

```
In[*]:= QMRITools`$Verbose = False;
<< QMRIToolsDev`

In[*]:= Grid[Transpose@{{"Packages", "Functions", "Options"}, Length@Flatten[QMRIToolsFunctions[]][All, #]] & /@ {1, 2, 3}}, Alignment → Left]
Out[*]=
Packages    25
Functions   491
Options     435

In[*]:= Column@QMRIToolsPackages[]
Out[*]=
CardiacTools
CoilTools
DenoiseTools
DixonTools
ElastixTools
FasciculationTools
GeneralTools
GradientTools
IVIMTools
JcouplingTools
LoggingTools
MaskingTools
MuscleBidsTools
NiftiTools
PlottingTools
ProcessingTools
ReconstructionTools
RelaxometryTools
ScientificColorData
SegmentationTools
SimulationTools
SpectroTools
TaggingTools
TensorTools
TractographyTools

In[*]:= QMRIToolsFunctions["All", 4]
```

QMRITools

Functions

ADCCalc	ExportBval	Link3DGraphic	RegisterCardiacData
AddLossLayer	ExportBvec	ListSpherePlot	RegisterData
AddNoise	ExportDirectoryTree	LLeastSquares	RegisterDataSplit
AddScientificColors	ExportLog	LoadCoilSetup	RegisterDataTransform
AddToJson	ExportNii	LoadCoilTarget	RegisterDataTransformSplit
AddToLog	ExportSparSdat	LogNoZero	RegisterDiffusionData
AnalyseNetworkFeatures	ExportTracts	LogTensor	RegisterDiffusionDataSplit
AnalyzeActivations	ExpTensor	MADNoZero	RegisterTensorData
AngleCalc	ExtractColorData	MakeChannelClassGrid	RemoveIsoImages
AngleMap	ExtractDemoData	MakeChannelClassImage	RemoveMaskOverlaps
AnisoFilterData	ExtractFromJSON	MakeChannelImage	ReplaceSegmentations
AnisoFilterTensor	ExtractNiiFiles	MakeCheckFile	ResampleTracts
AnnalyzeTagging	FACalc	MakeClassifyImage	RescaleData
ApodizeEcho	FConvert	MakeClassifyNetwork	RescaleSegmentation
ApodizeFid	FConverti	MakeClassImage	RescaleTracts
ApodizePadEcho	FiberLength	MakeCoilLayout	ResetLog
ApodizePadFid	FiberTractography	MakeCoordinates	ResidualCalc
ApodizePadSpectra	FileSelect	MakeDistanceMap	ReverseCrop
ApodizeSpectra	FilterTracts	MakeECVBloodMask	ReverseDimensions
ApplyCrop	FinalGrads	MakeFunctionGraph	RMSNoZero
ApplySegmentationNetwork	FindActivations	MakeHammingFilter	RotateData
AugmentImageData	FindCoilPosition	MakeIntFunction	RotateDimensionsLeft
AugmentTrainingData	FindCrop	MakeLineImage	RotateDimensionsRight
AutoCropData	FindInPhaseEchos	MakeMaskImage	RotateTensor
B1MapCalc	FindMaxDimensions	MakeNiiOrentationQ	RotationMatrixToQuaternion
B1Shimming	FindOrder	MakeNiiOrentationS	RotationMatrixToQuaternionVector
BayesianIVIMFit2	FindOutliers	MakeNode	SagitalTranspose
BayesianIVIMFit3	FindSpectraPpmShift	MakeNoisePlots	SaveImage
BidsDcmToNii	FindTensorPermutation	MakeSense	SeedDensityMap
BlochSeries	FitData	MakeSliceImages	SegmentationVolume
Bmatrix	FitGradientMap	MakeSpectraResultPlot	SegmentData
BmatrixCalc	FitSpectra	MakeSpinSystem	SegmentDataGUI
BmatrixConv	FitSpectraResultTable	MakeUnet	SegmentLinesToMask
BmatrixInv	FitTracts	MakeWeightMask	SegmentMask
BmatrixRot	FixDixonFlips	Mask	SegmentsPerSlice
BmatrixToggle	FlipGradientOrientation	MaskData	SegmentTracts
BSplineCurveFit	FlipTensorOrientation	MaskHelix	SelectActivations
BullseyePlot	FocalLossLayer	MaskSegmentation	SelectBidsFolders
CalculateDisplacementParameters	FourierKspace2D	MaskToLines	SelectBidsSessions
CalculateGfactor	FourierKspace3D	MBCount	SelectBidsSubjects
CalculateMoments	FourierKspaceCSI	MeanBvalueSignal	SelectBvalueData
CalculateWallMap	FourierRescaleData	MeanNoZero	SelectMaskComponents
CalibrateEPGT2Fit	FourierShift	MeanRange	SelectSegmentations
CardiacCoordinateSystem	FourierShifted	MeanSignal	SequencePulseAcquire
CardiacSegment	FracCorrect	MeanStd	SequenceSpaceEcho
CardiacSegmentGUI	FullGrad	MeanType	SequenceSpinEcho
CentralAxes	GenerateAmps	MedCouple	SequenceSteam
ChangeDwellTimeFid	GenerateBidsFileName	MedFilter	SequenceTSE
ChangeNetDimensions	GenerateBidsFolderName	MedianNoZero	ShiftedFourier
CheckDataDiscription	GenerateBidsName	MemoryUsage	ShiftedInverseFourier
CheckFile	GenerateGradients	MergeJSON	ShiftPulseProfile
CheckSegmentation	GenerateGradientsGUI	MergeSegmentations	ShiftSpectra
ClassDecoder	GenerateRotationFrames	MonitorCalc	ShowLog

Options

4	All-Functions.np.			
	AcquisitionMethod	DixonMaskThreshold	MaxSeedPoints	RejectMap
	ActivationBackground	DixonNucleus	MaxTracts	ReportFits
	ActivationIterations	DixonPhases	MaxTrainingRounds	RescaleMethod
	ActivationOutput	DixonPrecessions	MeanMethod	RescaleRecon
	ActivationSize	DixonRelaxivity	MeanRes	Resolutions
	ActivationThreshold	DixonTollerance	Method	ResolutionsA
	ActivationType	DownsampleSchedule	MethodReg	ReverseData
	AffineDirections	DropoutRate	MethodRegA	ReverseDirection
	AllowSelfDependencies	DropSamples	Monitor	ReversePoints
	AnisoFilterSteps	DropSlices	MonitorCalc	ReverseSets
	AnisoItterations	EchoShiftData	MonitorInterval	RobustFit
	AnisoKappa	EPGCalibrate	MonitorIVIMCalc	RobustFitParameters
	AnisoKernel	EPGFatShift	MonitorTagging	RotateGradients
	AnisoStepTime	EPGFitFat	MonitorUnwrap	RotationCorrect
	AnisoWeightType	EPGFitPoints	MotionCorrectSets	RoundLength
	ApodizationFunction	EPGMethod	NetworkArchitecture	RowSize
	AspectRatio	EPGMethodCal	NetworkDepth	Runs
	AugmentData	EPGRelaxPars	NiiDataType	Scaling
	AxesLabel	EPGSmoothB1	NiiLegacy	SegmentationMethod
	AxesMethod	FatFieldStrength	NiiMethod	SelectSubjects
	B1EqualPower	FeatureSchedule	NiiOffset	SenseRescale
	B1FilterData	FiberAngle	NiiScaling	SettingSchedule
	B1Masking	FiberLengthRange	NiiSliceCode	ShowMetric
	B1MaxPower	FieldStrength	NNThreshold	ShowOutliers
	B1Output	FileType	NoiseSize	ShowPlot
	B1Scaling	Filling	NoiseType	SimNucleus
	B1ShimMethod	FilterMaps	NormalizeDensity	SliceRange
	BackgroundValue	FilterShape	NormalizeIVIM	SliceRangeSamples
	BasisSequence	FilterSize	NormalizeMethod	SmartMaskOutput
	BatchSize	FilterType	NormalizeOutputSpectra	SmartMethod
	BidsIncludeSession	FindTransform	NormalizeOverlap	SmoothHelix
	BidsTractographyMethod	FineTuneFit	NormalizeSets	SmoothIterations
	BlockType	FitConstrains	NormalizeSignal	SmoothSNR
	BloodMaskRange	FitFunction	NumberSamples	SortVecs
	Boxed	FitLineShape	NumberSamplesA	SparID
	BsplineDirections	FitOrder	OrderSpan	SparName
	BsplineSpacing	FitOutput	OutlierIncludeZero	SparOrientation
	BullPlotMethod	FittingOrder	OutlierIterations	SpectraBandwith
	CenterFrequency	FitTractSegments	OutlierMethod	SpectraFieldStrength
	CenterRange	FixPseudoDiff	OutlierOutput	SpectraNucleus
	CenterVoxel	FixPseudoDiffSD	OutlierRange	SpectraOutputPlots
	ChainSteps	FlipAxes	OutputCalibration	SpectraPpmShift
	Channels	FlipBvec	OutputCheckImage	SpectraSamples
	Classes	FlipGrad	OutputCoilSurface	SpectraSpacing
	CleanUpSegmentations	FullOutput	OutputForm	SphereColor
	ClippingStyle	FullSphere	OutputImage	SphereSize
	CoilArrayPlot	GetMaskOnly	OutputLabels	SplineDegree
	CoilSamples	GetMaskOutput	OutputPlot	SplineKnotsNumber
	CoilSurfaceVoxelSize	GOutput	OutputSamples	SplineRegularization
	ColorFunction	GradType	OutputSense	SplineSpacingFactor
	ColorValue	GRegularization	OutputSNR	SplitMethod
	CompressNii	GridLines	OutputTransformation	StartPoints
	ContrastCoils	GridLinesSpacing	OutputType	StartPointsA
	ContrastCoilsA	GridLinesSpacingA	OutputTypeA	StartPointsAa
	ContrastCoilsB	GridLinesSpacingB	OutputTypeB	StartPointsBa
	ContrastCoilsB1	GridLinesSpacingB1	OutputTypeB1	StartPointsB1a
	ContrastCoilsB2	GridLinesSpacingB2	OutputTypeB2	StartPointsB2a
	ContrastCoilsB3	GridLinesSpacingB3	OutputTypeB3	StartPointsB3a
	ContrastCoilsB4	GridLinesSpacingB4	OutputTypeB4	StartPointsB4a
	ContrastCoilsB5	GridLinesSpacingB5	OutputTypeB5	StartPointsB5a
	ContrastCoilsB6	GridLinesSpacingB6	OutputTypeB6	StartPointsB6a
	ContrastCoilsB7	GridLinesSpacingB7	OutputTypeB7	StartPointsB7a
	ContrastCoilsB8	GridLinesSpacingB8	OutputTypeB8	StartPointsB8a
	ContrastCoilsB9	GridLinesSpacingB9	OutputTypeB9	StartPointsB9a
	ContrastCoilsB10	GridLinesSpacingB10	OutputTypeB10	StartPointsB10a
	ContrastCoilsB11	GridLinesSpacingB11	OutputTypeB11	StartPointsB11a
	ContrastCoilsB12	GridLinesSpacingB12	OutputTypeB12	StartPointsB12a
	ContrastCoilsB13	GridLinesSpacingB13	OutputTypeB13	StartPointsB13a
	ContrastCoilsB14	GridLinesSpacingB14	OutputTypeB14	StartPointsB14a
	ContrastCoilsB15	GridLinesSpacingB15	OutputTypeB15	StartPointsB15a
	ContrastCoilsB16	GridLinesSpacingB16	OutputTypeB16	StartPointsB16a
	ContrastCoilsB17	GridLinesSpacingB17	OutputTypeB17	StartPointsB17a
	ContrastCoilsB18	GridLinesSpacingB18	OutputTypeB18	StartPointsB18a
	ContrastCoilsB19	GridLinesSpacingB19	OutputTypeB19	StartPointsB19a
	ContrastCoilsB20	GridLinesSpacingB20	OutputTypeB20	StartPointsB20a
	ContrastCoilsB21	GridLinesSpacingB21	OutputTypeB21	StartPointsB21a
	ContrastCoilsB22	GridLinesSpacingB22	OutputTypeB22	StartPointsB22a
	ContrastCoilsB23	GridLinesSpacingB23	OutputTypeB23	StartPointsB23a
	ContrastCoilsB24	GridLinesSpacingB24	OutputTypeB24	StartPointsB24a
	ContrastCoilsB25	GridLinesSpacingB25	OutputTypeB25	StartPointsB25a
	ContrastCoilsB26	GridLinesSpacingB26	OutputTypeB26	StartPointsB26a
	ContrastCoilsB27	GridLinesSpacingB27	OutputTypeB27	StartPointsB27a
	ContrastCoilsB28	GridLinesSpacingB28	OutputTypeB28	StartPointsB28a
	ContrastCoilsB29	GridLinesSpacingB29	OutputTypeB29	StartPointsB29a
	ContrastCoilsB30	GridLinesSpacingB30	OutputTypeB30	StartPointsB30a
	ContrastCoilsB31	GridLinesSpacingB31	OutputTypeB31	StartPointsB31a
	ContrastCoilsB32	GridLinesSpacingB32	OutputTypeB32	StartPointsB32a
	ContrastCoilsB33	GridLinesSpacingB33	OutputTypeB33	StartPointsB33a
	ContrastCoilsB34	GridLinesSpacingB34	OutputTypeB34	StartPointsB34a
	ContrastCoilsB35	GridLinesSpacingB35	OutputTypeB35	StartPointsB35a
	ContrastCoilsB36	GridLinesSpacingB36	OutputTypeB36	StartPointsB36a
	ContrastCoilsB37	GridLinesSpacingB37	OutputTypeB37	StartPointsB37a
	ContrastCoilsB38	GridLinesSpacingB38	OutputTypeB38	StartPointsB38a
	ContrastCoilsB39	GridLinesSpacingB39	OutputTypeB39	StartPointsB39a
	ContrastCoilsB40	GridLinesSpacingB40	OutputTypeB40	StartPointsB40a
	ContrastCoilsB41	GridLinesSpacingB41	OutputTypeB41	StartPointsB41a
	ContrastCoilsB42	GridLinesSpacingB42	OutputTypeB42	StartPointsB42a
	ContrastCoilsB43	GridLinesSpacingB43	OutputTypeB43	StartPointsB43a
	ContrastCoilsB44	GridLinesSpacingB44	OutputTypeB44	StartPointsB44a
ContrastCoilsB45	GridLinesSpacingB45	OutputTypeB45	StartPointsB45a	
ContrastCoilsB46	GridLinesSpacingB46	OutputTypeB46	StartPointsB46a	
ContrastCoilsB47	GridLinesSpacingB47	OutputTypeB47	StartPointsB47a	
ContrastCoilsB48	GridLinesSpacingB48	OutputTypeB48	StartPointsB48a	
ContrastCoilsB49	GridLinesSpacingB49	OutputTypeB49	StartPointsB49a	
ContrastCoilsB50	GridLinesSpacingB50	OutputTypeB50	StartPointsB50a	
ContrastCoilsB51	GridLinesSpacingB51	OutputTypeB51	StartPointsB51a	
ContrastCoilsB52	GridLinesSpacingB52	OutputTypeB52	StartPointsB52a	
ContrastCoilsB53	GridLinesSpacingB53	OutputTypeB53	StartPointsB53a	
ContrastCoilsB54	GridLinesSpacingB54	OutputTypeB54	StartPointsB54a	
ContrastCoilsB55	GridLinesSpacingB55	OutputTypeB55	StartPointsB55a	
ContrastCoilsB56	GridLinesSpacingB56	OutputTypeB56	StartPointsB56a	
ContrastCoilsB57	GridLinesSpacingB57	OutputTypeB57	StartPointsB57a	
ContrastCoilsB58	GridLinesSpacingB58	OutputTypeB58	StartPointsB58a	
ContrastCoilsB59	GridLinesSpacingB59	OutputTypeB59	StartPointsB59a	
ContrastCoilsB60	GridLinesSpacingB60	OutputTypeB60	StartPointsB60a	
ContrastCoilsB61	GridLinesSpacingB61	OutputTypeB61	StartPointsB61a	
ContrastCoilsB62	GridLinesSpacingB62	OutputTypeB62	StartPointsB62a	
ContrastCoilsB63	GridLinesSpacingB63	OutputTypeB63	StartPointsB63a	
ContrastCoilsB64	GridLinesSpacingB64	OutputTypeB64	StartPointsB64a	
ContrastCoilsB65	GridLinesSpacingB65	OutputTypeB65	StartPointsB65a	
ContrastCoilsB66	GridLinesSpacingB66	OutputTypeB66	StartPointsB66a	
ContrastCoilsB67	GridLinesSpacingB67	OutputTypeB67	StartPointsB67a	
ContrastCoilsB68	GridLinesSpacingB68	OutputTypeB68	StartPointsB68a	
ContrastCoilsB69	GridLinesSpacingB69	OutputTypeB69	StartPointsB69a	
ContrastCoilsB70	GridLinesSpacingB70	OutputTypeB70	StartPointsB70a	
ContrastCoilsB71	GridLinesSpacingB71	OutputTypeB71	StartPointsB71a	
ContrastCoilsB72	GridLinesSpacingB72	OutputTypeB72	StartPointsB72a	
ContrastCoilsB73	GridLinesSpacingB73	OutputTypeB73	StartPointsB73a	
ContrastCoilsB74	GridLinesSpacingB74	OutputTypeB74	StartPointsB74a	
ContrastCoilsB75	GridLinesSpacingB75	OutputTypeB75	StartPointsB75a	
ContrastCoilsB76	GridLinesSpacingB76	OutputTypeB76	StartPointsB76a	
ContrastCoilsB77	GridLinesSpacingB77	OutputTypeB77	StartPointsB77a	
ContrastCoilsB78	GridLinesSpacingB78	OutputTypeB78	StartPointsB78a	
ContrastCoilsB79	GridLinesSpacingB79	OutputTypeB79	StartPointsB79a	
ContrastCoilsB80	GridLinesSpacingB80	OutputTypeB80	StartPointsB80a	
ContrastCoilsB81	GridLinesSpacingB81	OutputTypeB81	StartPointsB81a	
ContrastCoilsB82	GridLinesSpacingB82	OutputTypeB82	StartPointsB82a	
ContrastCoilsB83	GridLinesSpacingB83	OutputTypeB83	StartPointsB83a	
ContrastCoilsB84	GridLinesSpacingB84	OutputTypeB84	StartPointsB84a	
ContrastCoilsB85	GridLinesSpacingB85	OutputTypeB85	StartPointsB85a	
ContrastCoilsB86	GridLinesSpacingB86	OutputTypeB86	StartPointsB86a	
ContrastCoilsB87	GridLinesSpacingB87	OutputTypeB87	StartPointsB87a	
ContrastCoilsB88	GridLinesSpacingB88	OutputTypeB88	StartPointsB88a	
ContrastCoilsB89	GridLinesSpacingB89	OutputTypeB89	StartPointsB89a	
ContrastCoilsB90	GridLinesSpacingB90	OutputTypeB90	StartPointsB90a	
ContrastCoilsB91	GridLinesSpacingB91	OutputTypeB91	StartPointsB91a	
ContrastCoilsB92	GridLinesSpacingB92	OutputTypeB92	StartPointsB92a	
ContrastCoilsB93	GridLinesSpacingB93	OutputTypeB93	StartPointsB93a	
ContrastCoilsB94	GridLinesSpacingB94	OutputTypeB94	StartPointsB94a	
ContrastCoilsB95	GridLinesSpacingB95	OutputTypeB95	StartPointsB95a	
ContrastCoilsB96	GridLinesSpacingB96	OutputTypeB96	StartPointsB96a	
ContrastCoilsB97	GridLinesSpacingB97	OutputTypeB97	StartPointsB97a	
ContrastCoilsB98	GridLinesSpacingB98	OutputTypeB98	StartPointsB98a	
ContrastCoilsB99	GridLinesSpacingB99	OutputTypeB99	StartPointsB99a	
ContrastCoilsB100	GridLinesSpacingB100	OutputTypeB100	StartPointsB100a	

