

Default Definitions SyntaxInformation[SimEvolve] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SimEvolve

Symbol

SimHamiltonian[sysi] simulates the hamiltionan for a given spin system. The spinsystem is generated by GetSpinSystem.

The output is the spin system and hamiltonian structure.

SimHamiltonian[] is based on DOI: 10.1016/j.jmr.2010.12.008 and 10.1002/mrm.24340.

Documentation Local »

Default Definitions SyntaxInformation[SimHamiltonian] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {FieldStrength → 3, SimNucleus → 1H, CenterFrequency → 0}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SimHamiltonian

Symbol

SimReadout[din, H] performs a readout of a spinsystem din with hamiltonian H.

Output is {time,fids,ppm,spec,dout}, which are the free induction decay fids with its time, the spectrum spec with its ppm and the evolved spin system dout.

Documentation Local »

Default Definitions SyntaxInformation[SimReadout] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options › ReadoutOutput → all ... (8 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SimReadout

Symbol

SimRotate[din, H ,angle] rotates the spin system din given the hamiltonian H over angele with phase 90 degrees.

SimRotate[din, H ,angle, phase] rotates the spin system din given the hamiltonian H over angele with phase.

din and H are generated by SimHamiltonian.

The angle and phase are defined in degree.

The output is a new spinsystem dout.

Documentation Local »

Default Definitions SyntaxInformation[SimRotate] = {ArgumentsPattern → {_, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SimRotate

Symbol

SimSignal[din, H] performs a readout of a spinsystem din with hamiltonian H.

Output is the complex signal.

Documentation Local »

Default Definitions SyntaxInformation[SimSignal] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options ReadoutOutput → all

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SimSignal

Symbol

SimSpoil[din] spoils all the non zeroth order states of a spin system.

The output is a new spinsystem dout.

Documentation Local »

Default Definitions SyntaxInformation[SimSpoil] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SimSpoil

Symbol

SysTable[sys] shows the spinsystem as a table. The spinsytem is obtained form GetSpinSystem.

Documentation Local »

Default Definitions SyntaxInformation[SysTable] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SysTable

Options

Symbol

CenterFrequency is an option for GetSpinSystem and defines the center frequency in ppm.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`CenterFrequency

Symbol

FieldStrength is an option for SimHamiltonian. It defines the field strength for which the hamiltonian is calculated defined in Tesla.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`FieldStrength

Symbol

Linewidth is an option for SimReadout and defines the spectral linewidth in Hz.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`Linewidth

Symbol

LinewidthShape is an option for SimReadout and defines the linewidth shape, values can be "Lorentzian", "Gaussian" or "Voigt".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`LinewidthShape

Symbol

ReadoutBandwith is an option for SimReadout defines the spectral bandwith in Hz.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`ReadoutBandwith

Symbol

ReadoutMethod is an option of SimReadout and can be "Fid" or "Echo". With "Fid" it is also possbile to define a delay time in ms {"Fid", delay}.

With "Echo" it is also possbile to define a delay time in ms {"Echo", delay} and it than assumes te is half the readout, or a custom te can be defined {"Echo", delay, te}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`ReadoutMethod

Symbol

ReadoutOutput is an option for SimReadout and SimSignal and values can be "all" and "each".

When set to "all" the total signal and signal is given, when set to "each" the signal or spectrum for each peak is given seperately.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`ReadoutOutput

Symbol

ReadoutPhase is an option for SimReadout and defines the readout phase in degrees.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`ReadoutPhase

Symbol

ReadoutSamples is an option for SimReadout and defines the number of readout samples for the spectrum.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`ReadoutSamples

Symbol

SimNucleus is an option for SimHamiltonian. It defines the nucleus for which to simulate the spectra.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`SimNucleus

MaskingTools

Functions

Symbol

GetMaskData[data, mask] retruns the data selected by the mask.

Documentation Local »

Default Definitions SyntaxInformation[GetMaskData] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options {GetMaskOutput → All, GetMaskOnly → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`GetMaskData

Symbol

GrowMask[mask,size] if size > 0 the mask is dilated and if size < 0 the mask is eroded.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`GrowMask

Symbol

HomoginizeData[data, mask] tries to homoginize the data within the mask by removing intensity gradients.

Documentation Local »

Default Definitions SyntaxInformation[HomoginizeData] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`HomoginizeData

Symbol

Mask[data] creates a mask by automatically finding a threshold.

Mask[data, min]creates a mask which selects only data above the min value.

Mask[data,{min,max}] creates a mask which selects data between the min and max value.

Documentation Local »

Default Definitions SyntaxInformation[Mask] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › MaskSmoothing → False … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`Mask

Symbol

MaskData[data, mask] applies a mask to data. mask can be 2D or 3D, data can be 2D, 3D or 4D.

Documentation Local »

Default Definitions SyntaxInformation[MaskData] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MaskData

Symbol

MeanSignal[data] calculates the mean signal per volume of 4D data.

MeanSignal[data, pos] calculates the mean signal per volume of 4D data for the given list of positions.

Documentation Local »

Default Definitions SyntaxInformation[MeanSignal] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options UseMask → True

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MeanSignal

Symbol

MergeSegmentations[masks, labels] generates an ITKsnap or slices3D compatible segmentation from individual masks and label numbers.

Output is a labled segmentation.

Documentation Local »

Default Definitions SyntaxInformation[MergeSegmentations] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MergeSegmentations

Symbol

NormalizeData[data] normalizes the data to the mean signal of the data. For 4D data it normalizes to the first volume of the 4th dimension.

NormalizeData[data,{min,max}] normalizes the data between min and max.

Documentation Local »

Default Definitions SyntaxInformation[NormalizeData] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`NormalizeData

Symbol

NormalizeMeanData[data] calculates the mean normalized data from a 4D dataset.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`NormalizeMeanData

Symbol

RemoveMaskOverlaps[mask] removes the overlaps between multiple masks. Mask is a 4D dataset with {z, masks, x, y}.

Documentation Local »

Default Definitions SyntaxInformation[RemoveMaskOverlaps] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`RemoveMaskOverlaps

Symbol

RescaleSegmentation[data, dim] rescales segmentations to given dimensions.

RescaleSegmentation[data, {vox1, vox2}] rescales segmentations from voxelsize vox1 to voxelsize vox2.

Documentation Local »

Default Definitions SyntaxInformation[RescaleSegmentation] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`RescaleSegmentation

Symbol

ROIMask[maskdim, {name–>{{{x,y},slice}..}..}] crates mask from coordinates x and y at slice.

maskdim is the dimensions of the output {zout,xout,yout}.

Documentation Local »

Default Definitions SyntaxInformation[ROIMask] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`ROIMask

Symbol

SegmentMask[mask, n] divides a mask in n segments along the slice direction, n must be an integer. The mask is divided in n equal parts where each parts has the same number of slices.

Documentation Local »

Default Definitions SyntaxInformation[SegmentMask] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SegmentMask

Symbol

SmoothMask[mask] generates one clean masked volume form a noisy mask.

Documentation Local »

Default Definitions SyntaxInformation[SmoothMask] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {MaskComponents → 1, MaskClosing → 5, MaskFiltKernel → 2}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SmoothMask

Symbol

SmoothSegmentation[masks] smooths segmentations and removes the overlaps between multiple segmentations.

Documentation Local »

Default Definitions SyntaxInformation[SmoothSegmentation] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {MaskFiltKernel → 2, MaskComponents → 1}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SmoothSegmentation

Symbol

SplitSegmentations[segmentation] splits a lable mask from ITKsnap or slicer3D in seperate masks and label numbers.

Output is masks and label numbers, {mask, labs}.

Documentation Local »

Default Definitions SyntaxInformation[SplitSegmentations] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SplitSegmentations

Options

Symbol

GetMaskOnly is an option for GetMaskData. If set True all values in the mask are given, if set False only non zero values in the mask are give.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`GetMaskOnly

Symbol

GetMaskOutput is an option for GetMaskData. Defaul is "Slices" which gives the mask data per slices. Else the entire mask data is given as output.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`GetMaskOutput

Symbol

MaskClosing is an option for Mask and SmoothMask. The size of the holes in the mask that will be closed.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MaskClosing

Symbol

MaskComponents is an option for Mask and SmoothMask. Determinse the amount of largest clusters used as mask.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MaskComponents

Symbol

MaskDilation is an option for Mask. If the value is greater than 0 it will dilate the mask, if the value is smaller than 0 it will erode the mask.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MaskDilation

Symbol

MaskFiltKernel is an option for Mask, SmoothMask and SmoothSegmentation. How mucht the contours are smoothed.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MaskFiltKernel

Symbol

MaskSmoothing is an options for Mask, if set to True it smooths the mask, by closing holse and smoothing the contours.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MaskSmoothing

Symbol

UseMask is a function for MeanSignal and DriftCorrect.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`UseMask

NiftiTools

Functions

Symbol

CompressNiiFiles[] promts for a folder. It then compresses all nii files to .nii.gz files in the selected folder.

CompressNiiFiles[folder] compresses all nii files to .nii.gz files in folder.

Documentation Local »

Default Definitions SyntaxInformation[CompressNiiFiles] = {ArgumentsPattern → {_., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`CompressNiiFiles

Symbol

CorrectNiiOrientation[data,hdr] corrects the data orientation based on the nii header.

Documentation Local »

Default Definitions SyntaxInformation[CorrectNiiOrientation] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`CorrectNiiOrientation

Symbol

DcmToNii[] converts a dicom folder to nii, you will be prometed for the location of the folders.

DcmToNii[{"input","ouput"}] converts the "input" dicom folder to nii files which are place in the "output" folder.

For this function to work the dcm2niix.exe file should be present in the QMRITools aplication folder.

Documentation Local »

Default Definitions SyntaxInformation[DcmToNii] = {ArgumentsPattern → {_., _., OptionsPattern[]}}

Options › CompressNii → True … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`DcmToNii

Symbol

ExportBmat[bmat] exports the diffusion bmatrix to exploreDTI format.

ExportBmat[bmat, "file"] exports the diffusion bmatrix to "file" in the exploreDTI format.

Documentation Local »

Default Definitions SyntaxInformation[ExportBmat] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ExportBmat

Symbol

ExportBval[bvals] exports the diffusion bvalues to exploreDTI format.

ExportBval[bvals, "file"] exports the diffusion bvalues to "file" in the exploreDTI format.

Documentation Local »

Default Definitions SyntaxInformation[ExportBval] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ExportBval

Symbol

ExportBvec[grad] exports the diffusion gradients to exploreDTI format.

ExportBvec[grad, "file"] exports the diffusion gradients to "file" in the exploreDTI format.

Documentation Local »

Default Definitions SyntaxInformation[ExportBvec] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ExportBvec

Symbol

ExportNii[data, vox] exports the nii file and will promt for a file name.

ExportNii[data, vox, "file"] exports the nii file to the location "file".

Documentation Local »

Default Definitions SyntaxInformation[ExportNii] = {ArgumentsPattern → {_, _, _., OptionsPattern[]}}

Options › NiiDataType → Automatic ... (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ExportNii

Symbol

ExtractNiiFiles[] promts for a folder. It then extracts all nii.gz files to .nii files in the selected folder.

ExtractNiiFiles[folder] extracts all nii.gz files to .nii files in folder.

Documentation Local »

Default Definitions SyntaxInformation[ExtractNiiFiles] = {ArgumentsPattern → {_., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ExtractNiiFiles

Symbol

GetNiiOrientation[hdr] get the sform and qform orientations from a nii header.

Documentation Local »

Default Definitions SyntaxInformation[GetNiiOrientation] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`GetNiiOrientation

Symbol

ImportBmat[] will promt to select the *.txt file containing the bmatrix.

ImportBmat[*.txt] imports the given *.txt file containing the bmatrix.

Documentation Local »

Default Definitions SyntaxInformation[ImportBmat] = {ArgumentsPattern → {_., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportBmat

Symbol

ImportBval[] will promt to select the *.bval file.

ImportBval[*.bval] imports the given *.bval file.

Documentation Local »

Default Definitions SyntaxInformation[ImportBval] = {ArgumentsPattern → {_.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportBval

Symbol

ImportBvalvec[] will promt to select the *.bval and *.bvec files.

ImportBvalvec[file] if file is either a *.bval or *.bvec it will automatically import the *.bval and *.bvec files.

ImportBvalvec[*.bvec,*.bval] imports the given *.bval and *.bvec files.

Documentation Local »

Default Definitions SyntaxInformation[ImportBvalvec] = {ArgumentsPattern → {_., _., OptionsPattern[]}}

Options FlipBvec → False

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportBvalvec

Symbol

ImportBvec[] will promt to select the *.bvec file.

ImportBvec[*.bvec] imports the given *.bvec file.

Documentation Local »

Default Definitions SyntaxInformation[ImportBvec] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {FlipBvec → False, PositiveZ → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportBvec

Symbol

ImportExploreDTItens["file"] imports the *.nii export for the tensor from explore DTI.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportExploreDTItens

Symbol

ImportNii[] promts to select the nii file to import.

ImportNii["file"] imports the nii file.

The default output is {data, vox}, however using NiiMethod various outputs can be given.

The Nii import is also suported using the native Import function from Mathematica.

Documentation Local »

Default Definitions SyntaxInformation[ImportNii] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {NiiMethod → default, NiiScaling → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportNii

Symbol

ImportNiiDiff[] will promt for the *.nii, *.bvec and *.bval file to import.

ImportNiiDiff[*.nii] will import the *.nii file and automatically also imports the *.bvec and *.bval is they have the same name.

ImportNiiDiff[*.nii,*.bvec,*.bval] will import the given files.

The output will be {data,grad,bvec,vox}.