CardiacTools

Functions

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

Options

AxesMethod	GridLineSpacing	PlotLabel	ShowOutliers
BackgroundValue	GroupPerSegment	PlotRange	ShowPlot
BloodMaskRange	ImageSize	PlotStyle	SmoothHelix
BullPlotMethod	LCMMethod	RadialSamples	StartPoints
ColorFunction	MakeSegmentPlots	ReverseDirection	StartSlices
CutOffMethod	MaskWallMap	ReversePoints	TextNumberForm
DistanceMeasure	Method	RowSize	TextOffset
DropSamples	OutputCheckImage	SegmentationMethod	TextSize

CoilTools

Functions

CoilSNRCalc	LoadCoilSetup	MakeCoilLayout	MakeWeightMask
FindCoilPosition	LoadCoilTarget	MakeNoisePlots	

Options

CoilArrayPlot	ColorFunction	OutputCoilSurface
CoilSurfaceVoxelSize	ImageSize	PlotRange

DenoiseTools

Functions

AnisoFilterData
AnisoFilterTensor

DeNoise
DenoiseCSIdata

DenoiseDynamicSpectraData
NNDeNoise

PCADeNoise
WeightMapCalc

Options

AnisoFilterSteps
AnisoItterations
AnisoKappa
AnisoKernel
AnisoStepTime

AnisoWeightType
DeNoiseIterations
DeNoiseKernel
DeNoiseMonitor
Method

MonitorCalc
NNThreshold
PCAClipping
PCAComplex
PCAKernel

PCANoiseSigma
PCAOutput
PCATolerance
PCAWeighting

DixonTools

Functions

DixonPhase
DixonReconstruct
DixonToPercent
FindInPhaseEchos

FixDixonFlips
GenerateAmps
OptimizeDixonEcho
SimulateDixonSignal

Unwrap
UnwrapDCT
UnwrapList
UnwrapSplit

Wrap

Options

DixonAmplitudes
DixonBipolar
DixonClipFraction
DixonConstrainPhase
DixonCorrectT1
DixonFieldStrength
DixonFilterInput

DixonFilterOutput
DixonFilterSize
DixonFilterType
DixonFitPhase
DixonFixT2
DixonFrequencies
DixonIterations

DixonMaskThreshold
DixonNucleus
DixonPhases
DixonPrecessions
DixonRelaxivity
DixonTolerance
MaxIterations

MonitorCalc
MonitorUnwrap
PhaseEchos
UnwrapDimension
UnwrapThresh

ElastixTools

Functions

ReadTransformParameters	RegisterDataSplit	RegisterDiffusionData	TransformData
RegisterCardiacData	RegisterDataTransform	RegisterDiffusionDataSplit	\$debugElastix
RegisterData	RegisterDataTransformSplit	RegisterTensorData	

Options

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

FasciculationTools

Functions

AnalyzeActivations	EvaluateActivation	FindActivations	SelectActivations
--------------------	--------------------	-----------------	-------------------

Options

ActivationBackground	ActivationOutput	ActivationThreshold	MaskDilation
ActivationIterations	ActivationSize	IgnoreSlices	ThresholdMethod

GeneralTools

Functions

ApplyCrop	FileSelect	MedFilter	RotateDimensionsLeft
AutoCropData	FindCrop	MedianNoZero	RotateDimensionsRight
BSplineCurveFit	FindMaxDimensions	MemoryUsage	RotationMatrixToQuaternion
ClearTemporaryVariables	GetAssetLocation	MonitorCalc	RotationMatrixToQuaternionVector
CompilableFunctions	GridData	NiiFileExistQ	SaveImage
ConvertExtension	GridData3D	NNLeastSquares	SignNoZero
CropData	GyromagneticRatio	PadToDimensions	Squeeze
CutData	LapFilter	QMRIToolsFuncPrint	StandardDeviationNoZero
DataToVector	LLeastSquares	QMRIToolsFunctions	StdFilter
DecomposeAffineMatrix	LogNoZero	QMRIToolsPackages	StichData
DecomposeScaleMatrix	MADNoZero	QuaternionToRotationMatrix	StringPadInteger
DevideNoZero	MakeCoordinates	QuaternionVectorToRotationMatrix	SumOfSquares
DynamicPartition	MakeFunctionGraph	RescaleData	TensMat
EmptyDirectoryQ	MakeIntFunction	ReverseCrop	TensVec
ExpNoZero	MBCount	ReverseDimensions	VectorToData
ExtractDemoData	MeanNoZero	RMSNoZero	

Options

AllowSelfDependencies	CropOutput	InterpolationOrder	PadValue
CenterRange	CropPadding	LabelPlacement	SplineDegree
CenterVoxel	FileType	OutputWeights	SplineKnotsNumber
CropAlways	ImageResolution	Padding	SplineRegularization
CropInit	ImageSize	PadDirection	WindowTitle

GradientTools

Functions

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

Options

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

IVIMTools

Functions

BayesianIVIMFit2
 BayesianIVIMFit3
 CorrectParMap
 FConvert

FConverti
 FracCorrect
 HistogramPar
 IVIMCalc

IVIMCorrectData
 IVIMFunction
 IVIMResiduals
 MeanBvalueSignal

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

LoggingTools

Functions

AddToLog
 CheckFile
 DirectoryTree

ExportDirectoryTree
 ExportLog
 ImportLog

MakeCheckFile
 PrintDirectoryTree
 ResetLog

ShowLog

Options

MaskingTools

Functions

DilateMask	MaskData	ReplaceSegmentations	SmoothMask
FitGradientMap	MaskSegmentation	RescaleSegmentation	SmoothSegmentation
GetCommonSegmentation	MergeSegmentations	SegmentationVolume	SplitSegmentations
GetSegmentationLabels	NormalizeData	SegmentMask	
HomogenizeData	NormalizeMeanData	SelectMaskComponents	
Mask	RemoveMaskOverlaps	SelectSegmentations	

Options

FitOrder	MaskComponents	MaskFiltKernel	NormalizeMethod
MaskClosing	MaskDilation	MaskSmoothing	SmoothIterations

MuscleBidsTools

Functions

AddToJson	GenerateBidsName	MuscleBidsMerge	SelectBidsFolders
BidsDcmToNii	GetConfig	MuscleBidsProcess	SelectBidsSessions
CheckDataDiscription	GetJSONPosition	MuscleBidsSegment	SelectBidsSubjects
ExtractFromJSON	ImportJSON	MuscleBidsTractography	ViewConfig
GenerateBidsFileName	MergeJSON	PartitionBidsFolderName	
GenerateBidsFolderName	MuscleBidsConvert	PartitionBidsName	

Options

BidsIncludeSession	DeleteAfterConversion	VersionCheck
BidsTractographyMethod	SelectSubjects	

NiftiTools

Functions

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

Options

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

PlottingTools

Functions

ColorFAPlot
GenerateRotationFrames
GetSliceData
GetSlicePositions
GradientPlot

Link3DGraphic
ListSpherePlot
MakeSliceImages
PlotContour
PlotCorrection

PlotData
PlotData3D
PlotDefGrid
PlotDuty
PlotIVIM

PlotMoments
PlotSegmentations
PlotSequence

Options

ClippingStyle
ColorFunction
ContourColor
ContourColorRange
ContourOpacity
ContourResolution

ContourScaling
ContourSmoothRadius
DropSlices
ImageLegend
ImageOrientation
ImageSize