Documentation Local »

Default Definitions SyntaxInformation[ImportNiiDiff] = {ArgumentsPattern → {_., _., _., OptionsPattern[]}}

Options {RotateGradients → False, FlipBvec → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportNiiDiff

Symbol

ImportNiiDix["file"] imports the dixon nii file which should contain all possible outputs given by the scanner and corrects them accordingly.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportNiiDix

Symbol

ImportNiiT1["file"] imports the T1 file which should contain the echos and the T1map calculated by the scanner and corrects them accordingly.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportNiiT1

Symbol

ImportNiiT2["file"] imports the T2 file which should contain the echos and the T2map calculated by the scanner and corrects them accordingly.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportNiiT2

Symbol

MakeNiiOrentationQ[R] makes the q vector from rotation matrix R.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`MakeNiiOrentationQ

Symbol

MakeNiiOrentationS[off, vox] maxes the srow values for nii header assuming not R and Q.

MakeNiiOrentationS[off, vox, R] maxes the srow values for nii header using rotation R.

MakeNiiOrentationS[off, vox, R, Q] maxes the srow values for nii header using rotation R and skew Q.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`MakeNiiOrentationS

Options

Symbol

CompressNii is an option for DcmToNii and ExportNii. If set True .nii.gz files will be created.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`CompressNii

Symbol

DeleteOutputFolder is an option of DcmToNii. If the ouput folder already exists it will be deleted.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`DeleteOutputFolder

Symbol

FlipBvec is an option for ImportBvalvec.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`FlipBvec

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

NiiDataType is an option of ExportNii. The number type of Nii file can be "Integer", "Real", "Complex", or "Automatic".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`NiiDataType

Symbol

NiiLegacy is an option for ExportNii, if set True default orientations are set instead of unknown.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`NiiLegacy

Symbol

NiiMethod is an option for ImportNIi. Values can be "data", "dataTR", "header", "scaling", "headerMat", "rotation", "all".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`NiiMethod

Symbol

NiiOffset is an option of ExportNii. Is {xoff, yoff, zoff}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`NiiOffset

Symbol

NiiScaling is an option for ImportNii. It scales the nii values with scale slope and offset for quantitative data.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`NiiScaling

Symbol

NiiSliceCode is an option for Export nii. Whith this you can set the slice code of the nii file.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`NiiSliceCode

Symbol

PositiveZ is an options for GradientPlot. If True all Gradients are displayed with a positive z direction.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PositiveZ

Symbol

RotateGradients is an option for ImportNiiDiff.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`RotateGradients

Symbol

UseSubfolders is an option for DcmToNii. If set True the nii conversion is done for each folder in the selected input folder.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`UseSubfolders

Symbol

UseVersion is an option for DcmToNii. For windows it allows to switch between different versions of dcm2niix.exe.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`UseVersion

PhysiologyTools

Functions

Symbol

AlignRespLog[physLog, respirect, scanTime] aligns respirect and physlog data. physLog is output from ImportPhyslog.

resirect is the first output from ImportRespirect.

Documentation Local »

Default Definitions SyntaxInformation[AlignRespLog] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options {OutputMethod → val, SampleStep → 0.005}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PhysiologyTools`AlignRespLog

Symbol

ImportPhyslog[] imports all physlog files from the folder selcted.

ImportPhyslog["forder"] imports all physlog files from "folder" selcted.

Documentation Local »

Default Definitions SyntaxInformation[ImportPhyslog] = {ArgumentsPattern → {_.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PhysiologyTools`ImportPhyslog

Symbol

ImportRespirect[] impors all the respirect log files from the folder selcted.

ImportRespirect["folder"] impors all the respirect log files from the "folder" selcted.

Documentation Local »

Default Definitions SyntaxInformation[ImportRespirect] = {ArgumentsPattern → {_.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PhysiologyTools`ImportRespirect

Symbol

PlotPhyslog[{time, resp}, {start, stop}] plots the physlog from ImportPhyslog.

PlotPhyslog[{time, resp}, {start, stop}, scanTime] plots the physlog from ImportPhyslog.

Documentation Local »

Default Definitions SyntaxInformation[PlotPhyslog] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PhysiologyTools`PlotPhyslog

Symbol

PlotRespiract[data, dataP, scantimes] plots the respirect data to correct peaks. data and dataP are the first outputs of ImportResirect. scantimes is the output from AlignRespLog.

PlotRespiract[data, dataP, scantimes, steps].

Documentation Local »

Default Definitions SyntaxInformation[PlotRespiract] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PhysiologyTools`PlotRespiract

Options

Symbol

OutputMethod can be "val" or "plot".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PhysiologyTools`OutputMethod

Symbol

SampleStep is an option for AlignRespiract.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PhysiologyTools`SampleStep

PlottingTools

Functions

Symbol

GetSliceData[data, offsets] gets the slices from the data defined by offsets which are obtained by GetSlicePosisions.

GetSliceData[data, offsets, vox] gets the slices from the data defined by offsets which are obtained by GetSlicePosisions in mm.

The offsets can also be provided manually which is {{AX,..},{COR,..},{SAG,..}}.

Documentation Local »

Default Definitions SyntaxInformation[GetSliceData] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`GetSliceData

Symbol

GetSlicePositions[data] finds the position of slices with the maximal signal in voxel index.

GetSlicePositions[data, vox] find the position of slices with the maximal signal in mm.

Documentation Local »

Default Definitions SyntaxInformation[GetSlicePositions] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options {MakeCheckPlot → False, DropSlices → {1, 1, 1}, PeakNumber → {1, 1, 2}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`GetSlicePositions

Symbol

GradientPlot[bvec, bval] plots the given bvec with position of the gradients scaled according to the bval.

Documentation Local »

Default Definitions SyntaxInformation[GradientPlot] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › PlotSpace → bspace … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`GradientPlot

Symbol

ListSpherePlot[points] plots 3D points as spheres.

Documentation Local »

Default Definitions SyntaxInformation[ListSpherePlot] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {SphereSize → 2, SphereColor → Automatic}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ListSpherePlot

Symbol

MakeSliceImages[imgData] generates images from the imgData which is obtained form GetSliceData.

MakeSliceImages[imgData, vox] generates images from the imgData which is obtained form GetSliceData, vox is used for the correct aspect ratio of the images.

MakeSliceImages[imgData, {labData, labels}] generates images from the imgData which is obtained
form GetSliceData with an overlay of the segmentations in labData, which can also be obtained using GetSliceData on the segmentations.

labels should be the label numbers used in the original segmentation (to allow correct scaling between slices).

MakeSliceImages[imgData, {labData, labels},vox] generates images from the imgData which is obtained form GetSliceData with an overlay
of the segmentations in labData, which can also be obtained using GetSliceData on the segmentations, vox is used for the correct aspect ratio of the images.

Documentation Local »

Default Definitions SyntaxInformation[MakeSliceImages] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › PlotRange → Automatic ... (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`MakeSliceImages

Symbol

PlotContour[data, vox] creates a contour of the data.

PlotContour[data, vox, scale] creates a contour of the data with the surface colored acording to scale.

PlotContour[data, vox, scale, range] creates a contour of the data with the surface colored acording to scale with a fixed plotrange.

Documentation Local »

Default Definitions SyntaxInformation[PlotContour] = {ArgumentsPattern → {_, _, _, _, OptionsPattern[]}}

Options ContourStyle → {■, 0.25}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotContour

Symbol

PlotCorrection[w] plots deformation vectors w {w1,w2..} generated by Registration2D and Registration3D for multiple datasets or registration steps.

Documentation Local »

Default Definitions SyntaxInformation[PlotCorrection] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotCorrection

Symbol

PlotData[data] plots the data.

PlotData[data, vox] plots the data and for 3D and 4D data assumes the voxelsize vox (z,x,y).

PlotData[data1, data2] plots data1 and data2.

PlotData[data1, data2, vox] plots data1 and data2 and for 3D and 4D data assumes the voxelsize vox (z,x,y).

Documentation Local »

Default Definitions SyntaxInformation[PlotData] = {ArgumentsPattern → {_, _., _., OptionsPattern[]}}

Options {PlotRange → Auto, ColorFunction → BlackToWhite}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotData

Symbol

PlotData3D[data,vox] is a 3D dataviewer, data is the 3D dataset and voxsize the size of the voxels in mm (z,x,y).

Documentation Local »

Default Definitions SyntaxInformation[PlotData3D] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotData3D

Symbol

PlotDefGrid[data, phasemap, shiftpar] plots the dataset on the background with on top the non deformed and the deformed grid, or arrows or lines.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotDefGrid

Symbol

PlotDuty[{grad, bval, ord}, mode] plot the gradient dutycycle.

Documentation Local »

Default Definitions SyntaxInformation[PlotDuty] = {ArgumentsPattern → {{_, _, _}, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotDuty

Symbol

PlotIVIM[vals, data, bvals] plots the results of the IVIM fits from IVIMCalc or BayesianIVIMFit2 or Baye.

Documentation Local »

Default Definitions SyntaxInformation[PlotIVIM] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › Method → … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotIVIM

Symbol

PlotMoments[{G(t),..}, te, t] plots the moments generated by CalculateMoments.

Documentation Local »

Default Definitions SyntaxInformation[PlotMoments] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotMoments

Symbol

PlotSequence[seq,var] where seq is the output from GradSeq.

Documentation Local »

Default Definitions SyntaxInformation[PlotSequence] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotSequence

Options

Symbol

ColorFunction is an option for graphics functions that specifies a function to apply to determine colors of elements.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ColorFunction

Symbol

ContourStyle is an option for contour plots that specifies the style in which contour lines or surfaces should be drawn.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ContourStyle

Symbol

DropSlices is an option for GetSlicePositions and specifies how many slices from the beginning and and should be ignored.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`DropSlices

Symbol

ImageLegend is an option for MakeSliceImages, if set true a barlegend is added to the image.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ImageLegend

Symbol

ImageOrientation is an option for MakeSliceImages. Can be Automatic, "Vertical" or "Horizontal".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ImageOrientation

Symbol

ImageSize is an option that specifies the overall size of an image to display for an object.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ImageSize

Symbol

MakeCheckPlot is an option for GetSlicePositions and if set true gives a plot of the slices locations.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`MakeCheckPlot

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

NormalizeIVIM is an option for IVIMplot. If True the signal at b=0 is 1.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`NormalizeIVIM

Symbol

PeakNumber is an option of GetSlicePostitions and specifies how many slices per direction need to be found.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PeakNumber

Symbol

PlotColor is an option for GradientPlot can be any color or gradient color name.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotColor

Symbol

PlotRange is an option for graphics functions that specifies what range of coordinates to include in a plot.

Documentation Local » | Web »

Attributes {Protected, ReadProtected}

Full Name System`PlotRange

Symbol

PlotSpace is an option for GradientPlot can be "bspace" or "qspace".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotSpace

Symbol

PositiveZ is an options for GradientPlot. If True all Gradients are displayed with a positive z direction.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PositiveZ

Symbol

SphereColor ListSpherePlor. Default value is Automatic, If a color is given this color will be used for all spheres.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`SphereColor

Symbol

SphereSize is an option for GradientPlot and ListSpherePlor. Sets the size of the spheres thar represent the gradients.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`SphereSize

ProcessingTools

Functions

Symbol

B1MapCalc[data, TR, alpha] calculates the B1 map from a dual TR {tr1, tr2} acquisition (AFI) using magnitude data with reference angle alpha.

data has dimensions {z, {tr1,tr2}, x, y}.

B1MapCalc[dataTr1, dataTr2, TR, alpha] where dataTr1 and and dataTr2 can have any dimensions.

The Output can be "Map", "MagPhase", or "Complex"}

B1MapCalc[] is based on DOI: 10.1002/mrm.21120.

Documentation Local »

Default Definitions SyntaxInformation[B1MapCalc] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › B1Output → Map … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`B1MapCalc

Symbol

B1Shimming[b10, b190, mask] finds the optimal shim values to shim to 100% b1. Assumes B1Scaling "Relative".

B1Shimming[b10, b190, mask, target] finds the optimal shim values to shim to target, which can be a number or a map.

Documentation Local »

Default Definitions SyntaxInformation[B1Shimming] = {ArgumentsPattern → {_, _, _, _., OptionsPattern[]}}

Options › B1ShimMethod → All … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`B1Shimming

Symbol

CobineB1[b10,b190,{f1,f2,ang}] combines the complex b1 maps with relative amplitudes f1 and f2 using phase angle ang.

Documentation Local »

Default Definitions SyntaxInformation[CombineB1] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options B1Scaling → Relative

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`CombineB1

Symbol

CorrectJoinSetMotion[{{dat1, dat2, ...}, vox, over] motion correts multiple sets with overlap. Over is the number of slices overlap between stes. A Translation registration is performed.

Documentation Local »

Default Definitions SyntaxInformation[CorrectJoinSetMotion] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options {JoinSetSplit → True, PaddOverlap → 2}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`CorrectJoinSetMotion

Symbol

DataTransformation[data, vox, w] transforms a 3D dataset accordint to the affine transformation vector w.

Documentation Local »

Default Definitions SyntaxInformation[DataTransformation] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options InterpolationOrder → 1

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`DataTransformation

Symbol

DatTot[{data1, data2, ..}, name, vox] calculates the parameter table conating the volume, mean, std and 95 CI for each of the diffusion parameters.

Documentation Local »

Default Definitions SyntaxInformation[DatTot] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`DatTot

Symbol

DatTotXLS[{data1, data2, ..}, name, vox] is the same as DatTot, but gives the parameters as strings for easy export to excel.

Documentation Local »

Default Definitions SyntaxInformation[DatTotXLS] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`DatTotXLS

Symbol

ErrorPlot[data, xdata] plots a errorplot of the data where the first dim of the data is the xrange which matches the xdata list.

ErrorPlot[data, xdata, range] similar with a given y range.

Documentation Local »

Default Definitions SyntaxInformation[ErrorPlot] = {ArgumentsPattern → {_, _, _., OptionsPattern[]}}

Options › ColorValue → {■, ■} … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`ErrorPlot

Symbol

FiberDensityMap[fiberPoins, dim, vox] generates a fiber density map for the fiberPoins which are imported by LoadFiberTracts.

The dimensions dim should be the dimensions of the tracked datasets van vox its volxel size.

Documentation Local »

Default Definitions SyntaxInformation[FiberDensityMap] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options SeedDensity → Automatic

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`FiberDensityMap

Symbol

FiberLengths[fpoints, flines] calculates the fiber lenght using the output from LoadFiberTacts.

FiberLengths[{fpoints, flines}] calculates the fiber lenght using the output from LoadFiberTacts.

Documentation Local »

Default Definitions SyntaxInformation[FiberLengths] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`FiberLengths

Symbol

FindOutliers[data] finds the outliers of a list of data.

Documentation Local »

Default Definitions SyntaxInformation[FindOutliers] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › OutlierMethod → IQR … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`FindOutliers

Symbol

FitData[data,range] converts the data into 100 bins within the +/− range around the mean. Function is used in ParameterFit.

Documentation Local »

Default Definitions SyntaxInformation[FitData] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`FitData

Symbol

GetMaskMeans[dat, mask, name] calculates the mean, std, 5,50 and 95% CI form the given data for each of the given masks.

Mask can be genereated by SplitSegmentations. name is a string that is added to the header.

Documentation Local »

Default Definitions SyntaxInformation[GetMaskMeans] = {ArgumentsPattern → {_, _, _., OptionsPattern[]}}

Options {MeanMethod → SkewNormalDist, Method → Automatic}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`GetMaskMeans

Symbol

Hist[data, range] plots a probability density histogram of the data from xmin to xmax with a fitted (skew)normal distribution. Uses ParameterFit.

Hist[data, range, label] plots a probability density histogram of the data from xmin to xmax with a fitted (skew)normal distribution and label as x–axis label.

Hist[{data1.., data2,..}, {range1, range2,..}] plots a probability density histogram of the data from xmin to xmax with a fitted (skew)normal distribution. Uses ParameterFit.

Hist[{data1, data2,..}, {range1, range2,..}, {label1, label2,..}] plots a probability density histogram of the data from xmin to xmax with a fitted (skew)normal distribution and label as x–axis label.

Documentation Local »

Default Definitions SyntaxInformation[Hist] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › ColorValue → {{■, □}, ■, ■, ■} … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`Hist

Symbol

Hist2[pars, range] plots a probability density histogram of the data over range with two fitted (skew)normal distribution. Uses ParameterFit2.

Hist2[pars, range, label] plots a probability density histogram of the data over range with two fitted (skew)normal distribution. Uses ParameterFit2.

Documentation Local »

Default Definitions SyntaxInformation[Hist2] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options Scaling → False

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`Hist2

Symbol

InvertDataset[data] inverts the data along the x y and z axes. In other words it is rotated aroud the origin such that (x,y,z)=(–x,–y,–z) and (0,0,0)=(0,0,0).

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`InvertDataset

Symbol

JoinSets[{dat1, dat2, ...}, over] joins dat1, dat2, ... with over slices overlap.

JoinSets[{dat1, dat2, ...},{over1, over2, ...}] joins dat1 and dat2 with over1 slices overlap, Joins dat2 and dat3 with over2 slices overlap and so on.

JoinSets[{dat1, dat2, ...},{{over, drop1, drop2}, ...}] joins dat1, dat2 with over slices overlap and drops drop1 slices for dat1 and drop2 from drop 2.

DOI: 10.1148/radiol.14140702.

Documentation Local »

Default Definitions SyntaxInformation[JoinSets] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options › ReverseSets → True ... (7 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`JoinSets

Symbol

MeanRange[Range] calculates the medain (50%) and standard deviation (14% and 86%) range and reports it as a string.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`MeanRange

Symbol

MeanStd[data] calculates the mean and standard deviation and reports it as a string.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`MeanStd

Symbol

MedCouple[data] calculates the medcouple of a list of data.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`MedCouple

Symbol

NumberTableForm[data] makes a right aligned table of the numbers with 3 decimal percision.

NumberTableForm[data, n] makes a right aligned table of the numbers with n decimal percision.

Documentation Local »

Default Definitions SyntaxInformation[NumberTableForm] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › TableMethod → NumberForm … (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`NumberTableForm

Symbol

ParameterFit[data] fits a (skew)Normal probability density function to the data.

ParameterFit[{data1, data2,...}] fits a (skew)Normal probability density function to each of the datasets. Is used in Hist.

Documentation Local »

Default Definitions SyntaxInformation[ParameterFit] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {FitFunction → SkewNormal, FitOutput → Parameters, Method → Automatic}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`ParameterFit

Symbol

ParameterFit2[data] fits two skewNormal probaility density fucntions to the data. Assuming two compartments, one for fat and one for muscle.

Documentation Local »

Default Definitions SyntaxInformation[ParameterFit2] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options FitOutput → BestFitParameters

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`ParameterFit2

Symbol

RotateData[data] rotates the data 180 degree, e.g. inversion of the z direction.

Documentation Local »

Default Definitions SyntaxInformation[RotateData] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`RotateData

Symbol

RotateTensor[tens] rotates the tensor 180 degree, e.g. inversion of the z direction with also inversing the tensor orientation.

Documentation Local »

Default Definitions SyntaxInformation[RotateTensor] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`RotateTensor

Symbol

SetupDataStructure[dcmFolder] makes nii folders and generates nii files for a directory of dmc data where the data is structured per subject.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SetupDataStructure

Symbol

SmartMask[input] crates a smart mask of input, which is either the tensor or the tensor parameters calculated using ParameterCalc.

SmartMask[input, mask] crates a smart mask of input and used the mask as a prior selection of the input.

Documentation Local »

Default Definitions SyntaxInformation[SmartMask] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › Strictness → 0.5 … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SmartMask

Symbol

SNRCalc[data,masksig,masknoise] calculates the Signal to noise ratio of the signal selected by masksig and the noise selected by masknoise.

Documentation Local »

Default Definitions SyntaxInformation[SNRCalc] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SNRCalc

Symbol

SNRMapCalc[data1,noisemap] calcualtes the signal to noise ratio of the data using MN[data]/(1/sqrt[pi/2] sigma),
where sigma is the local mean of the noise map assuming it is a rician distribution.

SNRMapCalc[{data1,data2}] calcualtes the signal to noise ratio from two identical images using
MN[data1,data2] / (.5 SQRT[2] STDV[data2−data1]).

SNRMapCalc[{data1, .. dataN}] calcualtes the signal to noise ratio of the data using MN/sigma where the mean signal MN is the average voxel value over all dynamics N and the sigma is the standard deviation over all dynamics N.

Documentation Local »

Default Definitions SyntaxInformation[SNRMapCalc] = {ArgumentsPattern → {_, _., _., OptionsPattern[]}}

Options {OutputSNR → SNR, SmoothSNR → 2}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SNRMapCalc

Symbol

SplitSets[data, Nsets, Nover] splits the data in Nsets with Nover slices overlap.

Documentation Local »

Default Definitions SyntaxInformation[SplitSets] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options {ReverseSets → False, ReverseData → True, PaddOverlap → 0}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SplitSets

Options

Symbol

AxesLabel is an option for graphics functions that specifies labels for axes.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`AxesLabel

Symbol

B1EqualPower is an option for B1shimming. If true equal power for both channels is used.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`B1EqualPower

Symbol

B1FilterData is an option for B1MapCalc. If True HammingFilterData is applied to the data before B1 calculation.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`B1FilterData

Symbol

B1Masking is an option for B1MapCalc. If True then values where S2 is larger than S1 are masked.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`B1Masking

Symbol

B1MaxPower is an option for B1Shimming. Specifies how much power can be used per channel.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`B1MaxPower

Symbol

B1Output is an option for B1MapCalc. Values can be "Map", "MagPhase", or "Complex".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`B1Output

Symbol

B1Scaling is an option for B1Shimming and CombineB1. Values can be "Relative" or "Absolute". "Absolute" assurmes b1 maps are given in uT, "Relative" assumes that maps are in %.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`B1Scaling

Symbol

B1ShimMethod is an option for B1Shimming. Values can be "All", "Phase" or "Magnitude".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`B1ShimMethod

Symbol

ColorValue is an option for Hist and ErrorPlot. Default {Black, Red}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`ColorValue

Symbol

FitFunction is an option for ParameterFit. Options are "Normal" or "SkewNormal". Indicates which function wil be fitted.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`FitFunction

Symbol

FitOutput is an option for ParameterFit and ParameterFit2. Option can be "Parameters", "Function" or "BestFitParameters".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`FitOutput

Symbol

ImageSize is an option that specifies the overall size of an image to display for an object.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ImageSize

Symbol

InterpolationOrder is an option for Interpolation, as well as ListLinePlot, ListPlot3D, ListContourPlot, and related functions, that specifies what order of interpolation to use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`InterpolationOrder

Symbol

JoinSetSplit is an option ofr CorrectJoinSetMotion. If True RegisterDataTransformSplit is used else RegisterDataTransform is used.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`JoinSetSplit

Symbol

MaskCompartment is an option for SmartMask. Can be "Muscle" or "Fat".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`MaskCompartment

Symbol

MeanMethod is an option for GetMaskMeans. The option can be "NormalDist", "SkewNormalDist", or "Mean".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`MeanMethod

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

MotionCorrectSets is an option for JoinSets. True motion corrects the individual stacs before joining using CorrectJoinSetMotion.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`MotionCorrectSets

Symbol

NormalizeOverlap is an option for JoinSets. True removes strong signal dropoff at the end of a stack.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`NormalizeOverlap

Symbol

NormalizeSets is an option for JoinSets. True normalizes the individual stacs before joining.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`NormalizeSets

Symbol

OutlierIncludeZero is an option for FindOutliers. If set to True all values that are zero are ignored and considered outliers.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`OutlierIncludeZero

Symbol

OutlierIterations is an option for FindOutliers. Specifies how many iterations are used to find the outliers.

Each itteration the outliers are reevaluated on the data with the previously found outliers alread rejected.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`OutlierIterations

Symbol

OutlierMethod is an option for FindOutliers. values can be "IQR", "SIQR" or "aIQR". "IRQ" is used for normly distributed data, "SIQR" or "aIQR" are better for skewed distributions.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`OutlierMethod

Symbol

OutlierOutput is an option for FindOutliers. If value is "Mask" it gives a list of 1 for data and 0 for outliers. Else the output is {data, outliers}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`OutlierOutput

Symbol

OutlierRange is an option for FindOutliers. Specifies how many times the IQR is considred an oulier.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`OutlierRange

Symbol

OutputSNR is an option for SNRMapCalc.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`OutputSNR

Symbol

PaddOverlap is an option of CorrectJoinSetMotion and JoinSets. it allows for extra motion in the z direction.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`PaddOverlap

Symbol

PlotLabel is an option for graphics functions that specifies an overall label for a plot.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`PlotLabel

Symbol

ReferenceB1 is an option for B1MapCalc. Default value is None. Can be given a numeric value in uT.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`ReferenceB1

Symbol

ReverseData is an option for JoinSets. Reverses each individual datset given as input for the JoinSets function. True by default.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`ReverseData

Symbol

ReverseSets is an option for JoinSets. Reverses the order of the datsets, False by default.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`ReverseSets

Symbol

Scaling is an option for Hist2. Scales the individual fits of the fat and muscle compartment.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`Scaling

Symbol

SeedDensity is an option for FiberDensityMap. The seedpoint spacing in mm.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SeedDensity

Symbol

SmartMaskOutput is an option for Smartmask. Can be set to "mask" to output only the mask or "full" to also output the probability mask.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SmartMaskOutput

Symbol

SmartMethod is an option for SmartMask. This specifies how the mask is generated. Can be "Continuous" or "Catagorical".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SmartMethod

Symbol

SmoothSNR is an option for SNRMapCalc.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SmoothSNR

Symbol

Strictness is an option for SmartMask value between 0 and 1. Higer values removes more data.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`Strictness

Symbol

TableAlignments is an option for TableForm and MatrixForm which specifies how entries in each dimension should be aligned.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`TableAlignments

Symbol

TableDepth is an option for TableForm and MatrixForm that specifies the maximum number of levels to be printed in tabular or matrix format.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`TableDepth

Symbol

TableDirections is an option for TableForm and MatrixForm which specifies whether successive dimensions should be arranged as rows or columns.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`TableDirections

Symbol

TableHeadings is an option for TableForm and MatrixForm that gives the labels to be printed for entries in each dimension of a table or matrix.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`TableHeadings

Symbol

TableMethod is an option for NumberTableForm. It specifies which number form to uses. Values can be NumberForm, ScientificForm or EngineeringForm.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`TableMethod

Symbol

TableSpacing is an option for TableForm and MatrixForm that specifies how many spaces should be left between each successive row or column.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`TableSpacing

ReconstructionTools

Functions

Symbol

CoilCombine[sig] combines the coil signals sig. Where sig is {nCoils, ...}.

CoilCombine[sig, cov] combines the coil signals sig. Where sig is {nCoils, ...} and cov the complex noise correlation matrix.

CoilCombine[sig, cov, sens] combines the coil signals sig. Where sig is {nCoils, ...} and cov the complex noise correlation matrix and sense the coils sensitivitys.

Possible coil combination methods are "Sum", "RootSumSqaures", "RoemerEqualNoise", "RoemerEqualSignal", "WSVD".

RootSumSquares needs the signal. Can be performed with and without the noise covaricance matrix

RoemerEqualNoise needs the signal and the noise covaricance matrix. Can be performed with

and without the sense data, without sense data the sensisity is estimated using the singal and the RSS reconstrucntion of the signa.

RoemerEqualSignal needs the signal and the noise covaricance matrix and the sense data.

WSVD needs the signal and the noise covariance matrix.