MakeCheckPlot
Method
NormalizeIVIM
PeakNumber
PlotColor
PlotRange

PlotSpace
PositiveZ
RandomizeColor
SphereColor
SphereSize

ProcessingTools

Functions

B1MapCalc
B1Shimming
CombineB1
CorrectJoinSetMotion
DataTransformation
DatTot
DatTotXLS
ErrorPlot

FindOutliers
FitData
GetMaskData
GetMaskMeans
GetTractMeans
Hist
Hist2
InvertDataset

JoinSets
MeanRange
MeanSignal
MeanStd
MedCouple
NumberTableForm
ParameterFit
ParameterFit2

RotateData
RotateTensor
SmartMask
SNRCalc
SNRMapCalc
SplitSets

Options

AxesLabel
B1EqualPower
B1FilterData
B1Masking
B1MaxPower
B1Output
B1Scaling
B1ShimMethod
ColorValue
FitFunction
FitOutput
GetMaskOnly

GetMaskOutput
ImageSize
InterpolationOrder
JoinSetSplit
MaskCompartment
MeanMethod
Method
MonitorCalc
MotionCorrectSets
NormalizeOverlap
NormalizeSets
OutlierIncludeZero

OutlierIterations
OutlierMethod
OutlierOutput
OutlierRange
OutputSNR
PaddOverlap
PadOutputDimensions
PlotLabel
ReferenceB1
ReverseData
ReverseSets
Scaling

SmartMaskOutput
SmartMethod
SmoothSNR
Strictness
TableAlignments
TableDepth
TableDirections
TableHeadings
TableMethod
TableSpacing
UseMask

ReconstructionTools

Functions

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

Options

AcquisitionMethod	EchoShiftData	NormalizeOutputSpectra	RescaleRecon
CoilSamples	HammingFilter	OutputSense	SenseRescale
DeconvolutionMethod	Method	ReconFilter	WienerRegularization

RelaxometryTools

Functions

CalibrateEPGT2Fit	EPGSignal	ShiftPulseProfile	T2Fit
CreateT2Dictionary	EPGT2Fit	T1Fit	TriExponentialT2Fit
DictionaryMinSearch	NonLinearEPGFit	T1rhoFit	

Options

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

ScientificColorData

Functions

AddScientificColors	ExtractColorData
---------------------	------------------

Options

SegmentationTools

Functions

AddLossLayer	DiceLossLayer	MakeClassifyImage	PrepareTrainingData
AnalyseNetworkFeatures	DiceSimilarity	MakeClassifyNetwork	SegmentData
ApplySegmentationNetwork	FocalLossLayer	MakeClassImage	SegmentDataGUI
AugmentImageData	GetNeuralNet	MakeDistanceMap	ShowTrainLog
AugmentTrainingData	GetTrainData	MakeNode	SplitDataForSegementation
ChangeNetDimensions	ImportITKLabels	MakeUnet	SurfaceDistance
CheckSegmentation	JaccardLossLayer	MuscleLabelToName	TrainSegmentationNetwork
ClassDecoder	JaccardSimilarity	MuscleNameToLabel	TverskyLossLayer
ClassEncoder	MakeChannelClassGrid	NetDimensions	\$debugUnet
ClassifyData	MakeChannelClassImage	NetSummary	
DataToPatches	MakeChannelImage	PatchesToData	

Options

Channels	DistanceRange	LossFunction	PatchesPerSet
Classes	DownsampleSchedule	MaxPatchSize	PatchNumber
Dimensions	DropoutRate	MaxTrainingRounds	PatchPadding
ActivationType	FeatureSchedule	Method	PatchSize
AugmentData	ImageSize	Monitor	RescaleMethod
BatchSize	InputLabels	MonitorCalc	RoundLength
BlockType	L2Regularization	MonitorInterval	SettingSchedule
CleanUpSegmentations	LabelTag	NetworkArchitecture	TargetDevice
DataPadding	LearningRate	NetworkDepth	TestRun
DataTag	LoadTrainingData	OutputLabels	

SimulationTools

Functions

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

Options

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

SpectroTools

Functions

ApodizeEcho
 ApodizeFid
 ApodizePadEcho
 ApodizePadFid
 ApodizePadSpectra
 ApodizeSpectra
 ChangeDwellTimeFid
 CompareFidFitPlot
 CompareSpectraFitPlot
 CorrectTEFid

CorrectTESpec
 CSIIInterface
 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
 Filling
 FineTuneFit
 FitLineShape
 GridLines

GridLineSpacing
 ImageSize
 InitializeFit
 Method
 PaddingFactor
 PlotColor
 PlotLabel
 PlotLabels

PlotRange
 ReadoutType
 SparID
 SparName
 SparOrientation
 SpectraBandwidth
 SpectraFieldStrength
 SpectraNucleus

SpectraOutputPlots
 SpectraPpmShift
 SpectraSamples
 SpectraSpacing
 SplineSpacingFactor

TaggingTools

Functions

AnalyzeTagging
 CalculateDisplacementParameters

Options

HistoryWeighting
 MonitorTagging

TensorTools

Functions

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

Options

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

TractographyTools

Functions

ExportTracts	FitTracts	PlotTracts	TractAngleMap
FiberLength	GetTractValues	ResampleTracts	TractDensityMap
FiberTractography	ImportTracts	RescaleTracts	TractLengthMap
FilterTracts	MoveTracts	SeedDensityMap	
FindTensorPermutation	PlotSegmentedTracts	SegmentTracts	

Options

Boxed	ImageSize	NormalizeDensity	TensorPermutations
ColorFunction	InterpolationOrder	OutputForm	TracMonitor
FiberAngle	MaxSeedPoints	PerformanceGoal	TractColoring
FiberLengthRange	MaxTracts	StepSize	TractReduction
FittingOrder	Method	StopThreshold	TractScaling
FitTractSegments	Monitor	TensorFilps	TractSize

```
In[ ]:= QMRIToolsFuncPrint[]
```

CardiacTools

Functions

Symbol i

BullseyePlot[data, segmask] generates a AHA-17 segment bullseye plot.

BullseyePlot[list] generates a AHA-17 segment 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 i

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 respectively.

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

Documentation [Local »](#)

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

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`CardiacCoordinateSystem



Symbol



CardiacSegment[mask, vox, pts] segments the mask in the AHA17 segmentation using pts to indicate the attachemnts.

CardiacSegment[mask, back, vox, pts] the same where back is used for image generation.

CardiacSegment[mask, vox, pts, seg] does the same but seg can be an alternate segmentation to the AHA17.

CardiacSegment[mask, back, vox, pts, seg] does the same but seg can be an alternate segmentation to the AHA17 where back is used for image generation.

Documentation [Local »](#)

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

Options {ReversePoints → True, ReverseDirection → False, MakeSegmentPlots → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`CardiacSegment



Symbol i

CardiacSegmentGUI[data, mask, vox] allows to segment the heart in 1, 4, 6 or AHA–17 segments 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 {points, slices , {rev, flip}}.

Documentation [Local »](#)

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

Options ► StartPoints → Default ... (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`CardiacSegmentGUI

^

Symbol i

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 i

CreateHeart[] creates a simulated left ventricle shape.

CreateHeart[pars] creates a simulated left ventricle shape with predefined 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 i

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 assumed to be in ms.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name `QMRITools`CardiacTools`ECVCalc`

^

Symbol i

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 dimensions {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 i

GetMaskSegmentPoints[mask] get the attachment points from a cardiac segmentation where the heart has label 1, and the attachment points have label 2 and 3.

Output is {maks, pts} where now in mask the points are removed.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[GetMaskSegmentPoints] = {ArgumentsPattern -> {...}}`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`CardiacTools`GetMaskSegmentPoints`

^

Symbol i

GetSegmentLines[lines, lineIndex, segments] groups the transmural lines per segment.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[GetSegmentLines] = {ArgumentsPattern -> {...}}`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`CardiacTools`GetSegmentLines`

^

Symbol i

GetSegmentSlices[mask] based on the mask it gives back the slice numbers of the apex, apical, mid-ventircal, and basal slices.
GetSegmentSlices[points] does the same but then based on the points obtained form GetMaskSegmentPoints.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[GetSegmentSlices] = {ArgumentsPattern -> {...}}`

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 ventricle 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 following values are calculated automatically 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 calculated with CentralAxes.

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

Output is the 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, LCMMMethod → WallMap, AxesMethod → Quadratic}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`HelixAngleCalc



Symbol i

LinesToSegmentIndex[lines, points, segments] finds the lines indices corresponding to the points and the segments borders. Additionally it finds all the lines indices for all lines within each segment. The lines are computed by MaskToLines, the points are computed by GetMaskSegmentPoints, and the segments is the output of SegmentsPerSlices.

Output {pointIndex, segmentIndex, lineIndex}.

Documentation [Local](#) »

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

Options {ReversePoints → True, ReverseDirection → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`LinesToSegmentIndex

^

Symbol i

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 assumed 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 i

MakeLineImage[back, segLines, pts] makes an image of the cardiac segmentation lines.

Documentation [Local](#) »

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MakeLineImage

^

Symbol i

MakeMaskImage[back, mask] masks an image of the cardiac segmentation mask.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MakeMaskImage

^

Symbol i

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 i

MaskToLines[mask, vox] calculates lines perpendicular to the heart wall per slice within the mask. Internally it uses CalculateWallMap and CentralAxes to obtain the cardiac geometry from mask.

MaskToLines[mask, wall, cent] where mask is the first output of CalculateWallMap and cent is the first output of CentralAxes.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MaskToLines

^

Symbol i

PlotSegmentMask[mask, segmask, vox] plots the mask segments 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 i

PlotSegments[mask, data, segang] shows how the heart will 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 i

RadialSample[mask, data, segang] radially 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 i

SegmentLinesToMask[mask, segLines] cuts the mask based one the tranmural lines per segments which can be obtained by GetGesmentLines.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`SegmentLinesToMask

^

Symbol i

SegmentsPerSlice[points] gives the number of segments per slice where the slice distribution is determined by GetSegmentSlices.

SegmentsPerSlice[slices, points] does the same but the slices are given manually.

Documentation [Local »](#)

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

Options {GroupPerSegment → True, SegmentationMethod → AHA}

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`SegmentsPerSlice

^

Symbol i

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 i

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 i

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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`BackgroundValue

^

Symbol i

BloodMaskRange is an option for MakeECVBloodMask.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`BloodMaskRange

^

Symbol i

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 i

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 i

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 i

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 i


DropSamples is an option for RadialSample and PlotSegments. Defines how many samples are dropped 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 


GroupPerSegment is an option for SegmentsPerSlice. If set False segments are grouped per slice and not per segment.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`GroupPerSegment

^

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 


LCMMMethod is an option for HelixAngleCalc and LMCSysCalc. Can be "CentralAxes" or "WallMap".
"CentralAxes" uses wall distance calculation using projection of the central axes and circular approximation of the ventricle. This method is fairly fast and uses CentralAxes internally.
"WallMap" uses wall distance interpolation and subsequential gradient calculation. Can take long for high res datasets but is most accurate. Uses CalculateWallMap internally.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`LCMMMethod

^

Symbol 


MakeSegmentPlots is an option for CardiacSegment. If True plots of the segmentation are made.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`MakeSegmentPlots

^

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 i

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 i

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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`RadialSamples

^

Symbol i

ReverseDirection is an option for LinesToSegmentIndex, CardiacSegment. Defines the direction of rotation, clockwise or anti-clockwise, can be True or False.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`ReverseDirection

^

Symbol i

ReversePoints is an option for LinesToSegmentIndex, CardiacSegment. Defines at which point to start, can be True or False.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`ReversePoints

^

Symbol i


RowSize is an option for CentralAxes. defines the number of images per showing the segmentation.
Can be "Automatic" or an integer.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`RowSize

^

Symbol 


SegmentationMethod is an option for SegmentsPerSlice. Values can be "AHA", "AHA+", 1, 2, 3, 6 or 8.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`CardiacTools`SegmentationMethod

^

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 CardiacSegmentGUI. 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 CardiacSegmentGUI. 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 i

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 i

FindCoilPosition[weights] finds the coil position by locating the highest intensity location in the coil weight map, which can be obtained 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 i

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

The input file is the Nii file that contains the individually reconstructed coil images and the noise data.

Internally 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 i

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 i

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 i

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 i

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 i

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 i

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

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`CoilTools`CoilSurfaceVoxelSize`

^

Symbol i

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 i


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 and 10.1016/j.mri.2009.10.001 and 10.1371/journal.pone.0126953.

Documentation [Local »](#)

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

Options {AnisoStepTime → 1, AnisoIterations → 1, AnisoKernel → {0.25, 0.5}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoFilterData

^

Symbol i

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.

AnisoFilterTensor[tens] Same but does not use the data for edge identification.

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 i

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

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 i

DenoiseCSldata[spectra] performs 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[DenoiseCSldata] = {ArgumentsPattern → {_, OptionsPattern[]}}`

Options `{PCAKernel → 5, PCANoiseSigma → Corners}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`DenoiseTools`DenoiseCSldata`

^

Symbol i

DenoiseDynamicSpectraData[spectra] performs 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 i

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 `NNThreshold → 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 voxels, 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 ... (8 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCADeNoise



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 i

AnisoIterations is an options for AnisoFilterData. It specifies the amount of denoising iterations.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoIterations

^

Symbol i

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 i

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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`AnisoKernel

^

Symbol i

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 i


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 


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 many processing functions. When true the proceses of the calculation is shown.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MonitorCalc

^

Symbol 


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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`NNThreshold

^

Symbol 


PCAClipping is an option of PCADeNoise and can be True or 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 


PCAComplex is an option of PCADeNoise and can be True or False. If set true the input data is expected to be {real, imag}.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DenoiseTools`PCAComplex

^

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 i

PCAOOutput 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`PCAOOutput

^

Symbol i

PCATolerance 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`PCATolerance

^

Symbol i

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 i

DixonPhase[real, imag, echos] calculates the b0 and ph0 maps.