Documentation Local »

Default Definitions SyntaxInformation[CoilCombine] = {ArgumentsPattern → {_, _., _., OptionsPattern[]}}

Options {Method → RoemerEqualNoise, SenseRescale → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`CoilCombine

Symbol

CoilWeightedRecon[kspace, noise, head] performs reconstuction of raw MS2D MRI data. The input kspace, noise and head are obtained using ReadListData.

The coil combination Methods can be "Roemer" or "RSS".

Documentation Local »

Options › EchoShiftData → 0 … (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`CoilWeightedRecon

Symbol

CoilWeightedReconCSI[kspace, noise, head] performs reconstuction of raw 3DCSI data. The input kspace, noise and head are obtained using ReadListData.

The coil combination Methods can be "Roemer" or "WSVD".

Documentation Local »

Options › HammingFilter → False … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`CoilWeightedReconCSI

Symbol

DeconvolveCSIdata[spectra] deconvolves the CSI spectra after HammingFilterCSI to revert the blurring of the hammingfiltering.

DeconvolveCSIdata[spectra, ham] deconvolves the CSI spectra with the acquired weighting ham to revert the blurring of the kspace weighting.

Documentation Local »

Default Definitions SyntaxInformation[DeconvolveCSIdata] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options {WienerRegularization → 0.007, DeconvolutionMethod → Wiener}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`DeconvolveCSIdata

Symbol

FourierKspace2D[kspace,head] performs a 2D reconstruction of 2D kspace data. Where kspace and head are generated by ReadListData.

Documentation Local »

Default Definitions SyntaxInformation[FourierKspace2D] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`FourierKspace2D

Symbol

FourierKspace3D[kspace,head] performs a 3D reconstruction of 3D kspace data. Where kspace and head are generated by ReadListData.

Documentation Local »

Default Definitions SyntaxInformation[FourierKspace3D] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`FourierKspace3D

Symbol

FourierKspaceCSI[kspace,head] performs a 3D reconstruction of 3D CSI kspace data. Where kspace and head are generated by ReadListData.

Documentation Local »

Default Definitions SyntaxInformation[FourierKspaceCSI] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`FourierKspaceCSI

Symbol

FourierRescaleData[data] rescales the data to double the dimensions using zeropadding in fourier space.

FourierRescaleData[data, facotr] rescales the data to factor times the dimensions using zeropadding in fourier space.

Documentation Local »

Default Definitions SyntaxInformation[FourierRescaleData] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`FourierRescaleData

Symbol

FourierShift[data] shift the data to the right by half the data dimensions.

Documentation Local »

Default Definitions SyntaxInformation[FourierShift] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`FourierShift

Symbol

FourierShifted[kspace] shifts the kspace half the kspace dimensions and then performs a FourierTransform on the kspace.

Documentation Local »

Default Definitions SyntaxInformation[FourierShifted] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`FourierShifted

Symbol

HammingFilterCSI[kspace] apllies a Hammingfilter to the k−space data. The data can be can be 1D, 2D or 3D, the spectral dimensions is the last dimensions (x,y,z, spectra).

Documentation Local »

Default Definitions SyntaxInformation[HammingFilterCSI] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`HammingFilterCSI

Symbol

HammingFilterData[kspace] apllies a Hammingfilter to the data. The data is in immage space and can be 1D, 2D or 3D.

Documentation Local »

Default Definitions SyntaxInformation[HammingFilterData] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`HammingFilterData

Symbol

InverseFourierShift[data] shift the data to the left by half the data dimensions.

Documentation Local »

Default Definitions SyntaxInformation[InverseFourierShift] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`InverseFourierShift

Symbol

InverseFourierShifted[data] performs a InverseFourierTransform on the data and then shifts the kspace half the kspace dimensions.

Documentation Local »

Default Definitions SyntaxInformation[InverseFourierShifted] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`InverseFourierShifted

Symbol

MakeHammingFilter[xdim] makes a 1D HammingKernel for filtering k−space.

MakeHammingFilter[{xdim}] makes a 1D HammingKernel for filtering k−space.

MakeHammingFilter[{xdim, ydim}] makes a 2D HammingKernel for filtering k−space in 2D CSI data of size {xdim, ydim}.

MakeHammingFilter[{xdim, ydim, zdim}] makes a 3D HammingKernel for filtering k−space in 3D CSI data of size {xdim, ydim, zdim}.

Documentation Local »

Default Definitions SyntaxInformation[MakeHammingFilter] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`MakeHammingFilter

Symbol

MakeSense[coils, cov] makes a sense map for coils. Each coil signal is devided by the RSS reconstuction of the coils.

Documentation Local »

Default Definitions SyntaxInformation[MakeSense] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options SenseRescale → True

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`MakeSense

Symbol

MeanType[kspace, types, type] calcualtes the Mean of the kspace data on type, where type is part of types. The kspace and types are generated by ReadListData.

MeanType[{kspace, types}, type] calcualtes the Mean of the kspace data on type, where type is part of types.

MeanType[kspace, types, {type,..}] calcualtes the Mean of the kspace data on each of the list type, where type is part of types.

Output is {kspace, types}.

Documentation Local »

Default Definitions SyntaxInformation[MeanType] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`MeanType

Symbol

NoiseCorrelation[noise] calculates the noise correlation matrix, noise is {nrCoils, noise Samples}.

Documentation Local »

Default Definitions SyntaxInformation[NoiseCorrelation] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`NoiseCorrelation

Symbol

NoiseCovariance[noise] calculates the noise covariance matrix, noise is {nrCoils, noise Samples}.

Documentation Local »

Default Definitions SyntaxInformation[NoiseCovariance] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`NoiseCovariance

Symbol

NormalizeSpectra[spec] normalizes spectra to be scaled to the max value of the absolute signal = 1000. Can be any dimension.

Documentation Local »

Default Definitions SyntaxInformation[NormalizeSpectra] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`NormalizeSpectra

Symbol

OrderKspace[kspace, types, order] reorders the kspace data to order, where order is a list and each value is a part of types. The kspace and types are generated by ReadListData.

Documentation Local »

Default Definitions SyntaxInformation[OrderKspace] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`OrderKspace

Symbol

ReadListData[file] reads a list/data raw data file from the philips MR platform. The input file can either be .list or .data file.

Ouput is {{rawData, noise}, {head, types}}.

ReadListData[file, print] does the same but if print is set False no reporting is done.

Documentation Local »

Default Definitions SyntaxInformation[ReadListData] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`ReadListData

Symbol

SagitalTranspose[data] makes a transpose of the data of the second level ande reverses the slices.

Documentation Local »

Default Definitions SyntaxInformation[SagitalTranspose] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`SagitalTranspose

Symbol

ShiftedFourier[kpace] performs a FourierTransform on the kspace and then shifts the data half the data dimensions.

Documentation Local »

Default Definitions SyntaxInformation[ShiftedFourier] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`ShiftedFourier

Symbol

ShiftedInverseFourier[data] shifts the data half the data dimensions and then performs a InverseFourierTransform on the data.

Documentation Local »

Default Definitions SyntaxInformation[ShiftedInverseFourier] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`ShiftedInverseFourier

Symbol

TotalType[kspace, types, type] calcualtes the Total of the kspace data on type, where type is part of types. The kspace and types are generated by ReadListData.

TotalType[{kspace, types}, type] calcualtes the Total of the kspace data on type, where type is part of types.

TotalType[kspace, types, {type,..}] calcualtes the Total of the kspace data on each of the list type, where type is part of types.

Output is {kspace, types}.

Documentation Local »

Default Definitions SyntaxInformation[TotalType] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`TotalType

Options

Symbol

AcquisitionMethod is an option for CoilWeightedReconCSI. Values can be "Fid" or "Echo".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`AcquisitionMethod

Symbol

CoilSamples is an option for CoilWeightedReconCSI and specifies how many fud samples are used to calcualte the coil sensitivity for Roemer reconstruction.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`CoilSamples

Symbol

DeconvolutionMethod is an option for DeconvolveCSIdata. It specifies which deconvolution method to used.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`DeconvolutionMethod

Symbol

EchoShiftData is an option for CoilWeightedRecon.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`EchoShiftData

Symbol

HammingFilter is an option for CoilWeightedReconCSI. If True it applies a spatial hamming filter to the data.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`HammingFilter

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

NormalizeOutputSpectra is an option for CoilWeightedReconCSI.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`NormalizeOutputSpectra

Symbol

OutputSense is an option for CoilWeightedRecon. If set true the function will also output the used Sense map.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`OutputSense

Symbol

QMRITools`ReconstructionTools`Private`ReconFilter

Full Name QMRITools`ReconstructionTools`Private`ReconFilter

Symbol

RescaleRecon is an option for CoilWeightedRecon. If set true the data will be scaled to the range 0–1000.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`RescaleRecon

Symbol

SenseRescale is an option for MakeSense. If set True the data is first downscaled by a factor 2 before making the sense map.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`SenseRescale

Symbol

WienerRegularization is an option for DeconvolveCSIdata. It defines te amount of regularization used in the wiener deconvoltuion.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`WienerRegularization

RelaxometryTools

Functions

Symbol

CalibrateEPGT2Fit[datan, times, angle] calculates the Fat T2 ralaxation that will be used in the EPGT2fit.

Outputs the fat T2 value.

Documentation Local »

Default Definitions SyntaxInformation[CalibrateEPGT2Fit] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › EPGRelaxPars → {{0, 100}, {20, 300}, {1400., 365.}} … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`CalibrateEPGT2Fit

Symbol

CreateT2Dictionary[{T1m, T1f}, {Necho, detlaTE}, angle] Creates a EPG signal dictionary used for EPGT2fit.

Every dictionary that is defined is cached.

The output is in units as defined by the detlaTE, e.g. if detlaTE is in ms the output is in ms.

The TR and TE should be in the same units as Dela.

Output is {dictionary, vals}.

Documentation Local »

Default Definitions SyntaxInformation[CreateT2Dictionary] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › DictB1Range → {0.5, 1.4, 0.01} … (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`CreateT2Dictionary

Symbol

DictionaryMinSearch[dictionary, y] performs dictionary minimization of data y. dictionary is generated with CreateT2Dictionary.

Output is {{T2, B1}, fwfraction, residualError}.

Documentation Local »

Default Definitions SyntaxInformation[DictionaryMinSearch] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`DictionaryMinSearch

Symbol

EPGSignal[{Necho, echoSpace}, {T1, T2}, {ex_angle,ref_angle}, B1] generates a EPG T2 curve with stimulated echos.

T1, T2 and echoSpace are in ms, angel is in degree, B1 is between 0 and 1.

Output is the EPG Signal vector.

EPGSignal[] is based on DOI: 10.1002/jmri.24619.

Documentation Local »

Default Definitions SyntaxInformation[EPGSignal] = {ArgumentsPattern → {_, _, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGSignal

Symbol

EPGT2Fit[data, {Necho, detlaTE}, {exitation, refoucs}] fits the T2 based on Marty B et.al.

Simultaneous muscle water T2 and fat fraction mapping using transverse relaxometry with stimulated echo compensation.

Exitation and refocus are the RF pulse angles e.g. 90,180. They can also be a range of angeles over the slice profile as defined by GetSliceProfile.

The output is in units as defined by the detlaTE, e.g. if detlaTE is in ms the output is in ms.

The exitation and refocus are defined in Degrees.

Output is {{{T2map,B1Map},{wat, fat, fatMap}, residual},callibration} or {{T2map,B1Map},{wat, fat, fatMap}, residual}.

EPGT2Fit[] is based on DOI: 10.1002/nbm.3459.

Documentation Local »

Default Definitions SyntaxInformation[EPGT2Fit] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options › EPGRelaxPars → {1400., 365.}… (17 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGT2Fit

Symbol

NonLinearEPGFit[{vals, T2cons}, y] performs dictionary minimization of data y. vals = {{T1muscle, T1fat, T2fat}, {Necho, echoSpace, angle}}.

Output is {{T2, B1}, fwfraction, residualError}.

Documentation Local »

Default Definitions SyntaxInformation[NonLinearEPGFit] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`NonLinearEPGFit

Symbol

ShiftPulseProfile[angs, shift] shifts the reference pulse profile by shift and makes the.

ans = {exitation, refocus} as generated by GetPulseProfile. Shift is the shift in sample points.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`ShiftPulseProfile

Symbol

T1Fit[data, TR] fits the T1 value to the data using a nonlinear method. The output is in units as defined by the TR, e.g. if TR is in ms the TR is in ms. Output is {t1, apar, bpar}.

Documentation Local »

Default Definitions SyntaxInformation[T1Fit] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`T1Fit

Symbol

T1rhoFit[data, EchoTimes] fits the T1rho value to the data using linear or nonlinear methdos.

The output is in units as defined by the EchoTimes, e.g. if EchoTimes is in ms the output is in ms.

Output is {S(0), T1rhomap}.

Documentation Local »

Default Definitions SyntaxInformation[T1rhoFit] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options Method → Linear

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`T1rhoFit

Symbol

T2Fit[data, EchoTimes] fits the T2 value to the data using linear or nonlinear

methods.vThe output is in units as defined by the EchoTimes, e.g. if EchoTimes is in ms the output is in ms.vOutput is {S(0), T2}.

Documentation Local »
Default Definitions SyntaxInformation[T2Fit] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options Method → Linear
Attributes {Protected, ReadProtected}
Full Name QMRITools`RelaxometryTools`T2Fit

Symbol

TriExponentialT2Fit[data, EchoTimes] fits the T2 based on Azzabou N et.al. Validation

of a generic approach to muscle water T2 determination at 3T in fat–infiltrated skeletal muscle. J. Magn. Reson. 2015.

The fat T2 parameters are automatically estimated from the high signal voxels from the last echo.

The output is in units as defined by the EchoTimes, e.g. if EchoTimes is in ms the output is in ms.

The output fraction is between 0 an 1.

Output is {{S(0), fatFraction, muscleFraction, T2map},callibration} or {S(0), fatFraction, muscleFranction, T2map}.

Documentation Local »
Default Definitions SyntaxInformation[TriExponentialT2Fit] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options OutputCalibration → False
Attributes {Protected, ReadProtected}
Full Name QMRITools`RelaxometryTools`TriExponentialT2Fit

Options

Symbol

DictB1Range is an option for CreateT2Dictionary and EPGT2Fit. It specifies the range and step of the B1 values in the dictionary {min, max, step}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`DictB1Range

Symbol

DictT2fRange is an option for CreateT2Dictionary and EPGT2Fit. is specifies the range and step of the T2 fat values in the dictionary {min, max, step} in ms.

If a single value is given this fixed value is used a long as EPGCalibrate is False.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`DictT2fRange

Symbol

DictT2fValue is an option for EPGFit.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`DictT2fValue

Symbol

DictT2IncludeWater is an options for EPGT2Fit.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`DictT2IncludeWater

Symbol

DictT2Range is an option for CreateT2Dictionary and EPGT2Fit. is specifies the range and step of the T2 values in the dictionary {min, max, step} in ms.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`DictT2Range

Symbol

EPGCalibrate is an option for EPGT2Fit. If set to True it does autmatic callibration of the T2 fat relaxation time.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGCalibrate

Symbol

EPGFatShift is an options for EPGT2Fit. Specfies the amount of shift of the fat refocussing pulse relative to the fat exitation pulse.

Can be obtained form GetPulseProfile.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGFatShift

Symbol

EPGFitFat is an option for EPGT2Fit.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGFitFat

Symbol

EPGFitPoints is a option for CalibrateEPGT2Fit and EPGT2Fit. Number of points is 200 by default.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGFitPoints

Symbol

EPGMethod is an optionf for EPGT2Fit. Values can be "NLLS", "dictionary" or "dictionaryM".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGMethod

Symbol

EPGMethodCal is an option for CalibrateEPGT2Fit and EPGT2Fit. The calibration can be done using "1comp", "2comp", "2compF".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGMethodCal

Symbol

EPGRelaxPars is and option for EPGT2Fit. Needs to be {T1muscl, T1Fat, T2Fat} in ms, defaul is {1400,365,137} in ms.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGRelaxPars

Symbol

EPGSmoothB1 is an options for EPGT2Fit. If set to True the B1 map of the fit will be smoothed after which the minimization if perfomed again but with a fixed B1.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`EPGSmoothB1

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

MonitorEPGFit show waitbar during EPGT2Fit.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`MonitorEPGFit

Symbol

OutputCalibration is an option for EPGT2Fit and TriExponentialT2Fit. If true it outputs the calibartion values.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`OutputCalibration

Symbol

WaterFatShift is an options for EPGT2Fit. It specifies the amount of water fat shift in voxels.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`WaterFatShift

Symbol

WaterFatShiftDirection is an options for EPGT2Fit. It specifies the water fat shift direction: "left", "right", "up" and "down".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`WaterFatShiftDirection

SimulationTools

Functions

Symbol

AddNoise[data, noise] ads rician noise to the data with a given sigma or SNR value.

Documentation Local »

Default Definitions SyntaxInformation[AddNoise] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options {NoiseSize → Sigma, NoiseType → Absolute}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`AddNoise

Symbol

BlochSeries[vectorIn, deltat, freqRange, B1] performs a Bloch simulation of an RF pulse.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`BlochSeries

Symbol

CalculateGfactor[factors, sensitivity, Wmat] calculates a gfactor for given sensitivity maps and noise corraltion W. given the sense factors which is a list of three integers.

Documentation Local »

Default Definitions SyntaxInformation[CalculateGfactor] = {ArgumentsPattern → {_, _, _, _., OptionsPattern[]}}

Options GRegularization → 0.

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`CalculateGfactor

Symbol

CreateDiffData[sig, eig, bvec, gradients, dim] creates a DTI datasets of dimensions dim with sig as unweighted signal S0 and bvec and gradients.

eig can be {l1, l2, l3}, {{l1, l2, l3}, {e1, e2, e3}}, {{l1, l2, l3}, "Random"}, {{l1, l2, l3}, "RandomZ"} or {{l1, l2, l3}, "OrtRandom"}.

Uses Tensor internally.

CreateDiffData[] is based on DOI: 10.1002/nbm.2959.

Documentation Local »

Default Definitions SyntaxInformation[CreateDiffData] = {ArgumentsPattern → {_, _, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`CreateDiffData

Symbol

GESignal[ang, {tr, t1}] calculates the gradient echo signal for flipangles ang using tr and t1.

GESignal[ang_?ListQ, {{tr1_, tr2_}, t1_}] calculates the dual tr gradient echo signal for flipangles ang using tr1, tr2 and t1.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`GESignal

Symbol

GetPulseProfile[excitation, refocus] gives the pusl angle profiles for the exitation and refocussing pulses.

a pulse is defined as {"name", flipangle, {G_strnth, Dur, BW}}.

GetPulseProfile[{"name", flipangle, {G_strnth, Dur, BW}}] gives detaile slice profile information of one pulse.

output is {ex_angle_profiel, ref_angel_profile, {plots}}.

output for single pulse is {{distance, Mt, Mz, Mx, My, ang, phase}, plots}.

Documentation Local »

Default Definitions SyntaxInformation[GetPulseProfile] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options › MagnetizationVector → {0, 0, 1} … (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`GetPulseProfile

Symbol

GfactorSimulation[sensitivity, cov, {dir,sense}] calculates the gfactormaps for given sensitivity maps and noise corraltion cov in one direction.

The sensefactors are a list of integers in a given direction: "LR", "FH", or "AP".

GfactorSimulation[sensitivity, cov, {dir1,sense1}, {dir2,sense2}] calculates the gfactormaps for given sensitivity maps and noise corraltion W in two directions.

Documentation Local »

Default Definitions SyntaxInformation[GfactorSimulation] = {ArgumentsPattern → {_, _, _, _., _., OptionsPattern[]}}

Options {GRegularization → 0., GOutput → Grid}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`GfactorSimulation

Symbol

PlotSimulation[pars, xval, true, label, color] plots the pars (output form Parameters). Using label as PlotLabel and xval as x axis Thics.

tr are the true parameter values. color are the color used for the plot.

Documentation Local »

Default Definitions SyntaxInformation[PlotSimulation] = {ArgumentsPattern → {_, _, _, _, _, OptionsPattern[]}}

Options PlotRange → {{0, 3}, {0, 3}, {0, 3}, {0, 3}, {0, 1}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`PlotSimulation

Symbol

PlotSimulationAngle[par, xdata, label, col] plots pars (output from Anlge Parameters).

Documentation Local »

Options PlotRange → {0, 90}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`PlotSimulationAngle

Symbol

PlotSimulationAngleHist[pars, label, xdata] plots pars (output from Anlge Parameters).

Documentation Local »

Default Definitions SyntaxInformation[PlotSimulationAngleHist] = {ArgumentsPattern → {_, _, _, _, _., OptionsPattern[]}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`PlotSimulationAngleHist

Symbol

PlotSimulationHist[pars, label, xdata, tr] plots the pars (output form Parameters).

Using label as plotlabel and xdata as x axis label. tr are the true parameter values.

Documentation Local »

Default Definitions SyntaxInformation[PlotSimulationHist] = {ArgumentsPattern → {_, _, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`PlotSimulationHist

Symbol

PlotSimulationVec[tens, xdata, label] plots the eigenvectors from simulated tensors.

Documentation Local »

Default Definitions SyntaxInformation[PlotSimulationVec] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options SortVecs → True

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`PlotSimulationVec

Symbol

Pulses[name] gives the pulse shape of some predefinec Philips pulse shapes.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`Pulses

Symbol

Signal[par, TR, TE] calculates the MRI signal at a given TR and TE. Par is defineds as {pd, T1, T2}.

Documentation Local »

Default Definitions SyntaxInformation[Signal] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`Signal

Symbol

SimAngleParameters[tens,vec] caculates the diffusion eigenvectors for tens compared to the true values vec.

The output can be used in PlotSimulationAngleHist and PlotSimulationAngle.

Documentation Local »

Default Definitions SyntaxInformation[SimAngleParameters] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SimAngleParameters

Symbol

SimParameters[tens] caculates the diffusion parameters for tens. The output can be used in PlotSimulationHist and PlotSimulation.

Documentation Local »

Default Definitions SyntaxInformation[SimParameters] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options Reject → False

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SimParameters

Symbol

SimulateDualTR[] simulates the signal of a Dual TR T1 map.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SimulateDualTR

Symbol

SimulateSliceEPG[exitation, refocus, {{T1, T2}, {Necho, echoSp}, b1}] gives a simulated slice profile and EPG singnal plot.

exitation and refocus are generated by GetPulseProfiel.

Documentation Local »

Default Definitions SyntaxInformation[SimulateSliceEPG] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options ReportFits → False

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SimulateSliceEPG

Symbol

Tensor[{l1, l2, l3}] creates a diffuison tensor with vectors {{0,0,1},{0,1,0},{1,0,0}} and eigenvalues {l1, l2, l3}.

Tensor[{l1, l2, l3}, {e1, e2, e3}] creates a diffuison tensor with vectors {e1, e2, e3} and eigenvalues {l1, l2, l3}.

Tensor[{l1, l2, l3}, "Random"] creates a diffuison tensor with random orthogonal eigenvectors {e1, e2, e2} and eigenvalues {l1, l2, l3}.

Tensor[{l1, l2, l3}, "RandomZ"] creates a diffuison tensor with random orthogonal eigenvectors {{1,0,0}, e2, e3} with random eigenvectors and eigenvalues {l1, l2, l3}.

Tensor[{l1, l2, l3}, "OrtRandom"] creates a diffuison tensor with random orthogonal eigenvectors {{1,0,0},{0,1,0},{0,0,1}} and eigenvalues {l1, l2, l3}.

Tensor[] is based on DOI: 10.1002/nbm.2959.

Documentation Local »

Default Definitions SyntaxInformation[Tensor] = {ArgumentsPattern → {_, _., OptionsPattern[]}}

Options TensOutput → Vector

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`Tensor

Options

Symbol

FatFieldStrength is an option for GetPulseProfile. If the value >0 it will calculate the shift of the fat refocussing pulse compared to the fat exitation pulse.

The shift is in SliceRangeSamples steps.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`FatFieldStrength

Symbol

GOutput is an option for GfactorSimulation. can be "Grid" or "List".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`GOutput

Symbol

GRegularization is an option for CalculateGfactor and GfactorSimulation.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`GRegularization

Symbol

MagnetizationVector is an option for GetPulseProfile. It defines the start magnetization vector for the bloch simulation.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`MagnetizationVector

Symbol

NoiseSize is an option for AddNoise. Values can be "Sigma", then the noise sigma is given or "SNR", then the SNR is given.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`NoiseSize

Symbol

NoiseType is an option for AddNoise. Values can be "Absolute" or "Complex", and will add either Rician absolute noise or complex noise to the data.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`NoiseType

Symbol

PlotRange is an option for graphics functions that specifies what range of coordinates to include in a plot.

Documentation Local » | Web »

Attributes {Protected, ReadProtected}

Full Name System`PlotRange

Symbol

Reject is an option for EigenvalCalc. It True then voxels with negative eigenvalues are rejected and set to 0.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`Reject

Symbol

ReportFits is an option for SimulateSliceEPG. If True it also reports the fit values.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`ReportFits

Symbol

SliceRange is an option for GetPulseProfile. It specifies over which range the slice profile is generated (in mm). the total profile is 2xSliceRange.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SliceRange

Symbol

SliceRangeSamples is an option for GetPulseProfile. defines how many samples are used to generate half a puls profile.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SliceRangeSamples

Symbol

SortVecs is an option for PlotSimulationVec.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SortVecs

Symbol

TensOutput is an option for Tensor. Values can be "Vector" or "Matrix".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`TensOutput

SpectroTools

Functions

Symbol

ApodizeEcho[echo] performs apodization on the echo. The apodization function is set with the option ApodizationFunction.

Documentation Local »

Default Definitions SyntaxInformation[ApodizeEcho] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options ApodizationFunction → Hanning

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ApodizeEcho

Symbol

ApodizeFid[fid] performs apodization on the fid. The apodization function is set with the option ApodizationFunction.

Documentation Local »

Default Definitions SyntaxInformation[ApodizeFid] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options ApodizationFunction → Hanning

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ApodizeFid

Symbol

ApodizePadEcho[echo] performs apodization on the echo and pads the echo with zeros to increase its length.

Documentation Local »

Default Definitions SyntaxInformation[ApodizePadEcho] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {ApodizationFunction → Hanning, PaddingFactor → 2}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ApodizePadEcho

Symbol

ApodizePadFid[fid] performs apodization on the fid and pads the fid with zeros to increase its length.

Documentation Local »

Default Definitions SyntaxInformation[ApodizePadFid] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {ApodizationFunction → Hanning, PaddingFactor → 2}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ApodizePadFid

Symbol

ApodizePadSpectra[spec] and doubles the number of spectral points while maintainig the dwell time.

Documentation Local »

Default Definitions SyntaxInformation[ApodizePadSpectra] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {ApodizationFunction → Hanning, PaddingFactor → 2, ReadoutType → Fid}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ApodizePadSpectra

Symbol

ApodizeSpectra[spec] performs apodization of the spectra. The apodization function is set with the option ApodizationFunction.

Documentation Local »

Default Definitions SyntaxInformation[ApodizeSpectra] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {ApodizationFunction → Hanning, ReadoutType → Fid}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ApodizeSpectra

Symbol

ChangeDwellTimeFid[fid, dt, dtnew] changes the sampleling time of an fid from dwelltime dt to dwelltime dtnew.