Documentation [Local »](#)

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

Options [DixonPrecessions → 1 ... \(9 total\)](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonPhase

^

Symbol i

DixonReconstruct[{real, imag}, echo] reconstructs Dixon data with initial guess $b_0 = 0$ and $T_{2\text{star}} = 0$.

DixonReconstruct[{real, imag}, echo, {b0}] reconstructs Dixon data with initial guess $T_{2\text{star}} = 0$.

DixonReconstruct[{real, imag}, echo, {b0, t2}] reconstructs Dixon data with $t_{2\text{star}}$ and B_0 .

DixonReconstruct[{real, imag}, echo, {b0, t2, ph0}] reconstructs Dixon data with initial phase.

DixonReconstruct[{real, imag}, echo, {b0, t2, ph0, phb}] reconstructs Dixon data with bipolar phase.

Output is $\{\{\text{watF}, \text{fatF}\}, \{\text{watSig}, \text{fatSig}\}, \{\text{inphase}, \text{outphase}\}, \{\{b_0, \text{ph0}, \text{phb}\}, \{t_2, t_2\}\}, \text{iterations}\}$.

The fractions are between 0 and 1, the B_0 field map is in Hz and the $T_{2\text{star}}$ 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 ... (20 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonReconstruct

^

Symbol i

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

Output is {waterFraction, fatFraction}.

The values of water and fat are arbitrary units and the output fractions are between 0 and 1.

DixonToPercent[] is based on DOI: 10.1002/mrm.21301.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonToPercent

^

Symbol i

FindInPhaseEchos[echos, iop] finds the two nearest echos to inphase which are best used for unwrapping using the iop time.

Documentation [Local »](#)

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

Options `DixonBipolar → False`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`DixonTools`FindInPhaseEchos`

^

Symbol i

FixDixonFlips[{mag, phase, real, imag}] checks if any volumes are 180 degrees out of phase and corrects them.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`DixonTools`FixDixonFlips`

^

Symbol i

GenerateAmps[amp] generates the fat based chemical species amplitudes needed for DixonReconstruct.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`DixonTools`GenerateAmps`

^

Symbol i

OptimizeDixonEcho[] shows a manipulate pannel which allos to optimize the dixon echos.

OptimizeDixonEcho[echos] shows a manipulate pannel which allos to optimize the predefined dixon echos.


Documentation [Local »](#)

Options `{DixonNucleus → 1H, DixonFrequencies → {{0.}, {-3.81, -3.4, -3.12, -2.67, -2.45, -1.94, -0.63, -0.4, 0.52, 0.62}}, DixonAmplitudes → {{1}, {0.089, 0.577, 0.059, 0.093, 0.059, 0.013, 0.02, 0.01, 0.059}}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`DixonTools`OptimizeDixonEcho`

^


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 π .

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 -> False, UnwrapDimension -> 2D, UnwrapThresh -> 0.5}

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`Unwrap



Symbol 

UnwrapDCT[data] unwraps the given dataset using DCT transform . The data should be between $-\pi$ and π .


UnwrapDCT[] is based on DOI: 10.1364/JOSAA.11.000107.


Documentation [Local »](#)

Default Definitions `SyntaxInformation[UnwrapDCT] = {ArgumentsPattern -> {_, _}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`UnwrapDCT



Symbol 


UnwrapList[list] unwraps a 1D list of values between $-\pi$ and π .

Documentation [Local »](#)

Default Definitions `SyntaxInformation[UnwrapList] = {ArgumentsPattern -> {_, _}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`UnwrapList



Symbol i

UnwrapSplit[phase, data] unwraps 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 π . 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 → False, UnwrapDimension → 2D, UnwrapThresh → 0.5}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`DixonTools`UnwrapSplit`

^

Symbol i

Wrap[data] wraps phase values between $-\pi$ and π .

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`DixonTools`Wrap`

^

Options

Symbol i

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

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`DixonTools`DixonAmplitudes`

^

Symbol i


DixonBipolar is an option for FindInPhaseEchos. If set True the found echos will always be of the same polaritiy.

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 


DixonConstrainPhase is an option for DixonReconstruct. If set True everything is fitted as real values.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonConstrainPhase

^

Symbol 


DixonCorrectT1 is an option for DixonReconstruct. To perform T1 correction provide the TR and FA as a list, {TR, FA}. TR is in ms and FA in degrees.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonCorrectT1

^

Symbol 


DixonFieldStrength is an option 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 option 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 i

DixonFilterOutput is an option 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 i

DixonFilterSize is an option 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 i

DixonFilterType is an option for DixonReconstruct. FilterType can me "Median" or "Laplacian".

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonFilterType

^

Symbol

DixonFitPhase is an option for DixonReconstruct. If set to True the phase maps are fitted using polinomial functions.

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonFitPhase

^

Symbol i

DixonFixT2 is an option for DixonReconstruct. If set to true the R2' is fitted rather than the R2*. This is done by fixing T2–water to 30ms and T2–fat to 100ms.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonFixT2

^

Symbol i


DixonFrequencies is an option 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 option for DixonReconstruct. Defines the maximum iterations the fit can use.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonIterations

^

Symbol 


DixonMaskThreshold is an option for DixonReconstruct. Defines at which threshold the dixon reconstruction considers a voxel to be background noise. Default values is 0.05.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonMaskThreshold

^

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 


DixonPhases is an option for DixonReconstruct. It defines which phases to fit within the model.
The order is {T2*, B0, bipolar, initial, bipolar}.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonPhases

^

Symbol 


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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonPrecessions

^

Symbol 


DixonRelativity is an option for DixonReconstruct. Defines the T2 relaxation times of the fat peaks being used. Is only used if DixonFixT2 is set to True.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonRelativity

^

Symbol 


DixonTolerance is an option for DixonReconstruct. Defines at which change per iteration of b0 and R2star the iterative methods stops. Default value is 0.1.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`DixonTolerance

^

Symbol 


MaxIterations is an option that specifies the maximum number of iterations that should be tried in various built-in functions and algorithms.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected}

Full Name System`MaxIterations

^

Symbol 


MonitorCalc is an option for many processing functions. When true the process of the calculation is shown.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MonitorCalc

^

Symbol 

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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`MonitorUnwrap

^

Symbol i

PhaseEchos is an option for DixonPhase. Can be Automatic or a list of two integers.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`DixonTools`PhaseEchos

^

Symbol i

UnwrapDimension is an option for Unwrap. Can be "2D" or "3D". 2D is for unwarping 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 i

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 i

ReadTransformParameters[directory] reads the tranformation 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 i

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 within 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 ... (18 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterCardiacData

^

Symbol i

RegisterData[data] registers the data series. If data is 3D it performs multiple 2D registration, if data is 4D it performs multiple 3D registration.

The input data can be in the forms: data, {data, vox}, {data, mask} or {data, mask, vox}.

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.

The inputs target and moving can be in the forms: data, {data, vox}, {data, mask} or {data, mask, vox}.

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 ... (17 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterData

^

Symbol i

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 ... (18 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterDataSplit

^

Symbol i

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 registration is done the moving2nd data is deformed according to the output of the registration 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 ... (18 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterDataTransform

^

Symbol



RegisterDataTransformSplit[target, moving, {moving2nd, vox}] is identical to RegisterDataTransform with the same functionality as RegisterDataSplit. This means the data is split in two using the function CutData and merged with Stich data.

Output is {registered moving, deformed moving2nd}.

Documentation [Local »](#)

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

Options ▶ Iterations → 250 ... (19 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, vox}] registers a diffusion dataset. The diffusion data is also registered to the anatdata.

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

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

RegisterDiffusionData[{dtidata, dtimask, vox}, {anatdata, anatmask, vox}] 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 ... (24 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterDiffusionData



Symbol i

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

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

RegisterDiffusionDataSplit[{dtidata, dtimask, vox}, {anatdata, anatmask, vox}] 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 i

RegisterTensorData[target, moving] registers the moving tensor to the target tensor. After registration the tensor is reoriented appropriately.

The inputs target and moving can be in the forms: tensor, {tensor, vox}, {tensor, mask} or {tensor, mask, vox}.

RegisterTensorData[target, moving, tensor] registers the moving data to the target data and applies the deformation to the tensor.

The inputs target and moving can be in the form: data, {data, vox}, {data, mask} or {data, mask, vox}.

The input tensor can be in the form: tensor or {tensor, vox}.

only "rigid", "affine" and "bspline" can be used as MetReg.

Documentation [Local »](#)

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

Options [►](#) Iterations → 250 ... (17 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`RegisterTensorData

^

Symbol i

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... (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`TransformData

^

Symbol i

\$debugElastix is a parameter that allows to print Elastix commands if set to True.

Documentation [Local »](#)

Attributes {ReadProtected}

Full Name QMRITools`ElastixTools`\$debugElastix

^

Options

Symbol i

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 i

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 i

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 i

DeleteTempDirectory an options for RegisterData, RegisterDiffusionData, RegisterCardiacData and RegisterDataTransform.
It specifies if the temp directory should be deleted after the registration is finished.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools'ElastixTools>DeleteTempDirectory

^

Symbol i

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 i

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 i

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 i

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 i

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 i

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 i

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 i

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 i

MethodReg is an options for RegisterData, RegisterDiffusionData, RegisterCardiacData and RegisterDataTransform.
It specifies which registration method to use.
Methods 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 i

MethodRegA is an option for RegisterDiffusionData.
It specifies which registration method to use when registering diffusion data to anatomical space. Methods can be be "rigid", "affine" or "bspline".

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`ElastixTools`MethodRegA

^

Symbol i

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 i

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 i

OutputImage is an options for RegisterData, RegisterDiffusionData, and RegisterDataTransform. It specifies if the result image should be written in the TempDirectory as nii file.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools'ElastixTools'OutputImage

^

Symbol i

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 i

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 i

PrintTempDirectory is an options for RegisterData, RegisterDiffusionData, RegisterCardiacData and RegisterDataTransform. It spefifies if the location of the temp directory should be deployed.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools'ElastixTools'PrintTempDirectory

^

Symbol i

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 i

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 i

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 i

ShowMetric is an option for RegisterData. If set to True it will print a plot of the Metric.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`ElastixTools`ShowMetric

^

Symbol i

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 i

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 i

TransformMethod is an option for RegisterDataSplit and RegisterDataTransformSplit. values can be "Data", "Mask", "Segmentation". If set to "Mask" a binary mask is expected as the second moving input. If set to "Segmentation" a multi label segmenation is expected as the second moving input. Uses SplitSegmenations internally then.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`TransformMethod

^

Symbol i

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

^

FasciculationTools

Functions

Symbol i

AnalyzeActivations[actMap, mask] Analysis of the activation map generated from the mask.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`FasciculationTools`AnalyzeActivations

^

Symbol i

EvaluateActivation[out] allows to evaluate the activation detection using FindActivations, where out is the output of that function with the option Activationoutput set to True.

EvaluateActivation[out, actS] The same with the extra analysis of the SelectActivations function output given as actS.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`FasciculationTools`EvaluateActivation`

^

Symbol i

FindActivations[data] Finds the activation in MUMRI or DTI data after data normalization.

FindActivations[data, mask] Finds the activation in MUMRI or DTI data after data normalization within the mask.

Documentation [Local »](#)

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

Options ▶ `ActivationThreshold → {3., 0.6} ... (7 total)`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`FasciculationTools`FindActivations`

^

Symbol i

SelectActivations[act] selects the activations above the given ActivationSize.

SelectActivations[act, vox] selects the activations above the given ActivationSize where the activation size is in mm^3 .

SelectActivations[act, mask] selects the activations above the given ActivationSize within the given mask or masks. The mask can be 3D or 4D.

SelectActivations[act, {mask, back}] selects the activations above the given ActivationSize within the given mask or masks. All voxels outside the back are ignored.

Output is {actSelected, actTotal} is mask is 3D.

Output is {{actSelected, Total[actSelected]}, {actTotal, Total[actTotal]}} is mask is 4D where actSelected and actTotal are per mask.

Documentation [Local »](#)

Options `{ActivationSize → 4, IgnoreSlices → {0, 0}}`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`FasciculationTools`SelectActivations`

^

Options

Symbol i

ActivationBackground is an option for FindActivations. If all normalized signals, which range between 0–150, are below this value the algorithm does nothing.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools\FasciculationTools\ActivationBackground

^

Symbol i

ActivationIterations is an option for FindActivations. The maximum number of iteration that can be used for activation detection.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools\FasciculationTools\ActivationIterations

^

Symbol i

ActivationOutput is an option for ActivationOutput. If set to All also the mn and treshhold values are retured.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools\FasciculationTools\ActivationOutput

^

Symbol i

ActivationSize is an option for SelectActivations. Its the size of the activations selected defined in number of voxels if no voxel size is given. If a voxel size is given its the volume.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools\FasciculationTools\ActivationSize

^

Symbol i

ActivationThreshold is an option for FindActivations. Fist value is the number of standard deviations second is the pecentage threshold.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools\FasciculationTools\ActivationThreshold

^

Symbol i

IgnoreSlices is an option for FindActivations and SelectActivations. Determines how many slices of the start and end of the dataset are ignored.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`FasciculationTools`IgnoreSlices

^

Symbol i

MaskDilation is an option for Mask, SmoothMask and SmoothSegmentation. 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 i

ThresholdMethod is an option for FindActivations. Values can be "StandardDeviation", "Fraction" or "Both".