Documentation Local »

Default Definitions SyntaxInformation[ChangeDwellTimeFid] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ChangeDwellTimeFid

Symbol

CompareFidFitPlot[time, fidPlot, fitPlot] function not done.

Documentation Local »

Default Definitions SyntaxInformation[CompareFidFitPlot] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`CompareFidFitPlot

Symbol

CompareSpectraFitPlot[ppmPl, specPlot, fitPlot] function not done.

Documentation Local »

Default Definitions SyntaxInformation[CompareSpectraFitPlot] = {ArgumentsPattern → {_, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`CompareSpectraFitPlot

Symbol

CorrectTEFid[fid, dw, te] corrects the fid for 1st order phase by extrapolating the missing FID samples in the TE using Henkel matrix SVD analsis.

CorrectTEFid[fid, dw, te, gyro, ppmRan] corrects the fid for 1st order phase by extrapolating

the missing FID samples in the TE using Henkel matrix SVD analsis. Only the part of the spectra in the ppmRan is used for optimization.

Documentation Local »

Default Definitions SyntaxInformation[CorrectTEFid] = {ArgumentsPattern → {_, _, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`CorrectTEFid

Symbol

CorrectTESpec[spectra, dw, te] corrects the spectra for 1st order phase by extrapolating the missing FID samples in the TE using Henkel matrix SVD analsis.

CorrectTESpec[spectra, dw, te, gyro, ppmRan] corrects the spectra for 1st order phase by extrapolating

the missing FID samples in the TE using Henkel matrix SVD analsis. Only the part of the spectra in the ppmRan is used for optimization.

Documentation Local »

Default Definitions SyntaxInformation[CorrectTESpec] = {ArgumentsPattern → {_, _, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`CorrectTESpec

Symbol

CSIInterface[] opens the CSI interface. Function not done.

CSIInterface[te, bw] opens the CSI interface with known te and bw.

CSIInterface[file] opens the CSI interface with the data from file loaded.

CSIInterface[file, {tei, bwi}] opens the CSI interface with the data from file loaded with known te and bw.

Documentation Local »

Default Definitions SyntaxInformation[CSIInterface] = {ArgumentsPattern → {_., _., OptionsPattern[]}}

Options {SpectraFieldStrength → 7, SpectraNucleus → 31P}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`CSIInterface

Symbol

ExportSparSdat[file, specs, {bw ,te}, {gyro ,nuc}] exports specs to file. Function not done.

Documentation Local »

Default Definitions SyntaxInformation[ExportSparSdat] = {ArgumentsPattern → {_, _, {_, _}, {_, _}, _., OptionsPattern[]}}

Options {SparName → QMRITools, SparOrientation → {0, 0}, SparID → }

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ExportSparSdat

Symbol

FindSpectraPpmShift[spectra, {dw, gyro}, peaks] finds the ppm value that aligns the spectra with the given peak positions peaks wich is a list of ppm values.

FindSpectraPpmShift[spectra, {dw, gyro}, {peaks, amps}] finds the ppm value

that aligns the spectra with the given peak positions peaks wich is a list of ppm values and amps are ther relative amplitudes.

FindSpectraPpmShift[spectra, {dw, gyro}, specTar] finds the ppm value that aligns the spectra with the given target spectra specTar.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`FindSpectraPpmShift

Symbol

FitSpectra[specBasis, spec, {st,end}, dt, {lwvals,lwamsp}] Fits the basis spectra from GetSpectraBasisFunctions to the spec overt the ppm range {st, end} and dt the dweltime.

Documentation Local »

Default Definitions SyntaxInformation[FitSpectra] = {ArgumentsPattern → {_, _, _, _, _, OptionsPattern[]}}

Options › SpectraNucleus → 1H … (10 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`FitSpectra

Symbol

FitSpectraResultTable [parFit, parsF, names, ref, out] function not done.

Documentation Local »

Default Definitions SyntaxInformation[FitSpectraResultTable] = {ArgumentsPattern → {_, _, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`FitSpectraResultTable

Symbol

GetGyro[nuc, field] geth the gyromagnetic ratio with field the field strength in Tesla and nuc the nucleus availible in GyromagneticRatio.

Documentation Local »

Default Definitions SyntaxInformation[GetGyro] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`GetGyro

Symbol

GetPpmRange[spec, {dt, field, nuc}] get the ppm values of the spec where dt is the well time in ms, field the field strength in Tesla and nuc the nucleus availible in GyromagneticRatio.

GetPpmRange[spec, dt, field, nuc] get the ppm values of the spec.

GetPpmRange[spec, dt, gyro] get the ppm values of the spec.

Documentation Local »

Default Definitions SyntaxInformation[GetPpmRange] = {ArgumentsPattern → {_, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`GetPpmRange

Symbol

GetSpectraBasisFunctions[{met1, ..., metn}] generates a list of spectra baisis functions with names met1 to metn. The names are strings and are the metabolites availible in GetSpinSystem.

GetSpectraBasisFunctions[{{props1}, ..., {propsn}}] generates a list of spectra baisis functions with properties prop1 to propn. The properties are those specified in MakeSpinSystem.

GetSpectraBasisFunctions[inp, split] generates a list of spectra basisfunctions. Each metabolite name present in the list split wil be split in individual spectra per peak.

Documentation Local »

Default Definitions SyntaxInformation[GetSpectraBasisFunctions] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options › BasisSequence → {PulseAcquire, 0} … (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`GetSpectraBasisFunctions

Symbol

GetTimePpmRange[spec, {dt, field, nuc}] get the timing of the fid and the ppm values

of the spec where dt is the well time in ms, field the field strength in Tesla and nuc the nucleus availible in GyromagneticRatio.

GetTimePpmRange[spec, dt, field, nuc] get the timing of the fid and the ppm values of the spec.

GetTimePpmRange[spec, dt, gyro] get the timing of the fid and the ppm values of the spec.

Documentation Local »

Default Definitions SyntaxInformation[GetTimePpmRange] = {ArgumentsPattern → {_, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`GetTimePpmRange

Symbol

GetTimeRange[fid, dt] get the timing of the fid where dt is the well time in ms.

Documentation Local »

Default Definitions SyntaxInformation[GetTimeRange] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`GetTimeRange

Symbol

ImportSparSdat[fspar, fsdat] imports sfpar and fsdata file. Function not done.

Documentation Local »

Default Definitions SyntaxInformation[ImportSparSdat] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ImportSparSdat

Symbol

MakeSpectraResultPlot[ppmF, specF, {fit, basisFit}, names, sc, met] function not done.

Documentation Local »

Default Definitions SyntaxInformation[MakeSpectraResultPlot] = {ArgumentsPattern → {_, _, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`MakeSpectraResultPlot

Symbol

PadEcho[echo] pads the echo with zeros to increse its length.

Documentation Local »

Default Definitions SyntaxInformation[PadEcho] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options PaddingFactor → 2

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PadEcho

Symbol

PadFid[fid] pads the fid with zeros to increse its length.

Documentation Local »

Default Definitions SyntaxInformation[PadFid] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options PaddingFactor → 2

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PadFid

Symbol

PadSpectra[spec] doubles the number of spectral points while maintainig the dwell time.

Documentation Local »

Default Definitions SyntaxInformation[PadSpectra] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {PaddingFactor → 2, ReadoutType → Fid}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PadSpectra

Symbol

PhaseCorrectSpectra[spec] performs 0th order phase correction of the spectra by minimizing the difference between the real and absolute spectra velaue.

PhaseCorrectSpectra[spec, dw] performs 0th order phase correction of the spectra using Henkel matrix SVD fitting.

PhaseCorrectSpectra[spec, dw, te] := performs 0th and 1st order phase correction of the spectra

using Henkel matrix SVD fitting. The first order phase is corrected by padding the fid with the missing values in the time befroe the TE.

PhaseCorrectSpectra[spec, dw, te, gyro, ppmRan] performs 0th and 1st order phase

correction of the spectra using Henkel matrix SVD fitting. Only the part of the spectra in the ppmRan is used for optimization.

Documentation Local »

Default Definitions SyntaxInformation[PhaseCorrectSpectra] = {ArgumentsPattern → {_, _., _., _., _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PhaseCorrectSpectra

Symbol

PhaseShiftSpectra[spectra, phi0] aplies the 0th order phase phi0 to the spectra.

PhaseShiftSpectra[spectra, ppm, gyro, phi1] aplies the 1st order phase phi1 to the spectra. The ppm can be obtained using GetPpmRange and gyro with GetGyro.

PhaseShiftSpectra[spec, ppm, gyro, {phi0, phi1}] aplies the 0th and 1st order phases {phi0, phi1} to the spectra. The ppm can be obtained using GetPpmRange and gyro with GetGyro.

The 0th order phase phi0 is in radians and the 1st order phase phi1 is in ms.

Documentation Local »

Default Definitions SyntaxInformation[PhaseShiftSpectra] = {ArgumentsPattern → {_, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PhaseShiftSpectra

Symbol

PlotCSIData[spectra, {dwell, gyro}] plots the CSI spectra which has dimensions {z,y,x,nsamp}. The ppm axes is determined by dwell and gyro. Gyro can be obtained with GetGyro.

PlotCSIData[spectra, {dwell, field, nuc}] plots the CSI spectra which has dimensions {z,y,x,nsamp}. The ppm axes is determined by dwell and field and nuc.

Documentation Local »

Default Definitions SyntaxInformation[PlotCSIData] = {ArgumentsPattern → {_, _, _., _., OptionsPattern[]}}

Options PlotRange → Full

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PlotCSIData

Symbol

PlotFid[fid, dwell] plots the fid assuming dwell as the sampeling time.

PlotFid[time, fid] plot the fid where time is the timing of the fid which can be obtained with GetTimeRange.

Documentation Local »

Default Definitions SyntaxInformation[PlotFid] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options › PlotRange → Full … (8 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PlotFid

Symbol

PlotSpectra[spectra, {dwell, gyro}] plots the spectra, the ppm axes is determined by dwell and gyro. Gyro can be obtained with GetGyro.

PlotSpectra[spespectradwell, field, nuc}] plots the spectra, the ppm axes is determined by dwell field and nuc.

PlotSpectra[ppm, spectra] plots the spectra where ppm is the pmm range of the spectra which can be obtained with GetPpmRange.

Documentation Local »

Default Definitions SyntaxInformation[PlotSpectra] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options › PlotRange → Full … (11 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PlotSpectra

Symbol

ReadjMRUI[file] read a jMRUI spectrum file.

Output is the {time, spec, {begintime, samplingInterval}}.

Documentation Local »

Default Definitions SyntaxInformation[ReadjMRUI] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ReadjMRUI

Symbol

ShiftSpectra[speca, {dw, gyro}, shift] shifts the spectra by shift. The shift is in ppm.

Documentation Local »

Default Definitions SyntaxInformation[ShiftSpectra] = {ArgumentsPattern → {_, _, _, OptionsPattern[]}}

Options ReadoutType → Fid

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ShiftSpectra

Symbol

SpectraFitResult[spec, {fit, basisFit}, te, {dw, gyro}, {pars, names, metRef, log}, plots, OptionsPattern[]] function not done.

Documentation Local »

Default Definitions SyntaxInformation[SpectraFitResult] = {ArgumentsPattern → {_, _, _, _, _, _, _., OptionsPattern[]}}

Options PlotRange → Full

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SpectraFitResult

Symbol

TimeShiftEcho[fid, time, gam] aplies a linebroadening with linewidth gam and a Voigt lineshape to the fid. The time can be obtained using GetTimeRange.

TTimeShiftEcho[fid, time, {gam, f}] aplies a linebroadening with linewidth gam and a custom lineshape f to the fid (f=0, "Gaussinan", f=1 "Lorentzian").

TTimeShiftEcho[fid, time, gyro, {gam, eps}] aplies a linebroadening with linewidth gam to the fid and a phase eps that results in eps ppm shift of the spectra. The gyro can be obtained with GetGyro.

TTimeShiftEcho[fid, time, gyro, {gam, eps, f}] aplies a linebroadening with linewidth gam using a custom lineshape f to the fid and a phase eps that results in eps ppm shift of the spectra.

The linewidth gam is given in ms and the spectra shift eps is given in ppm.

Documentation Local »

Default Definitions SyntaxInformation[TimeShiftEcho] = {ArgumentsPattern → {_, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`TimeShiftEcho

Symbol

TimeShiftFid[fid, time, gam] aplies a linebroadening with linewidth gam and a Voigt lineshape to the fid. The time can be obtained using GetTimeRange.

TimeShiftFid[fid, time, {gam, f}] aplies a linebroadening with linewidth gam and a custom lineshape f to the fid (f=0, "Gaussian", f=1 "Lorentzian").

TimeShiftFid[fid, time, gyro, {gam, eps}] aplies a linebroadening with linewidth gam to the fid and a phase eps that results in eps ppm shift of the spectra. The gyro can be obtained with GetGyro.

TimeShiftFid[fid, time, gyro, {gam, eps, f}] aplies a linebroadening with linewidth gam using a custom lineshape f to the fid and a phase eps that results in eps ppm shift of the spectra.

The linewidth gam is given in ms and the spectra shift eps is given in ppm.

Documentation Local »

Default Definitions SyntaxInformation[TimeShiftFid] = {ArgumentsPattern → {_, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`TimeShiftFid

Symbol

TimeShiftFid[fid, time, gam] aplies a linebroadening with linewidth gam and a Voigt lineshape to the fid. The time can be obtained using GetTimeRange.

TimeShiftFid[fid, time, {gam, f}] aplies a linebroadening with linewidth gam and a custom lineshape f to the fid (f=0, "Gaussian", f=1 "Lorentzian").

TimeShiftFid[fid, time, gyro, {gam, eps}] aplies a linebroadening with linewidth gam to the fid and a phase eps that results in eps ppm shift of the spectra. The gyro can be obtained with GetGyro.