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`FasciculationTools`ThresholdMethod

^

GeneralTools

Functions

Symbol i

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

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

Documentation [Local »](#)

Default Definitions `SyntaxInformation[ApplyCrop] = {ArgumentsPattern -> {_, _, _}, OptionsPattern[]}`

Options CropAlways -> False

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`ApplyCrop

^

Symbol i

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

Documentation [Local »](#)

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

Options `CropPadding → 5`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`AutoCropData`

^

Symbol i

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 i

ClearTemporaryVariables[] Clear temporary variables.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`ClearTemporaryVariables`

^

Symbol i

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

Documentation [Local »](#)

Default Definitions `SyntaxInformation[CompilableFunctions] = {ArgumentsPattern → {}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`CompilableFunctions`

^

Symbol i

ConvertExtension[filename, extension] checks if file has correct extension. Removes .gz or changes the extension or adds extension if not present.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`ConvertExtension

^

Symbol i

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 i

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 i

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

DataToVector[data, mask] convert 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 i

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 i

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 i

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 i

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

DynamicPartition[data,part,last] partitions 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 i

EmptyDirectoryQ[dir] checks if directory dir is empty.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`GeneralTools`EmptyDirectoryQ`

^

Symbol i

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 i

ExtractDemoData[] Extracts the demo data archive.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name `QMRITools`GeneralTools`ExtractDemoData`

^

Symbol i

FileSelect[action] creates a systemdialog which returns 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 i

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 i

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 i

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

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

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`GetAssetLocation`

^

Symbol i

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 i

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 i

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 i

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 i

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 i

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 i

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 i

MakeCoordinates[data, vox] gives the coordiantes of every voxel.

MakeCoordinates[dim, vox] gives the coordiantes of every voxel for a dataset with dimensions dim.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`GeneralTools`MakeCoordinates`

^

Symbol i

MakeFunctionGraph[function] makes a function dependency graph of the function.

Documentation [Local »](#)

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

Options `{LabelPlacement → Tooltip, AllowSelfDependencies → False}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`MakeFunctionGraph`

^

Symbol i

MakeIntFunction[data] makes an interpolation function of the data using voxel size `{1, 1, 1}` and interpolation order 1

MakeIntFunction[data, int] makes an interpolation function of the data using voxel size `{1, 1, 1}` and interpolation order `int`.

MakeIntFunction[data, vox ,int] makes an interpolation function of the data using voxel size `vox` and interpolation order `int`.

Documentation [Local »](#)

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

Options `{CenterVoxel → True, CenterRange → False}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`MakeIntFunction`

^

Symbol i

MBCount[expr] gives the memory usage of the expression in MB.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`MBCount`

^

Symbol i

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 i

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

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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MedFilter

^

Symbol i

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 i

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 amount of definitions to show.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MemoryUsage

^

Symbol i

MonitorCalc is an option for many processing functions. When true the processes of the calculation is shown.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MonitorCalc

^

Symbol i

NiiFileExistQ[file] checks if the *.nii or *.nii.gz file exists.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`NiiFileExistQ

^

Symbol i

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 $Ax=y$.

output is the solution x.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[NNLeastSquares] = {ArgumentsPattern -> {_, _}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`NNLeastSquares`

^

Symbol i

PadToDimensions[data] pads the data to the max dimensions of data, using FindMaxDimensions.
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 i

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

Documentation [Local »](#)

Default Definitions `SyntaxInformation[QMRIToolsFuncPrint] = {ArgumentsPattern -> {_, _}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`QMRIToolsFuncPrint`

^

Symbol i

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

QMRIToolsFunctions["toolbox", p] gives a table of 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 i

QMRIToolsPackages[] give list of all the QMRITools packages.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`QMRIToolsPackages

^

Symbol i

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

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`QuaternionToRotationMatrix

^

Symbol i

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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`QuaternionVectorToRotationMatrix

^

Symbol i

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 i

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 i

ReverseDimensions[data] reverses the dimensions of the data.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`ReverseDimensions`

^

Symbol i

RMSNoZero[vec] return the RMS error of the vec which can be any dimension 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 i

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 i

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 i

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

Documentation [Local »](#)

Default Definitions `SyntaxInformation[RotationMatrixToQuaternion] = {ArgumentsPattern → {...}}`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`GeneralTools`RotationMatrixToQuaternion`

^

Symbol i

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

Documentation [Local »](#)

Default Definitions `SyntaxInformation[RotationMatrixToQuaternionVector] = {ArgumentsPattern → {...}}`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`GeneralTools`RotationMatrixToQuaternionVector`

^

Symbol i

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

SaveImage[image, "filename"] exports graph to image with "filename", 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 i

SignNoZero[val] gives the sign of the val, where the sign of $val > 0$ is 1 and $val < 0$ is -1.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[SignNoZero] = {ArgumentsPattern → {_}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`SignNoZero`

^

Symbol i

Squeeze[data] Removes the singleton dimensions from data.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`Squeeze`

^

Symbol i

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

Documentation [Local »](#)

Default Definitions `SyntaxInformation[StandardDeviationNoZero] = {ArgumentsPattern → {_}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`StandardDeviationNoZero`

^

Symbol i

`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 i

`StichData[datall,datarr]` 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 i

`StringPadInteger[num]` converts the integer num to a string and pads it with zeros to length 3.

`StringPadInteger[{num, len}]` converts the integer num to a string and pads it with zeros to length len.

`StringPadInteger[pre, num]` the same but with prefix pre.

`StringPadInteger[pre, {num, len}]` the same but with prefix pre.

`StringPadInteger[num, post]` the same but with postfix post.

`StringPadInteger[{num, len}, post]` the same but with postfix post.

`StringPadInteger[pre, num, post]` the same but with pre and postfix pre and post.

`StringPadInteger[post, {num, len}, post]` the same but with pre and postfix pre and post.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`StringPadInteger

^

Symbol i

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 i

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 i

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 i

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 i

AllowSelfDependencies is an option for `MakeFunctionGraph`. Can be `True` or `False`. If `True` a function that calls itself is also shown.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`AllowSelfDependencies`

^

Symbol i

CenterRange is an option for `MakeIntFunction`. If set `True` the centers of the dataset is the origin else its the corner.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`CenterRange`

^

Symbol i

CenterVoxel is an option for `MakeIntFunction`. If set `True` the centers of the voxels are interpolated else its the corners.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`CenterVoxel`

^

Symbol i

CropAlways is an option for `ApplyCrop`. If set `True` it will always crop even if outside the data.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GeneralTools`CropAlways`

^

Symbol i


CropInit is an option for `CropData`. By default the crop is not initialized but 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}

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 i

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 i

LabelPlacement is an option for MakeFunctionGraph. Defines where to place the label of the functiongraph. Accepts values that can be used in Placed.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`LabelPlacement

^

Symbol i

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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`OutputWeights

^

Symbol i

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 i


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 i

Bmatrix[bvec, grad] creates bmatrix form grad and bvec in form $\{-b_{xx}, -b_{yy}, -b_{zz}, -b_{xy}, -b_{xz}, -b_{yz}, 1\}$.

Bmatrix[{bvec, grad}] creates bmatrix form grad and bvec in form $\{b_{xx}, b_{yy}, b_{zz}, b_{xy}, b_{xz}, b_{yz}\}$.

Documentation [Local »](#)

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

Options `Method → DTI`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GradientTools`Bmatrix`

^

Symbol i

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 i

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 i

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 i

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

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GradientTools`BmatrixRot`

^

Symbol i

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 i

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 i

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

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GradientTools`ConditionNumberCalc`

^

Symbol i

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 i

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 i

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 i

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 i

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 i

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 i

FullGrad is an option for Grad. Default is True. When true the gradient directions will 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 must be an integer.

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

GenerateGradients[{numb1, numb2 ...}, alpha] optimizes a multi shell gradient set with numb gradients per shell. If alpha is set to 0.5 equal importance is given to the optimal distribution of each shell in the entire set. If alpha is 0 only the sub shells will be optimized, if alpha is set to 1 only the global set will 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 i

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 i

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 i

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, StepSize → 0.025}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GradientTools`GradBmatrix`

^

Symbol i

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 i

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 i

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 i

SelectBvalueData[{data, vals}, sel] selects the volumes and bvalues based on sel. The value of sel can be a number, for which al higher bvalues are selected, or a list {min, max}.

SelectBvalueData[{data, vals, grads}, sel] the same but also including the gradients.

Output is {data, vals} or {data, vals, grads}.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`GradientTools`SelectBvalueData`

^

Symbol i

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. Default 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 flipped.

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 will be produced, "Normal" or "OverPlus".

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`GradType

^

Symbol i

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 i

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

It specifies which registration method to use.

Methods can be "translation", "rigid", "affine", "bspline", "rigidDTI", "affineDTI", "PCAttranslation", "PCArigid", "PCAaffine", or "PCAbspline".

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ElastixTools`MethodReg

^

Symbol i

OrderSpan is an options for FindOrder.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`OrderSpan

^

Symbol i

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 i


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 


StepSize is an option for GradBmatrix. Specifies the integration stepsize is Method -> "Numerical" is used.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GradientTools`StepSize

^

Symbol 

SwitchAxes is an option for GradSeq. Default 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 i

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 i

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 i

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

^

IVIMTools

Functions

Symbol i

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 → {20000, 1000, 10}... (7 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`BayesianIVIMFit2

^

Symbol i

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 → {20000, 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 parameter maps.


Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`IVIMTools`CorrectParMap`

^

Symbol 

FConvert[f] converts 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`

^

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 i

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 i

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 and bvals using Bmatrix with the bvalues in s/mm².

init should be the initialization parameters for 2 components this is {s0, f, D, Dp} for 3 components this is {s0, f1, f2, D, Dp1, Dp2}.

The fraction is defined between 0 and 1, the D, Dp, Dp1 and Dp2 is in mm²/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 i

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 i

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 i

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

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`IVIMResiduals

^

Symbol i

MeanBvalueSignal[data, bval] calculates the geometric mean of the data for each unique bval.

output is the mean data and the unique bvalues.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`MeanBvalueSignal

^

Symbol i

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 i

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 i

ChainSteps is an option for BayesianIVIMFit2 and BayesianIVIMFit3. It determines how long the algorithm runs. three values must be given {iterations, burn steps, sample density}.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`ChainSteps

^

Symbol i

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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`CorrectPar

^

Symbol i

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 i

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 i


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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`IVIMTools`FilterType


^

Symbol 

FitConstrains is an option for BayesianIVIMFit2 and BayesianIVIMFit3. Gives the constraints 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 will 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 will 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 i

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 i

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

Definitions

```
Parallelize[Parallel`Kernels`Private`args$____] := (Parallel`Protected`doAutolaunch[TrueQ[Parallel`Static`$enableLaunchFeedback]]);
Parallelize[Parallel`Kernels`Private`args$])
```

Documentation [Local »](#) | [Web »](#)

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

Attributes {HoldFirst, Protected}

Full Name System`Parallelize

^

Symbol i

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 i

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 i

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 i

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

SequencePulseAcquire[din, ham, b1] performs a pulsaquire experiment of the spin system din given the hamiltonian ham 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 i

SequenceSpaceEcho[din, ham, 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 between 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 ham 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 i

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

SequenceSpinEcho[din, ham, te, b1] performs a spin echo experiment with echo time te of the spin system din given the hamiltonian ham 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 i

SequenceSteam[din, ham, {te, tm}] performs a stimulated echo experiment with echo time te and mixing time tm of the spin system din given the hamiltonian ham 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 i

SequenceTSE[din, ham, {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 ham using ex Degree excitation and ref Degree refocus pulses.

SequenceTSE[din, ham, {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 ham using ex Degree excitation 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 i

SimAddPhase[din,ham,phase] adds phase to the spin system din given the hamiltonian ham.
din and ham 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 i

SimEvolve[din,ham,t] evolves the spin system din given the hamiltonian ham over a time t. din and ham 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 i

SimHamiltonian[sysi] simulates the hamiltonian 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 i

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

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 i

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

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

din and ham 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 i

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

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 i

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 i

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

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`JcouplingTools`ReadoutBandwith

^

Symbol i

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 i

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 i


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

^

LoggingTools

Functions

Symbol 

AddToLog[list] add the list to the log at level 1. All elements of the list are converted to strings and joined with spaces.

AddToLog[list, level] add the list to the log at level.

AddToLog[list, True] add the list to the log at level 1 with a timestamp.

AddToLog[list, True, level] specifies both the level and the timestamp.

AddToLog[list, level, True] specifies both the level and the timestamp.


Documentation [Local »](#)

Default Definitions `SyntaxInformation[AddToLog] = {ArgumentsPattern -> {{_, _, _}}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`LoggingTools`AddToLog

^

Symbol 

CheckFile[{fol, name, tag}, stat, ver] checks a checkfile for status stat and software version.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[CheckFile] = {ArgumentsPattern -> {{_, _, _}, _, _}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`LoggingTools`CheckFile

^

Symbol i

DirectoryTree[dir] makes a directory tree from the dir.

Documentation [Local](#) »

Default Definitions `SyntaxInformation[DirectoryTree] = {ArgumentsPattern → {}}`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`LoggingTools`DirectoryTree`

^

Symbol i

ExportDirectoryTree[fol] make a DirectoryTree for the folder and exports it there as FileTree.txt.

ExportDirectoryTree[fol,name] make a DirectoryTree for the folder and exports it there as name .txt.

Documentation [Local](#) »

Default Definitions `SyntaxInformation[ExportDirectoryTree] = {ArgumentsPattern → {}}`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`LoggingTools`ExportDirectoryTree`

^

Symbol i

ExportLog[file] exports the log as a plain text to file.

Documentation [Local](#) »

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`LoggingTools`ExportLog`

^

Symbol i

ImportLog[file] imports the log as a list of string from a plain text file.

Documentation [Local](#) »

Default Definitions `SyntaxInformation[ImportLog] = {ArgumentsPattern → {}}`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`LoggingTools`ImportLog`

^

Symbol i

MakeCheckFile[{fol, name, tag}, stat] makes a check json file in the fol with filename containing name and tag. Where stat is a list of rules that contain any information that needs to be put in the check file.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[MakeCheckFile] = {ArgumentsPattern -> {{_, _, _}, _}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`LoggingTools`MakeCheckFile

^

Symbol i

PrintDirectoryTree[tree] prints the directory tree generated by DirectoryTree.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[PrintDirectoryTree] = {ArgumentsPattern -> {_, _}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`LoggingTools`PrintDirectoryTree

^

Symbol i

ResetLog[] restes the log to {}.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[ResetLog] = {ArgumentsPattern -> {}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`LoggingTools`ResetLog

^

Symbol i

ShowLog[] shows the log in a popup window.

ShowLog[False] shows the log in the notebook.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[ShowLog] = {ArgumentsPattern -> {_, _}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`LoggingTools`ShowLog

^

Options

MaskingTools

Functions

Symbol i

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

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`MaskingTools`DilateMask`

^

Symbol i

FitGradientMap[data, ord] fit of gradient trough all non zero values withing the data.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`MaskingTools`FitGradientMap`

^

Symbol i

GetCommonSegmentation[dat, seg, vox] For a list of multiple datasets dat the common segmentations from the list seg are determined.

Output is a list of segmentations where for each region only the part present in all datasets is selected.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`MaskingTools`GetCommonSegmentation`

^

Symbol i

GetSegmentationLabels[segmentation] gives a list of all labels in the segmentation.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name `QMRITools`MaskingTools`GetSegmentationLabels`

^

Symbol i

HomogenizeData[data, mask] tries to homogenize the data within the mask by removing intensity gradients.

Documentation [Local »](#)

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

Options FitOrder → 5

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`HomogenizeData

^

Symbol i

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 ... (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`Mask

^

Symbol i

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 i

MaskSegmentation[seg, mask] applies a mask to a splitted segmentation seg from SplitSegmentations.

The mask is 3D, seg is 4D.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MaskSegmentation

^

Symbol i

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

Output is a labeled segmentation.

MergeSegmentations[masks] does the same but automatically numbers the segmentations.

Documentation [Local](#) »

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MaskingTools`MergeSegmentations`

^

Symbol i

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, mask] normalizes the data based on the mean signal only within the mask.

Documentation [Local](#) »

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

Options `NormalizeMethod → Set`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MaskingTools`NormalizeData`

^

Symbol i

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

Documentation [Local](#) »

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

Options `NormalizeMethod → Set`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MaskingTools`NormalizeMeanData`

^

Symbol i

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 i

ReplaceSegmentations[seg, labs, new] relapaces the labels labs form the segmentation seg for labels new. Both labs and new should be lists of integers of the same size. If seg contains more labels then given in labs these will be replaced by 0.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[ReplaceSegmentations] = {ArgumentsPattern -> {_, _, _}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`ReplaceSegmentations

^

Symbol i

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 i

SegmentationVolume[seg] calculates the volume of each label in the segmentation.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[SegmentationVolume] = {ArgumentsPattern -> {_, _}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SegmentationVolume

^

Symbol i

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

SelectMaskComponents[mask] selects the largest connected component in the mask.

SelectMaskComponents[mask,n] selects the n largest connected components in the mask.

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SelectMaskComponents

^

Symbol i

SelectSegmentations[seg, labs] selects only the segmentaions from seg with label number labs.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SelectSegmentations

^

Symbol i

SmoothMask[mask] generates one clean masked volume form a noisy mask.

Documentation [Local »](#)

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

Options ▶ MaskComponents → 1 ... (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SmoothMask

^

Symbol i

SmoothSegmentation[segmentation] smooths segmentations and removes the overlaps between multiple segmentations.

SmoothSegmentation[segmentation, labs] only smooths the selected label number labs.

Documentation [Local »](#)

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

Options ▶ MaskComponents → 1 ... (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SmoothSegmentation

^

Symbol i

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 i

FitOrder is an option for HomogenizeData. It specifies the order of harmonics to be used for the homogenization.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MaskingTools`FitOrder`

^

Symbol i

MaskClosing is an option for Mask, SmoothMask and SmoothSegmentation. The size of the holes in the mask that will be closed.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MaskingTools`MaskClosing`

^

Symbol i

MaskComponents is an option for Mask, SmoothMask and SmoothSegmentation. Determinse the amount of largest clusters used as mask.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MaskingTools`MaskComponents`

^

Symbol i

MaskDilation is an option for Mask, SmoothMask and SmoothSegmentation. 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 i

MaskFiltKernel is an option for Mask, SmoothMask and SmoothSegmentation. How much the contours are smoothed.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`MaskFiltKernel

^

Symbol i

MaskSmoothing is an options for Mask, SmoothMask and SmoothSegmentation, 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 i

NormalizeMethod is an option for NormalizeData. Can be "Set" or "Volumes" wich normalizes to the first volume or normalizes each volume individually, respectively. If "Uniform" normalizes the histogram of the data to have a uniform distribution between 0 and 1 where 0 is treated as background of the data.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`NormalizeMethod

^

Symbol i

SmoothItterations is an option for Mask, SmoothMask and SmoothSegmentation and defines how often the smoothing is repeated.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MaskingTools`SmoothItterations

^

MuscleBidsTools

Functions

Symbol i

AddToJson[json, <[key->value..]>] adds new keys and values to the JSON list where duplicate keys are either removed or joined. AddToJson[json, "QMRITools"] adds the QMRITools software version to the json.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`AddToJson

^

Symbol i

BidsDcmToNii[dir] converts the bids sourceFolder with dicom files to raw nii files based on the config file.

Documentation [Local »](#)

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

Options `{BidsIncludeSession → True, SelectSubjects → All}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`BidsDcmToNii`

^

Symbol i

CheckDataDiscription[description] checks the data description config file used in BidsDcmToNii, MuscleBidsConvert, MuscleBidsProcess, and MuscleBidsMerge.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`CheckDataDiscription`

^

Symbol i

ExtractFromJSON[keys] if the keys exist they are extracted from the json.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`ExtractFromJSON`

^

Symbol i

GenerateBidsFileName[parts] generates a Bids file name from the Bids labels association which can be generated by PartitionBidsName.

GenerateBidsFileName[fol, parts] the same but with a custom root folder.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`GenerateBidsFileName`

^

Symbol i

GenerateBidsFolderName[parts] generates a Bids folder name from the Bids labels association which can be generated by PartitionBidsFolderName.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`GenerateBidsFolderName`

^

Symbol i

GenerateBidsName[parts] generates a Bids file name from the Bids labels association which can be generated by PartitionBidsName.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`GenerateBidsName`

^

Symbol i

GetConfig[folder] Imports a Muscle Bids config file from the given folder.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`GetConfig`

^

Symbol i

GetJSONPosition[{json..}, {{key, value}..}] gets the position from a list of JSON association lists where keys have the given value.

GetJSONPosition[{json..}, {{key, value}..}, sortkey] same but finally sorts the positions for the value of the sortkey.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`GetJSONPosition`

^

Symbol i

ImportJSON[file] imports a json file as rawJSON.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`ImportJSON`

^

Symbol i

MergeJSON[[json..]] merges a list of JSON association lists where duplicate keys with the same values are removed and duplicate keys with different values are merged.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`MergeJSON

^

Symbol i

MuscleBidsConvert[dir] converts all raw nii data in the to Muscle-Bids named nii based on the config file in the bids sourceFolder dir.

Documentation [Local »](#)

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

Options {DeleteAfterConversion → True, SelectSubjects → All}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`MuscleBidsConvert

^

Symbol i

MuscleBidsMerge[dir] merges multiple stack data for all Muscle-Bids named nii based on the config file in the bids sourceFolder dir.

Documentation [Local »](#)

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

Options {SelectSubjects → All, VersionCheck → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`MuscleBidsMerge

^

Symbol i

MuscleBidsProcess[dir] processes all Muscle-Bids named nii based on the config file in the bids sourceFolder dir.

Documentation [Local »](#)

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

Options {SelectSubjects → All, VersionCheck → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`MuscleBidsProcess

^

Symbol i

MuscleBidsSegment[dir] segments the data of Muscle–Bids named nii based on the config file in the bids sourceFolder dir. The segmentations are generated by the function SegmentData.

Documentation [Local](#) »

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

Options {SelectSubjects → All, VersionCheck → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`MuscleBidsSegment

^

Symbol i

MuscleBidsTractography[dir] performs tractography on the Muscle–Bids named nii based on the config file in the bids sourceFolder dir. If a segmentation is present it is used as a mask for per muscle segmentation.

Documentation [Local](#) »

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

Options {SelectSubjects → All, VersionCheck → False, BidsTractographyMethod → Full}

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`MuscleBidsTractography

^

Symbol i

PartitionBidsFolderName[fol] partitions the Bids folder and file name. It returns the bids root folder and the label parts using PartitionBidsName.

Documentation [Local](#) »

Default Definitions `SyntaxInformation[PartitionBidsFolderName] = {ArgumentsPattern → {_}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`PartitionBidsFolderName

^

Symbol i

PartitionBidsName[name] converts a Bids name to the Bids labels as an association, i.e. {"sub", "ses", "stk", "rep", "type", "suf"}.

Documentation [Local](#) »

Default Definitions `SyntaxInformation[PartitionBidsName] = {ArgumentsPattern → {_}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`PartitionBidsName

^

Symbol i

SelectBidsFolders[fol, tag] Selects all folders in the fol with the name tag.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`SelectBidsFolders`

^

Symbol i

SelectBidsSessions[fol] selects all sessions in the bids subject folder.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`SelectBidsSessions`

^

Symbol i

SelectBidsSubjects[fol] selects all subjects in the bids folder.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`SelectBidsSubjects`

^

Symbol i

ViewConfig[config] shows a config file for Muscle Bids processing.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`ViewConfig`

^

Options

Symbol i

BidsIncludeSession is an option for BidsDcmToNii. If True session folders will be used in output even if they are not specified.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`MuscleBidsTools`BidsIncludeSession`

^

Symbol

BidsTractographyMethod is an option for MuscleBidsTractography and can be "Full", "Tractography" or "Segmentation".

With Tractography only the tractography is performed without segmentation.

With Segmentation only the segmentation is performed without tractography. With Full both are performed.

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`BidsTractographyMethod

^

Symbol i

DeleteAfterConversion is an option for MuscleBidsConvert. If set True all files that have been converted will be deleted.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`DeleteAfterConversion

^

Symbol i

SelectSubjects is an option for Bids functions. Can be a list of bids subject names else it is All.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`SelectSubjects

^

Symbol i

VersionCheck is an option for all Bids functions. If set True data processed with an old version is reprocessed.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`MuscleBidsTools`VersionCheck

^

NiftiTools

Functions

Symbol i

CompressNiiFiles[] prompts 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 i

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 i

DcmToNii[] converts a dicom folder to nii, you will be promoted for the location of the folders.

DcmToNii[{"input","output"}] 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 application folder.

Documentation [Local »](#)

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

Options ▸ `CompressNii → True ... (6 total)`

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`DcmToNii

^

Symbol i

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 i

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 i

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 → {_, ...}, OptionsPattern[]}`

Options {FlipBvec → False, PositiveZ → False}

Attributes {Protected, ReadProtected}

Full Name `QMRITools`NiftiTools`ExportBvec`

^

Symbol i

ExportNii[data, vox] exports the nii file and will prompt 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 i

ExtractNiiFiles[] prompts 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 i

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 i

ImportBmat[] will prompt 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 i

ImportBval[] will prompt 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 i

ImportBvalvec[] will prompt 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, PositiveZ → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportBvalvec

^

Symbol i

ImportBvec[] will prompt 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 i

ImportExploreDTItens["file"] imports the *.nii export for the tensor from explore DTI.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`ImportExploreDTItens

^

Symbol i

ImportNii[] prompts 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 supported 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 i

ImportNiiDiff[] will prompt for the *.nii, *.bvec and *.bval file to import.

ImportNiiDiff[*nii] will import the *.nii file and automatically also imports the *.bvec and *.bval if 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 i

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 i

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 i

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 i


MakeNiiOrentationQ[rot] makes the q vector from rotation matrix rot.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`MakeNiiOrentationQ

^

Symbol 

MakeNiiOrentationS[off, vox] maxes the srow values for nii header assuming not rot and Q.

MakeNiiOrentationS[off, vox, rot] maxes the srow values for nii header using rotation rot.

MakeNiiOrentationS[off, vox, rot, Q] maxes the srow values for nii header using rotation rot 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 


MonitorCalc is an option for many processing functions. When true the proceses of the calculation is shown.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MonitorCalc

^

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 i

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 i

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 i

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 i

RotateGradients is an option for ImportNiiDiff.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`NiftiTools`RotateGradients

^

Symbol i

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 i

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

^

PlottingTools

Functions

Symbol i

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`PlottingTools`ColorFAPlot

^

Symbol

GenerateRotationFrames[3Dgraphics, nFrames] generates nFrames of the 3D graphics rotating around the z-axis.

Options ImageSize → 500

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`GenerateRotationFrames

^

Symbol i

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 i

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 i

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

Link3DGraphic[3Dgraphics] creates a 3D graphic with a linked view options between all 3D graphics that have been linked.

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`PlottingTools`Link3DGraphic`

^

Symbol i

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 i

MakeSlicelImages[imgData] generates images from the imgData which is obtained form GetSliceData.

MakeSlicelImages[imgData, vox] generates images from the imgData which is obtained form GetSliceData, vox is used for the correct aspect ratio of the images.

MakeSlicelImages[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).