TimeShiftFid[fid, time, gyro, {gam, eps, f}] aplies a linebroadening with linewidth gam using a custom lineshape f to the fid and a phase eps that results in eps ppm shift of the spectra.

The linewidth gam is given in ms and the spectra shift eps is given in ppm.

Documentation Local »

Default Definitions SyntaxInformation[TimeShiftFidV] = {ArgumentsPattern → {_, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`TimeShiftFidV

Options

Symbol

ApodizationFunction is an options for ApodizeFid, ApodizeSpectra, ApodizePadFid, and ApodizePadSpectra. Values can be "Hanning", "Hamming", "Gaussian", "Lorentzian", and "Voigt".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ApodizationFunction

Symbol

AspectRatio is an option for Graphics and related functions that specifies the ratio of height to width for a plot.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`AspectRatio

Symbol

BasisSequence is an option for GetSpectraBasisFunctions and specifies which sequence to use.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`BasisSequence

Symbol

CenterFrequency is an option for GetSpinSystem and defines the center frequency in ppm.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`CenterFrequency

Symbol

FineTuneFit is an option for FitSpectra and when True it performs a second fitting run where for each peak is an individual linewidth, lineshape and shift are fitted.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`FineTuneFit

Symbol

FitLineShape is an option for FitSpectra and when True allows to fit the lineshap. If False a voigt lineshape is used.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`FitLineShape

Symbol

GridLines is an option for two–dimensional graphics functions that specifies grid lines.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`GridLines

Symbol

GridLineSpacing is an option of TransmuralPlot. It defines the spacing of the gridlines.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`GridLineSpacing

Symbol

ImageSize is an option that specifies the overall size of an image to display for an object.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ImageSize

Symbol

InitializeFit is an option for FitSpectra and is used to set initila values for the global fit {gami,epsi,{phi0i,phi1i},lineshape}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`InitializeFit

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

PaddingFactor is an option for PadFid, PadSpectra, ApodizePadFid, ApodizePadSpectra and FitSpectra. It Specifies with which factro to lengthen the fid.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PaddingFactor

Symbol

PlotColor is an option for GradientPlot can be any color or gradient color name.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotColor

Symbol

PlotLabel is an option for graphics functions that specifies an overall label for a plot.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`PlotLabel

Symbol

PlotLabels is an option for visualization functions that specifies what labels to use for each data source.

Documentation Local » | Web »

Attributes {Protected, ReadProtected}

Full Name System`PlotLabels

Symbol

PlotRange is an option for graphics functions that specifies what range of coordinates to include in a plot.

Documentation Local » | Web »

Attributes {Protected, ReadProtected}

Full Name System`PlotRange

Symbol

ReadoutType is an option for FitSpectra and padding and apodization fucntions. Value can be "Fid" or "Echo".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ReadoutType

Symbol

SparID is an option for ExportSparSdat.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SparID

Symbol

SparName is an option for ExportSparSdat.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SparName

Symbol

SparOrientation is an option for ExportSparSdat.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SparOrientation

Symbol

SpectraBandwith is an option for GetSpectraBasisFunctions and sets the bandwith of the spectra.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SpectraBandwith

Symbol

SpectraFieldStrength is an option for GetSpectraBasisFunctions and FitSpectra and sets the field strenght at which the simulations and fitting is perforemd.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SpectraFieldStrength

Symbol

SpectraNucleus is an option for GetSpectraBasisFunctions and FitSpectra and specifies which nucleus to Simulate or fit, see GyromagneticRatio.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SpectraNucleus

Symbol

SpectraOutputPlots is an option for FitSpectra. If True the automatica calibration plot for the initial fit are generated.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SpectraOutputPlots

Symbol

SpectraPpmShift is an option for GetSpectraBasisFunctions and FitSpectra and defines how much the center frequency is shifted, default is water at 4.65 ppm.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SpectraPpmShift

Symbol

SpectraSamples is an option for GetSpectraBasisFunctions and sets the number of samples in the spectra.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SpectraSamples

Symbol

SpectraSpacing is an option for PlotSpectra and defines the amount of spacing between spectra when multiple spectra are plotted.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SpectraSpacing

Symbol

SplineSpacingFactor is an option for FitSpectra and defines the distance between the bsplien points relative the the mean linewithd of the peaks.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SplineSpacingFactor

TaggingTools

Functions

Symbol

AnnalyzeTagging[gridC].

Documentation Local »

Options {HistoryWeighting → 0.7, MonitorTagging → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TaggingTools`AnnalyzeTagging

Symbol

CalculateDispacementParameters[{motx, moty}, mask].

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TaggingTools`CalculateDispacementParameters

Options

Symbol

HistoryWeighting is an options for AnnalyzeTagging.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TaggingTools`HistoryWeighting

Symbol

MonitorTagging is an options for AnnalyzeTagging.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TaggingTools`MonitorTagging

TensorTools

Functions

Symbol

ADCCalc[eigenvalues] caculates the ADC from the given eigenvalues.

Documentation Local »

Default Definitions SyntaxInformation[ADCCalc] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`ADCCalc

Symbol

AngleCalc[data, vector] calculates the angel between the vector and the data. Data shoud be an array of dimensions {xxx,3}.

Documentation Local »

Default Definitions SyntaxInformation[AngleCalc] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options Distribution → 0–180

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`AngleCalc

Symbol

AngleMap[data] calculates the zennith and azimuth angles of a 3D dataset (z,x,y,3) containing vectors relative to the slice direction.

Documentation Local »

Default Definitions SyntaxInformation[AngleMap] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`AngleMap

Symbol

ColorFAPlot[tenor] create a color coded FA map from the tensor for l1, l2 and l3.

Documentation Local »

Default Definitions SyntaxInformation[ColorFAPlot] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`ColorFAPlot

Symbol

ConcatenateDiffusionData[{{data1, .., dataN}, {grad1, .., gradN}, {bval, .., bvalN}, {vox, .., voxN}}] concatenates the diffusion data sets.

ConcatenateDiffusionData[{data1, .., dataN}, {grad1, .., gradN}, {bval, .., bvalN}, {vox, .., voxN}] concatenates the diffusion data sets.

Documentation Local »

Default Definitions SyntaxInformation[ConcatenateDiffusionData] = {ArgumentsPattern → {_, _., _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`ConcatenateDiffusionData

Symbol

Correct[data, phase, shiftpar] corrects the dataset data using the phasemap and the shiftpar and interpolation order 1.

Correct[data, phase, shiftpar, int] corrects the dataset data using the phasemap and the shiftpar and interpolation order int.

Documentation Local »

Default Definitions SyntaxInformation[Correct] = {ArgumentsPattern → {_, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`Correct

Symbol

Deriv[disp, vox] calculates the derivative of the displacement along the three main axes. disp is the displacement field, vox is the voxel size.

Deriv[disp, vox, mask] calculates the derivative of the displacement along the three main

axes. Sharp edges between the background en disp are solved by the mask. mask is a mask delining the edge of the displacement field.

Documentation Local »

Default Definitions SyntaxInformation[Deriv] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`Deriv

Symbol

DriftCorrect[data, bval] dirft corrects the data using the signals of the lowest bvalue that has 6 or more unique volumes.

For the function to work optimal it is best to have these volumes evenly spread througout thet data and for the first and last volume to have this low bvalue.

DriftCorrect[] is based on DOI: 10.1002/mrm.26124.

Documentation Local »

Default Definitions SyntaxInformation[DriftCorrect] = {ArgumentsPattern → {_, _, _., OptionsPattern[]}}

Options {NormalizeSignal → True, UseMask → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`DriftCorrect

Symbol

ECalc[eigenvalues] caculates the E from the given eigenvalues.

Documentation Local »

Default Definitions SyntaxInformation[ECalc] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options MonitorCalc → True

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`ECalc

Symbol

EigensysCalc[tensor] caculates the eigensystem for the given tensor.

Documentation Local »

Default Definitions SyntaxInformation[EigensysCalc] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {RejectMap → False, Reject → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`EigensysCalc

Symbol

EigenvalCalc[tensor] caculates the eigenvalues for the given tensor.

Documentation Local »

Default Definitions SyntaxInformation[EigenvalCalc] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {RejectMap → False, Reject → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`EigenvalCalc

Symbol

EigenvecCalc[tensor] caculates the eigenvectors for the given tensor.

Documentation Local »

Default Definitions SyntaxInformation[EigenvecCalc] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options {RejectMap → False, Reject → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`EigenvecCalc

Symbol

FACalc[eigenvalues] caculates the FA from the given eigenvalues.

Documentation Local »

Default Definitions SyntaxInformation[FACalc] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`FACalc

Symbol

FlipGradientOrientation[grad, perm] permutes the internal orientation of the gradients, perm can be any permutation of {"x","y","z"}.

FlipGradientOrientation[grad, flip] flips the internal orientation of the gradients, flip can be {1,1,1}, {−1,1,1}, {1,−1,1} or {1,1,−1}.

FlipGradientOrientation[grad, flip, perm] flips and permuter the internal orientation of the gradients.

FlipGradientOrientation[grad, perm, flip]flips and permuter the internal orientation of the gradients.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`FlipGradientOrientation

Symbol

FlipTensorOrientation[tens, perm] permutes the internal orientation of the tensor, perm can be any permutation of {"x","y","z"}.

FlipTensorOrientation[tens, flip] flips the internal orientation of the tensor, flip can be {1,1,1}, {−1,1,1}, {1,−1,1} or {1,1,−1}.

FlipTensorOrientation[tens, flip, perm] flips and permuter the internal orientation of the tensor.

FlipTensorOrientation[tens, perm, flip]flips and permuter the internal orientation of the tensor.

Documentation Local »

Default Definitions SyntaxInformation[FlipTensorOrientation] = {ArgumentsPattern → {_, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`FlipTensorOrientation

Symbol

ParameterCalc[tensor] caculates the eigenvalues and MD and FA from the given tensor. The parameters are l1, l2, l3, MD and FA. l1, l2, l3, MD are in (10^−3 mm^2/s).

Documentation Local »

Default Definitions SyntaxInformation[ParameterCalc] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options Reject → False

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`ParameterCalc

Symbol

RemoveIsoImages[data, grad, bval] Romoves the ISO images from the philips scanner from the data. ISO images have g={0,0,0} and b>0.

Documentation Local »

Default Definitions SyntaxInformation[RemoveIsoImages] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`RemoveIsoImages

Symbol

ResidualCalc[DTI,{tensor,S0},gradients,bvector] calculates the tensor residuals for the given dataset.

ResidualCalc[DTI,{tensor,S0},outlier,gradients,bvector] calculates the tensor residuals for the given dataset taking in account the outliers.

ResidualCalc[DTI,{tensor,S0},bmat] calculates the tensor residuals for the given dataset.

ResidualCalc[DTI,{tensor,S0},outlier,bmat] calculates the tensor residuals for the given dataset taking in account the outliers.

ResidualCalc[DTI,tensor,gradients,bvector] calculates the tensor residuals for the given dataset. Tensor must contain Log[S0].

ResidualCalc[DTI,tensor,outlier,gradients,bvector] calculates the tensor residuals for the given dataset taking in account the outliers. Tensor must contain Log[S0].

ResidualCalc[DTI,tensor,bmat] calculates the tensor residuals for the given dataset. Tensor must contain Log[S0].

ResidualCalc[DTI,tensor,outlier,bmat] calculates the tensor residuals for the given dataset taking in account the outliers. Tensor must contain Log[S0].

Documentation Local »

Default Definitions SyntaxInformation[ResidualCalc] = {ArgumentsPattern → {_, _, _, _, OptionsPattern[]}}

Options MeanRes → All

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`ResidualCalc

Symbol

SigmaCalc[DTI,grad,bvec] calculates the noise sigma based on the tensor residual, using a blur factor of 10.

SigmaCalc[DTI,tens,grad,bvec] calculates the noise sigma based on the tensor residual, using a blur factor of 10.

SigmaCalc[DTI,grad,bvec,blur] calculates the noise sigma based on the tensor residual, If blur is 1 ther is no blurring.

SigmaCalc[DTI,tens,grad,bvec,blur] calculates the noise sigma based on the tensor residual. If blur is 1 ther is no blurring.

Documentation Local »

Default Definitions SyntaxInformation[SigmaCalc] = {ArgumentsPattern → {_, _, _, _., _., OptionsPattern[]}}

Options FilterShape → Median

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`SigmaCalc

Symbol

SortDiffusionData[data, grad, bval] sorts the diffusion datasets grad and bval for magnitude of bvalue.

Documentation Local »

Default Definitions SyntaxInformation[SortDiffusionData] = {ArgumentsPattern → {_, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`SortDiffusionData

Symbol

TensorCalc[data, gradients, bvalue] calculates the diffusion tensor for the given dataset. Allows for one unweighted image and one b value.

Gradient directions must be in the form {{x1,y1,z1}, ..., {xn,yn,zn}} without the unweighted gradient direction.

bvalue is a singe number indicating the b−value used.

TensorCalc[data, gradients, bvec] calculates the diffusion tensor for the given dataset. allows for multiple unweighted images and multiple bvalues.

allows for differnt tensor fitting methods. gradient directions must be in the form {{x1,y1,z1}, ..., {xn,yn,zn}} with the unweighted direction as {0,0,0}.

bvec the bvector, with a bvalue defined for each gradient direction. b value for unweighted images is 0.

TensorCalc[data, bmatix] calculates the diffusion tensor for the given dataset. allows for multiple unweighted images and multiple bvalues.

bmat is the bmatrix which can be generated usiong Bmatrix.

The bvalue assumed to be is in s/mm^2 and therfore the output is in mm^2/2.

TensorCalc[] is based on DOI: 10.1016/j.neuroimage.2013.05.028 and 10.1002/mrm.25165.