MakeSlicelImages[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[MakeSlicelImages] = {ArgumentsPattern → {_, _, _}, OptionsPattern[]}`

Options ► `PlotRange → Automatic ...` (6 total)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`PlottingTools`MakeSlicelImages`

^

Symbol i

PlotContour[data, vox] creates a contour of the data.

The data can be 3D or 4D, when its 4D each contour is given a random color.

Documentation [Local »](#)

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

Options ► `ContourColor → ■ ...` (7 total)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`PlottingTools`PlotContour`

^

Symbol i

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 i

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 i

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 i

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 i

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 i

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 i

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 i

PlotSegmentations[seg, vox] crates a contour of each of the segementations in seg. Uses PlotContour.

PlotSegmentations[seg, bone, vox] same but also generates an extra contour with the bones in gray.

Documentation [Local »](#)

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

Options » ImageSize → 400 ... (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotSegmentations

^

Symbol i

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



ClippingStyle is an option for plotting functions that specifies the style of what should be drawn when curves or surfaces would extend beyond the plot range.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected}

Full Name System`ClippingStyle



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



ContourColor is an option for PlotContour. It specifies the contour color and can be a single color or an array with the same dimensions as the data.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ContourColor



Symbol



ContourColorRange is an option for PlotContour. If the ContourColor is an array this values gives the plot range.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ContourColorRange



Symbol




ContourOpacity is an option for PlotContour and PlotSegmentations. It specifies the opacity of the contour.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ContourOpacity



Symbol 


ContourResolution is an option for PlotContour. It defines the mesh resolution used, can be a singel number or a list of 3 numbers.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ContourResolution

^

Symbol 


ContourScaling is an option for PlotCountour. The value can be "World" or "Voxel", if the value is "Wold" the segmentation is in mm else in voxel coordinates.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ContourScaling

^

Symbol 


ContourSmoothRadius is and option for PlotContour. It defines the smoothing radius with an integer, None or 0 does not smooth.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ContourSmoothRadius

^

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 MakeSlicelImages, if set true a barlegend is added to the image.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ImageLegend

^

Symbol i

ImageOrientation is an option for MakeSlicImages. Can be Automatic, "Vertical" or "Horizontal".

Documentation [Local](#) »

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`ImageOrientation

^

Symbol i

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 i

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 i

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 i

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 i

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 i

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 i

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 i

PlotSpace is an option for GradientPlot can be "bspace" or "qspace".

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`PlotSpace

^

Symbol i

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 i

RandomizeColor is an option for PlotSegmentations. If True the colors are randomly assigned to each segmentation.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`RandomizeColor

^

Symbol i

SphereColor ListSpherePlot. 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 i

SphereSize is an option for GradientPlot and ListSpherePlot. Sets the size of the spheres that represent the gradients.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`PlottingTools`SphereSize

^

ProcessingTools

Functions

Symbol i

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 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 i

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 i

CombineB1[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 i

CorrectJoinSetMotion[[{dat1, dat2, ...}, vox, over] motion corrects 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, MonitorCalc → True}

Attributes {Protected, ReadProtected}

Full Name `QMRITools`ProcessingTools`CorrectJoinSetMotion`

^

Symbol i

DataTransformation[data, vox, w] transforms a 3D dataset according to the affine transformation vector w.

Documentation [Local](#) »

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

Options {InterpolationOrder → 1, PadOutputDimensions → False}

Attributes {Protected, ReadProtected}

Full Name `QMRITools`ProcessingTools`DataTransformation`

^

Symbol i

DatTot[[data1, data2, ..], name, vox] calculates the parameter table containing 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 i

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 i

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 i

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 i

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 i

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`ProcessingTools`GetMaskData

^

Symbol i

GetMaskMeans[dat, mask] calculates the mean, std, 5,50 and 95% CI form the given data for each of the given masks.

Mask can be genereated by SegmentTracts.

GetMaskMeans[dat, mask, name] where name is a string that is added to the header.

Documentation [Local »](#)

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

Options MeanMethod → SkewNormalDist

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`GetMaskMeans

^

Symbol i

GetTractMeans[dat, tracts, vox] calculates the mean, std, 5,50 and 95% CI form the given data for each of the given tracts.

Tracts can be by SplitSegmentations. name is a string that is added to the header.

GetTractMeans[dat, tracts, vox, name] where name is a string that is added to the header.

Documentation [Local »](#)

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

Options {MeanMethod → SkewNormalDist, InterpolationOrder → 0}

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`GetTractMeans

^

Symbol i

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 i

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 i

InvertDataset[data] inverts the data along the x y and z axes. In other words it is rotated around 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 i

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 → False ... (8 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`JoinSets

^

Symbol i

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 i

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`ProcessingTools`MeanSignal`

^

Symbol i

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 i

MedCouple[data] calculates the medcouple of a list of data.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`ProcessingTools`MedCouple`

^

Symbol i

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 i

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 i

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 i

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 i

RotateTensor[tens] rotates the tensor 180 degree, e.g. inversion of the z direction with also inverting the tensor orientation.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[RotateTensor] = {ArgumentsPattern → {_}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`ProcessingTools`RotateTensor`

^

Symbol i

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 i

SNRCalc[data, sigma] calculates the ANR of the data using a noise sigma map for example generated by PCADeNoise.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`SNRCalc

^

Symbol i

SNRMapCalc[data1, noisemap] calualtes the signal to noise ratio of the data using $MN[data]/(1/\sqrt{\pi/2} \text{ sigma})$, where sigma is the local mean of the noise map assuming it is a rician distribution.

SNRMapCalc[{data1, data2}] calualtes the signal to noise ratio from two identical images using $MN[data1, data2] / (.5 \text{ SQRT}[2] \text{ STDV}[data2 - data1])$.

SNRMapCalc[{data1, .. dataN}] calualtes 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 i

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 i

AxesLabel is an option for graphics functions that specifies labels for axes.

Documentation [Local »](#) | [Web »](#)

Attributes `{Protected}`

Full Name `System`AxesLabel`

^

Symbol i

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 i

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 i


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" assumes 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 will 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 


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`ProcessingTools`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`ProcessingTools`GetMaskOutput

^

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 i

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 i

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 i

MaskCompartment is an option for SmartMask. Can be "Muscle" or "Fat".

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`MaskCompartment

^

Symbol i

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 i


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 many processing functions. When true the proceses of the calculation is shown.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MonitorCalc

^

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 iteration the outliers are reevaluated on the data with the previously found outliers already 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". "IQR" 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 i

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

PadOutputDimensions is an options for `DataTransformation`. If False the data is the same dimensions as the input else the data is padded.

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`PadOutputDimensions

^

Symbol i

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 i

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 i

ReverseData is an option for `JoinSets`. Reverses each individual dataset given as input for the `JoinSets` function. True by default.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`ReverseData

^

Symbol i


ReverseSets is an option for `JoinSets`. Reverses the order of the datasets, 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 


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 i

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

^

Symbol i

UseMask is a function for MeanSignal and DriftCorrect.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`UseMask

^

ReconstructionTools

Functions

Symbol i

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 sensitivities.

Possible coil combination methods are "Sum", "RootSumSquares", "RoemerEqualNoise", "RoemerEqualSignal", "WSVD".

RootSumSquares needs the signal. Can be performed with and without the noise covariance matrix

RoemerEqualNoise needs the signal and the noise covariance matrix. Can be performed with and without the sense data, without sense data the sensitivity is estimated using the signal and the RSS reconstruction of the signal.

RoemerEqualSignal needs the signal and the noise covariance 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 i

CoilWeightedRecon[kspace, noise, head] performs reconstruction 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 »](#)

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

Options ► EchoShiftData → 0 ... (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`CoilWeightedRecon

^

Symbol i

CoilWeightedReconCSI[kspace, noise, head] performs reconstruction 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 »](#)

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

Options ► HammingFilter → False ... (5 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`CoilWeightedReconCSI

^

Symbol i

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 i

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] applies 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] applies a Hammingfilter to the data. The data is in image 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 divided by the RSS reconstruction of the coils.

Documentation [Local »](#)

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

Options SenseRescale → True

Attributes {Protected, ReadProtected}

Full Name QMRITools`ReconstructionTools`MakeSense



Symbol i

MeanType[kspace, types, type] calculates 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] calculates the Mean of the kspace data on type, where type is part of types.

MeanType[kspace, types, {type,...}] calculates 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 i

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 i

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 i

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 i

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 i

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 i

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 i

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 i

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 i

TotalType[kspace, types, type] calculates 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] calculates the Total of the kspace data on type, where type is part of types.

TotalType[kspace, types, {type,...}] calculates 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 i

AcquisitionMethod is an option for CoilWeightedReconCSI. Values can be "Fid" or "Echo".

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`ReconstructionTools`AcquisitionMethod`

^

Symbol i


CoilSamples is an option for CoilWeightedReconCSI and specifies how many fud samples are used to calculate 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 


ReconFilter is an option for CoilWeighted recon. If true the reconstruction gets a hamming filter.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools'ReconstructionTools'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 i

CalibrateEPGT2Fit[datan, times, angle] calculates the Fat t2 relaxation 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 i

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} ... (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`RelaxometryTools`CreateT2Dictionary

^

Symbol i

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 i

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 i

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 methods.

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. 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), 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, muscleFrantion, 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. Specifies 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 i

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 i

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 i

MonitorCalc is an option for many processing functions. When true the proceses of the calculation is shown.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MonitorCalc

^

Symbol i

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 i

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

ScientificColorData

Functions

Symbol
AddScientificColors[dir] adds the scientific colour data (https://zenodo.org/records/8409685) from the specified folder.
Documentation Local » Attributes {Protected, ReadProtected} Full Name QMRITools`ScientificColorData`AddScientificColors

Symbol
ExtractColorData[] Extracts the scientific colordata archive (https://zenodo.org/records/8409685).
Documentation Local » Attributes {Protected, ReadProtected} Full Name QMRITools`ScientificColorData`ExtractColorData

Options

SegmentationTools

Functions

Symbol
AddLossLayer[net] adds three loss layers to a NetGraph, a DiceLossLayer, JaccardLossLayer, TverskyLossLayer, MeanSquaredLossLayer and a CrossEntropyLossLayer are added.
Documentation Local » Default Definitions SyntaxInformation[AddLossLayer] = {ArgumentsPattern → {..}} Attributes {Protected, ReadProtected} Full Name QMRITools`SegmentationTools`AddLossLayer

Symbol i

AnalyseNetworkFeatures[net, data] gives overview of the information density of the network features by analysing them with SVD.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`AnalyseNetworkFeatures

^

Symbol i

ApplySegmentationNetwork[data, net] segments the data using the pretrained net.

Documentation [Local »](#)

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

Options ▶ TargetDevice → GPU ... (4 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`ApplySegmentationNetwork

^

Symbol i

AugmentImageData[image, {rotate, flip}] augments the input image by rotating between -180 and 180 degrees and flipping. The inputs rotate and flip can be set to True or False.

AugmentImageData[{image, ..}, {rotate, flip}] same but for a list of images.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`AugmentImageData

^

Symbol i

AugmentTrainingData[{data, segmentation}, vox] augments the data and segmentation in the same way.

AugmentTrainingData[{data, segmentation}, vox, aug] by setting aug to True or False the augmentation can be turned on or off.

The value aug can also be a list of boolean values controlling various augmentation parameters {flip, rotate, translate, scale, noise, blur, brightness}.

The default settings are {True, True, True, True, False, False, False}.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`AugmentTrainingData

^

Symbol i

ChangeNetDimensions[netIn] changes input channels, output classes, the input patch dimension of the input network netIn.

Documentation [Local »](#)

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

Options {Dimensions → None, Channels → None, Classes → None}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`ChangeNetDimensions

^

Symbol

CheckSegmentation[seg] checks the segmentation for errors and returns a vector of two numbers, the first indicates if the segmentation has more than one region, the second indicates if it has holes.

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`CheckSegmentation

^

Symbol i

ClassDecoder[probability] decodes a probability vector of 1 and 0 into Integers of 0 to the value of the last dimension of probability minus one.

ClassDecoder[probability, nClass] decodes a probability vector of 1 and 0 into Integers of 0 to nClass - 1.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`ClassDecoder

^

Symbol i

ClassEncoder[label] encodes Integer label data of 0 to max value of label into a nClass + 1 vector of 1 and 0 as the last dimension.

ClassEncoder[label, nClass] encodes Integer label data of 0 to nClass into a nClass + 1 vector of 1 and 0 as the last dimension.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`ClassEncoder

^

Symbol i

ClassifyData[data, method] classifies the input data using the given method. The data is converted to images using MakeClassifyImages.

The input method can be a filename of a classify network or a classify network.

Additionally the input method can be one of the predefined methods "LegPosition" or "LegSide".

Documentation [Local »](#)

Default Definitions `SyntaxInformation[ClassifyData] = {ArgumentsPattern -> {_, _}, OptionsPattern[]}`

Options `TargetDevice -> GPU`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`ClassifyData`

^

Symbol i

DataToPatches[data, patchSize] creates the maximal number of patches with patchSize from data, where the patches have minimal overlap.

DataToPatches[data, patchSize, n] gives n random patches from the maximal number of patches with patchSize from data, where the patches have minimal overlap.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[DataToPatches] = {ArgumentsPattern -> {_, _}, OptionsPattern[]}`

Options `{PatchNumber -> 0, PatchPadding -> 0}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`DataToPatches`

^

Symbol i

DiceLossLayer[] represents a net layer that computes the Dice loss by comparing input class probability vectors with the target class vector.

DiceLossLayer[n] does the same but n defines the power of the denominator, with n=2 the squared dice score, is calculated.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[DiceLossLayer] = {ArgumentsPattern -> {_, _}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`DiceLossLayer`

^

Symbol



DiceSimilarity[ref, pred] gives the Dice Similarity between 1 and 0 of segmentations ref and pred for class equals 1.