Documentation Local »

Default Definitions SyntaxInformation[TensorCalc] = {ArgumentsPattern → {_, _, _., OptionsPattern[]}}

Options › MonitorCalc → True ... (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`TensorCalc

Symbol

TensorCorrect[tensor, phase, shift, vox] corrects the tensor based on B0 field map. Can perform both translation and rotation of tensor.

Documentation Local »

Default Definitions SyntaxInformation[TensorCorrect] = {ArgumentsPattern → {_, _, _, _, _., OptionsPattern[]}}

Options RotationCorrect → False

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`TensorCorrect

Symbol

WestinMeasures[eigenvalues] calculates the westin measures.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`WestinMeasures

Options

Symbol

Distribution is an option for AngleCalc. values can be "0–180", "0–90" and "–90–90".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`Distribution

Symbol

FilterShape is an option for SigmaCalc. Can be "Gaussian" of "Median".

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`FilterShape

Symbol

FullOutput is an option for TensorCalc when using bvector. When True also the S0 is given as output.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`FullOutput

Symbol

MeanRes is an option for ResidualCalc. When True the root mean square of the residual is calculated.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`MeanRes

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

MonitorCalc is an option for all Calc fucntions. When true the proceses of the calculation is shown.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`MonitorCalc

Symbol

NormalizeSignal is an option for DriftCorrect.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`NormalizeSignal

Symbol

Parallelize[*expr*] evaluates *expr* using automatic parallelization.

Definitions

```
Parallelize[Parallel`Evaluate`Private`expr_, Parallel`Evaluate`Private`opts : OptionsPattern[]] :=
 Module[{Parallel`Evaluate`Private`res, Parallel`Evaluate`Private`ehead, Parallel`Evaluate`Private`fopts}, Parallel`Protected`tryRelaunch[];
  If[Parallel`Protected`$seQ, Message[Parallelize::nopar];
   Return[Parallel`Evaluate`Private`expr]];
  Parallel`Evaluate`Private`fopts = Sequence @@ {Method → OptionValue[Method],
     DistributedContexts → Parallel`Protected`DistOptCheck[Parallelize, OptionValue[DistributedContexts]], ProgressReporting → OptionValue[ProgressReporting]};
  Parallel`Evaluate`Private`ehead = Replace[Head[Unevaluated[Parallel`Evaluate`Private`expr], Hold],
    {Hold[Parallel`Evaluate`Private`s_Symbol] :→ Parallel`Evaluate`Private`s, _ :→ $Failed}];
  With[{Parallel`Evaluate`Private`handler = Lookup[Parallel`Evaluate`$ParallelExtensions, Parallel`Evaluate`Private`ehead, $Failed]},
   If[Parallel`Evaluate`Private`handler =!= $Failed, Return[Parallel`Evaluate`Private`handler[Parallel`Evaluate`Private`expr, Parallel`Evaluate`Private`fopts]]]];
  Parallel`Evaluate`Private`res = Parallel`Evaluate`Private`tryCombine[Parallel`Evaluate`Private`expr, Parallel`Evaluate`Private`fopts];
  If[Head[Parallel`Evaluate`Private`res] =!= Parallel`Evaluate`Private`tryCombine, Return[Parallel`Evaluate`Private`res]];
  Block[{Parallel`Evaluate`Private`$seqWarning = False},
   Parallel`Evaluate`Private`res = Parallel`Evaluate`Private`wrapAround[Parallel`Evaluate`Private`expr, Parallel`Evaluate`Private`fopts];
   If[Head[Parallel`Evaluate`Private`res] =!= Parallel`Evaluate`Private`wrapAround, Return[Parallel`Evaluate`Private`res]];];
  Parallel`Evaluate`Private`res = Parallel`Evaluate`Private`silentFail[Parallel`Evaluate`Private`expr];
  If[Head[Parallel`Evaluate`Private`res] =!= Parallel`Evaluate`Private`silentFail, Return[Parallel`Evaluate`Private`res]];
  If[Parallel`Evaluate`Private`$seqWarning, Message[Parallelize::nopar1, Parallel`Evaluate`Private`expr]];
  Parallel`Evaluate`Private`expr]
```

Documentation Local » | Web »

Options {DistributedContexts :→ $Context, Method → Automatic, ProgressReporting :→ $ProgressReporting}

Attributes {HoldFirst, Protected}

Full Name System`Parallelize

Symbol

Reject is an option for EigenvalCalc. It True then voxels with negative eigenvalues are rejected and set to 0.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`Reject

Symbol

RejectMap is an option for EigenvalCalc. If Reject is True and RejectMap is True both the eigenvalues aswel as a map showing je rejected values is returned.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`RejectMap

Symbol

RobustFit is an option for TensorCalc. If true outliers will be rejected in the fit, only works with WLLS.

If FullOutput is given the outlier map is given.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`RobustFit

Symbol

RobustFitParameters is an option for TensorCalc. gives the threshold for stopping the itterations and the kappa for the outlier marging, {tr,kappa}.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`RobustFitParameters

Symbol

RotationCorrect is an option for TensorCorrect. Default is False. Is a tensor is deformed setting to True also the shear is accounted for by local rotation of the tensor.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`RotationCorrect

Symbol

UseMask is a function for MeanSignal and DriftCorrect.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`UseMask

TractographyTools

Functions

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`CombineROIs

Symbol

FiberTractography[tensor, vox] performs fibertractography on the tensor with voxels dimensions vox.

FiberTractography[tensor, vox, {par, {min, max}}] performs fibertractography on the tensor with voxels dimensions vox with additional stoppin criteria par, where tracts are only generated between values of par min and max.

FiberTractography[tensor, vox, {{par, {min, max}}, ..}] performs fibertractography on the tensor with voxels dimensions vox with multiple additional stopping criteria.

Documentation Local »

Default Definitions SyntaxInformation[FiberTractography] = {ArgumentsPattern → {_, _, _., OptionsPattern[]}}

Options › FiberLengthRange → {10, 200} … (10 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FiberTractography

Symbol

FilterTracts[tracts_, vox_, select_]

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FilterTracts

Symbol

FindTensorPermutation[tensor, vox] performs tractography for all tensor permutations and gives back the one that has the longest tracts.

FindTensorPermutation[tensor, vox, {par, {min, max}}] same but with additional stoppin criteria par, where tracts are only generated between values of par min and max.

FindTensorPermutation[tensor, vox, {{par, {min, max}}, ..}] same but with with multiple additional stopping criteria.

Ouput = {permutations, flips, plot}

FindTensorPermutation[] is based on DOI: 10.1016/j.media.2014.05.012.

Documentation Local »

Default Definitions SyntaxInformation[FindTensorPermutation] = {ArgumentsPattern → {_, _, _., OptionsPattern[]}}

Options › FiberLengthRange → {10, 200} … (7 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FindTensorPermutation

Symbol

FitTract[tract] fits a tract defined as a list of {x,y,z} coordinates with a polinomial function.

FitTract[{tract, ...}] fits a list of tracts defined as a list of {x,y,z} coordinates with a polinomial function.

Documentation Local »

Default Definitions SyntaxInformation[FitTract] = {ArgumentsPattern → {_, OptionsPattern[]}}

Options FittingOrder → 4

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FitTract

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`MakeColor

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`PartTracts

Symbol

Documentation Local »

Options {MaxTracts → 5000, ImageSize → 800}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`PlotTracts

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`SeedDensityMap

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`SelectTractInVol

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`SelectTractPartInVol

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`SelectTracts

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`SelectTractTroughPlane

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`SelectTractTroughVol

Symbol

TractDensityMap[tracts_, vox_, dim_]

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`TractDensityMap

Options

Symbol

FiberAngle is an option for FiberTractography and specifies the allowed angle change per tract step.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FiberAngle

Symbol

FiberLengthRange is an option for FiberTractography and specifies the allowed tract range.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FiberLengthRange

Symbol

FittingOrder is an option for FitTract. It specifies the polinominal order of the function to fit the tract.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FittingOrder

Symbol

ImageSize is an option that specifies the overall size of an image to display for an object.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`ImageSize

Symbol

InterpolationOrder is an option for Interpolation, as well as ListLinePlot, ListPlot3D, ListContourPlot, and related functions, that specifies what order of interpolation to use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`InterpolationOrder

Symbol

MaxSeedPoints is an option for FiberTractography and defines the maximum number of seedspoints to be used.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`MaxSeedPoints

Symbol

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`MaxTracts

Symbol

Method is an option for various algorithm–intensive functions that specifies what internal methods they should use.

Documentation Local » | Web »

Attributes {Protected}

Full Name System`Method

Symbol

StepSize is an option for FiberTractography and defines the tractography step size.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`StepSize

Symbol

StopThreshhold is an option for FiberTractography and defines the stop threshhold which is a value between 0 and 1.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`StopThreshhold

Symbol

TensorFilps is an option for FiberTractography and speciefies if the tensor orientation is fliped, see FlipTensorOrientation.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`TensorFilps

Symbol

TensorPermutations is an option for FiberTractography and speciefies if the tensor orientation is permuted, see FlipTensorOrientation.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`TensorPermutations

Symbol

TracMonitor is an option for FiberTractography. When set True it prints the progress.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`TracMonitor

ViisteTools

Functions

Symbol

DatRead[file] imports data from file (dtitool *.dat format) as binary data using Real32 format.

Documentation Local »

Default Definitions SyntaxInformation[DatRead] = {ArgumentsPattern → {_}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`DatRead

Symbol

DatWrite[file, data] exports data to *.dat file as binary data using Real32 format.

Documentation Local »

Default Definitions SyntaxInformation[DatWrite] = {ArgumentsPattern → {_, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`DatWrite

Symbol

DTItoolExp[tensor, voxsize] exports tensor to {XX.dat, YY.dat, ZZ.dat, XY.dat, XZ.dat, YZ.dat} and uses XX.dat as background and generates corresponding *dti files.

DTItoolExp[tensor, voxsize, folder] exports tensor to {XX.dat, YY.dat, ZZ.dat, XY.dat, XZ.dat, YZ.dat} to the given folder and uses XX.dat as background and generates corresponding *dti files.

DTItoolExp[tensor, voxsize, folder, add] exports tensor to {XX.dat, YY.dat, ZZ.dat, XY.dat, XZ.dat,

YZ.dat} to the given folder and uses XX.dat as background and generates corresponding *dti files adds – add to the filenames.

DTItoolExp[back, tensor, voxsize] exports background to back.dat and tensor to {XX.dat, YY.dat, ZZ.dat, XY.dat, XZ.dat, YZ.dat} and generates corresponding *dti files.

DTItoolExp[back, tensor, voxsize, folder] exports background to back.dat and tensor to {XX.dat, YY.dat, ZZ.dat, XY.dat, XZ.dat, YZ.dat} to the given folder and generates corresponding *dti files.

DTItoolExp[back, tensor, voxsize, folder, add] exports background to back.dat and tensor to

{XX.dat, YY.dat, ZZ.dat, XY.dat, XZ.dat, YZ.dat} to the given folder and generates corresponding *dti files and adds – add to the filenames.

Documentation Local »

Default Definitions SyntaxInformation[DTItoolExp] = {ArgumentsPattern → {_, _, _, _, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`DTItoolExp

Symbol

DTItoolExpFile[file, background, add, voxsize] exports a *.dti text file.

Documentation Local »

Default Definitions SyntaxInformation[DTItoolExpFile] = {ArgumentsPattern → {_, _, _, _}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`DTItoolExpFile

Symbol

DTItoolExpInd[data, file] exports a 3D array data to the file filename DTItool format (*.dat) using DatWrite.

DTItoolExpInd[data, file ,folder] exports data to given file and folder.

DTItoolExpInd[data, file ,folder, add] exports data to given file and folder and adds –add to the filename.

Documentation Local »

Default Definitions SyntaxInformation[DTItoolExpInd] = {ArgumentsPattern → {_, _, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`DTItoolExpInd

Symbol

DTItoolExpTens[tensor] exports a diffustion tensor array to the DTItool format (*.dat).

DTItoolExpTens[tensor, add] exports tensor and adds – add to the filenames.

DTItoolExpTens[tensor, add, folder] exports tensor to the given folder and adds – add to the filenames.

Documentation Local »

Default Definitions SyntaxInformation[DTItoolExpTens] = {ArgumentsPattern → {_, _., _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`DTItoolExpTens

Symbol

ExportVol[filename, data, voxsize] exports a .vol and .raw file which can be loaded in DTItool 3.0.

Documentation Local »

Default Definitions SyntaxInformation[ExportVol] = {ArgumentsPattern → {_, _, OptionsPattern[]}}

Options BinaryType → Integer16

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`ExportVol

Symbol

ImportDTI[folder] imports xx.dat, yy.dat, zz.dat, xy.dat, xz.dat and yz.dat from the given folder.

ImportDTI[folder, add] imports xx–add.dat, yy–add.dat, zz–add.dat, xy–add.dat, xz–add.dat and yz–add.dat from the given folder.

ImportDTI[{file1, file2, ..}] imports the given *.dat files.

Documentation Local »

Default Definitions SyntaxInformation[ImportDTI] = {ArgumentsPattern → {_, _.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`ImportDTI

Symbol

ImportVol[] promts for a vol file to open.

ImportVol["file"] inpormts the file.

the function returns data and voxsize.

Documentation Local »

Default Definitions SyntaxInformation[ImportVol] = {ArgumentsPattern → {_., OptionsPattern[]}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`ImportVol

Symbol

LoadFiberTracts[] promts for a .fbs to open.

LoadFiberTracts["file"] imports the file.

Documentation Local »

Default Definitions SyntaxInformation[LoadFiberTracts] = {ArgumentsPattern → {_.}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`LoadFiberTracts

Options

Symbol

BinaryType is an option for ExportVol and must be "Integer16" for an integer array and "Real32" for a Double array.

Documentation Local »

Attributes {Protected, ReadProtected}

Full Name QMRITools`ViisteTools`BinaryType