DiceSimilarity[x, y, class] gives the Dice Similarity of segmentations ref and pred for class.

DiceSimilarity[x, y, {class, ..}] gives the Dice Similarity of segmentations ref and pred for the list of gives classes.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`DiceSimilarity



Symbol

FocalLossLayer[] represents a net layer that computes the Focal loss by comparing input class probability vectors with the target class vector.

FocalLossLayer[g] does the same but uses g as the tunable focusing parameter gamma which needs to be larger than one.

FocalLossLayer[g, a] does the same but uses as the balancing factor alpha.

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`FocalLossLayer



Symbol



GetNeuralNet[name] loads a pretrained neural net that come with the toolbox. Current named nets are "LegSide", "LegSide", "SegThighMuscle", "SegLegMuscle", and "SegLegBones". The loading is cached within a session.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`GetNeuralNet



Symbol i

GetTrainData[data, batchsize, patch] creates a training batch of size batchsize with patchsize patch.

The input data can be out of memory in the form of a list of "*.wxf" files that contain the data, segmentation and voxel size or a list of "*.nii" files in the form {"data.nii", "segmentation.nii"}..}. The input data can be in memory in a list in the form {{data, segmentation, vox}..}

GetTrainData[data, batchsize, patch, nClass] If nClass is set to an value n > 0 the segmentations are decoded in n classes.

Documentation [Local »](#)

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

Options {PatchesPerSet → 1, AugmentData → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`GetTrainData

^

Symbol i

ImportITKLabels[file] imports the ITKSnap label file.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`ImportITKLabels

^

Symbol i

JaccardLossLayer[] represents a net layer that computes the Jaccard loss by comparing input class probability vectors with the target class vector.

JaccardLossLayer[n] does the same but n defines the power of the denominator, with n=2 the squared jaccard score is calculated.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`JaccardLossLayer

^

Symbol i

JaccardSimilarity[ref, pred] gives the Jaccard Similarity between 1 and 0 of segmentations ref and pred for class equals 1.

JaccardSimilarity[x, y, class] gives the Jaccard Similarity of segmentations ref and pred for class.

JaccardSimilarity[x, y, {class, ..}] gives the Jaccard Similarity of segmentations ref and pred for the list of gives classes.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`SegmentationTools`JaccardSimilarity`

^

Symbol

MakeChannelClassGrid[data, label] makes a 3 x 3 grid of crosssectional images of the channels data overlaid with a crosssectional image of the classes label of a training dataset generated

MakeChannelClassGrid[data, label, n] makes a n x n.

MakeChannelClassGrid[data, label, {n, m}] makes a n x m.

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`SegmentationTools`MakeChannelClassGrid`

^

Symbol i

MakeChannelClassImage[data, label] makes a crosssectional image of the channels data overlaid with a crosssectional image of the classes label of a training dataset generated

MakeChannelClassImage[data, label, {off,max}] same but with explicit definition of background value b and number of classes n.

MakeChannelClassImage[data, label, vox] same but with the aspect ratio determined by vox.

MakeChannelClassImage[data, label, {off,max}, vox] same with explicit definition and aspect ratio definition.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`SegmentationTools`MakeChannelClassImage`

^

Symbol i

MakeChannellImage[data] makes a crossectional image of the channels data of a training dataset generated by GetTrainData.

MakeChannellImage[data, vox] same but with the aspect ratio determined by vox.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`MakeChannellImage`

^

Symbol i

MakeClassifyImage[data] makes a image of the input data. The data is automatically cropped to remove the background and normalized.

If the input data is 3D a list of images is returned.

Documentation [Local »](#)

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

Options `ImageSize → {128, 128}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`MakeClassifyImage`

^

Symbol i

MakeClassifyNetwork[classes] makes a classify network with three convolution layers and 3 fully connected layers.

The input classes should be a list of strings. The input image dimensions should not be smaller than 64x64.

Documentation [Local »](#)

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

Options `ImageSize → {128, 128}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`MakeClassifyNetwork`

^

Symbol i

MakeClassImage[label] makes a crosssectional image of the classes label of a training dataset generated by GetTrainData

MakeChannellImage[label, {b, n}] same but with explicit definition of background value b and number of classes n.

MakeClassImage[data, vox] same but with the aspect ratio determined by vox.

MakeChannellImage[label, {b, n}, vox] same with explicit definition and aspect ratio definition.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`MakeClassImage

^

Symbol

MakeDistanceMap[mask] makes a distance map of the given mask in voxels. The distance map is negative inside the mask and positive outside the mask.

MakeDistanceMap[mask, vox] makes a distance map of the given mask in the same unit as vox. The distance map is negative inside the mask and positive outside the mask.

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

Options DistanceRange → Automatic

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`MakeDistanceMap

^

Symbol i

MakeNode[scale, conn, blockConfig] makes a node for a UNET.

The input scale defines the input and output scaling, is either an integer or a vector of lenth dim.

The input conn defines the connections, is a list of two integer values defining the number of input and output ports.

The blockConfig is defined as {{blockType, settings}, {features,..}, {act, dim}}.

Documentation [Local »](#)

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

Options {DropoutRate → 0, RescaleMethod → Pool}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`MakeNode

^

Symbol i

MakeUnet[nClasses, dimIn] Generates a UNET with one channel as input and nClasses as output.

MakeUnet[nChannels, nClasses, dimIn] Generates a UNET with nChannels as input and nClasses as output.

he number of parameter of the first convolution layer can be set with dep. The data dimensions can be 2D or 3D and each of the dimensions should be 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240 or 256. However dimensions can be different based on the network depth and the block type. The implemented block types are "Conv", "UNet", "ResNet", "DenseNet", "Inception", or "U2Net".

Documentation [Local »](#)

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

Options ▸ `NetworkArchitecture` → UNet ... (10 total)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`MakeUnet`

^

Symbol i

MuscleLabelToName[{lab, ..}] converts list of lab, which need to be integers to names using the file `GetAssetLocation["MusclesLegLabels"]`.

MuscleLabelToName[{lab, ..}, file] does the same but uses a user defined ITKSnap label definition file.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`MuscleLabelToName`

^

Symbol i

MuscleNameToLabel[{name, ..}] converts list of muscle names to integer labels using the file `GetAssetLocation["MusclesLegLabels"]`

MuscleNameToLabel[{name, ..}, file] does the same but uses a user defined ITKSnap label definition file.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`MuscleNameToLabel`

^

Symbol i

NetDimensions[net] extracts the input channels, output classes, the input patch dimension, and the number of input filters.

Documentation [Local »](#)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SegmentationTools`NetDimensions`

^

Symbol

NetSummary[net] gives a short summary of the convolution kernels and array elements in the network.

NetSummary[net, what] does the same but what can be "Full" which also includes net and node images or "Mem" which only reports the memory.

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`NetSummary



Symbol



PatchesToData[patches, ran] creates a continuous dataset from the patches. For each patch the range in the data needs to be specified in ran.

The patches are have dimensions {x, y, z} each and ran is specified as {{xmin, xmax}, {ymin, ymax}, {zmin, zmax}}.

PatchesToData[patches, ran, dim] creates a continuous dataset from the patches with dimensions dim.

Documentation [Local](#) »

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`PatchesToData



Symbol

PrepareTrainingData[inFolder, outFolder] prepares the data in de inFolder for training a neural network for segmentation and outputs in outFolder.

PrepareTrainingData[{labFolder, datFolder}, outFolder] does the same but the labels are stored in labFolder and data is stored in datFolder.

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

Options ▶ LabelTag → label ... (6 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`PrepareTrainingData



Symbol



SegmentData[data, what] segments the data. The what specifies the segmentation to be done.

It currently allows for "LegBones" for the bones or "Legs" for the muscles.

Documentation [Local](#) »

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

Options {TargetDevice → GPU, MaxPatchSize → Automatic, Monitor → False}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`SegmentData



Symbol i

SegmentDataGUI[] is a function that creates a graphical user interface (GUI) for segmenting data.

It prompts the user to enter the paths for the input and output files, and allows them to select the segmentation type.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`SegmentDataGUI

^

Symbol i

ShowTrainLog[log] shows the training log of a network training.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`ShowTrainLog

^

Symbol i

SplitDataForSegementation[data] is a specific function for leg data to prepare data for segmentation. It detects the side and location and will split and label the data accordingly.

SplitDataForSegementation[data ,seg] does the same but is rather used when preparing training data. Here the seg is split in exactly the same way as the data.

Documentation [Local »](#)

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

Options Monitor → False

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`SplitDataForSegementation

^

Symbol i

SurfaceDistance[ref, pred] gives the mean surface distance of segmentations ref and pred for class equals 1 in voxels.

SurfaceDistance[x, y, class] gives the mean surface distance of segmentations ref and pred for class in voxels.

SurfaceDistance[x, y, {class, ..}] gives the mean surface distance of segmentations ref and pred for the list of gives classes in voxels.

SurfaceDistance[x, y, class , vox] gives the mean surface distance of segmentations ref and pred for class in milimeter.

SurfaceDistance[x, y, {class, ..}, vox] gives the mean surface distance of segmentations ref and pred for the list of gives classes in milimeters.

Documentation [Local »](#)

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

Options Method → HD95

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`SurfaceDistance

^

Symbol i

TrainSegmentationNetwork[{inFol, outFol}] trains a segmentation network. The correctly prepared training data should be stored in inFol. The progress each round will be saved in outFol.

TrainSegmentationNetwork[{inFol, outFol}, netCont] does the same but defines how to continue with netCont. If netCont is "Start" training will be restarted.

If netCont is a initialized network or network file (wlnet) this will be used. If netCont is a a outFol the last saved network will be used.

Possible loss functions are {"SoftDice", "SquaredDiff", "Tversky", "CrossEntropy", "Jaccard"}.

Documentation [Local »](#)

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

Options ► LoadTrainingData → True ... (20 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`TrainSegmentationNetwork

^

Symbol i

TverskyLossLayer[] represents a net layer that computes the Tversky loss by comparing input class probability vectors with the target class vector.

TverskyLossLayer[b] does the same but b defines the tversky beta factor. With beta = 0.5 its is the Dice coefficient. Here alpha + beta = 1.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`TverskyLossLayer

^

Symbol

QMRITools`SegmentationTools`\$debugUnet

Attributes {ReadProtected}


Full Name QMRITools`SegmentationTools`\$debugUnet

^

Options

Missing[UnknownSymbol, Channels]


Missing[UnknownSymbol, Classes]

Symbol 

Dimensions[*expr*] gives a list of the dimensions of *expr*.
Dimensions[*expr*, *n*] gives a list of the dimensions of *expr* down to level *n*.

Documentation [Local »](#) | [Web »](#)
Options AllowedHeads → Automatic
Attributes {Protected}
Full Name System`Dimensions


^

Symbol 

ActivationType is an option for MakeUnet. It specifies which activation layer is used in the network. It can be "LeakyRELU" or any type allowed by a "name" definition in ElementwiseLayer.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`SegmentationTools`ActivationType


^

Symbol 

AugmentData is an option for GetTrainData and TrainSegmentationNetwork. If set True the trainingdata is augmented.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`SegmentationTools`AugmentData


^

Symbol 

BatchSize is an option for NetTrain and related functions that specifies the size of a batch of examples to process together.

Documentation [Local »](#) | [Web »](#)
Attributes {Protected, ReadProtected}
Full Name System`BatchSize

^

Symbol 

BlockType is an option for MakeUnet. It specifies the type of block used in the network. It can be "Conv", "UNet", "ResNet", "DenseNet", "Inception", or "U2Net".

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`SegmentationTools`BlockType

^


Symbol
CleanUpSegmentations is an option for PrepareTrainingData. If set to True the segmentations are cleaned up by removing holes reducing to one volume and smoothing.
Attributes {Protected, ReadProtected} Full Name QMRITools`SegmentationTools`CleanUpSegmentations
^

Symbol
DataPadding is an option for ApplySegmentationNetwork. Defines how much to pad the data patches in all directions.
Documentation Local » Attributes {Protected, ReadProtected} Full Name QMRITools`SegmentationTools`DataPadding
^

Symbol
DataTag is an option for PrepareTrainingData. It defines the tag used in the filenames of the data.
Attributes {Protected, ReadProtected} Full Name QMRITools`SegmentationTools`DataTag
^

Symbol
DistanceRange is an option for MakeDistanceMap. It defines the range of the distance map outside the segmentation in voxels. Values can be Automatic, All, or a integer value. If All the distance map is calculated for the whole image. If 0 the distance map is only calculated inside the segmentation.
Attributes {Protected, ReadProtected} Full Name QMRITools`SegmentationTools`DistanceRange
^

Symbol
DownsampleSchedule is an option for MakeUnet. It defines how the data is downsampled for each of the deeper layers of the Unet. By default is a factor two for each layer. A custom schedual for a 5 layer 3D Unet could be {{2,2,2},{1,2,2},{2,2,2},{1,2,2}, 1}. The deepest layer is always downsampled by 1 and therefore not needed to be specified.
Documentation Local » Attributes {Protected, ReadProtected} Full Name QMRITools`SegmentationTools`DownsampleSchedule
^

Symbol 


DropoutRate is an option for MakeUnet. It specifies how much dropout is used after each block. It is a value between 0 and 1, default is .2.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`DropoutRate

^

Symbol 

FeatureSchedule is an option for MakeUnet. It defines how the number of features is upsampled for each of the deeper layers of the Unet.


By default it increases the number of features by a factor 2 each layer, i.e. {1, 2, 4, 8, 16}.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`FeatureSchedule

^

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

InputLabels is an option for PrepareTrainingData. Can be set to a list of integers corresponding to the labels to be used from the given segmentation.

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`InputLabels

^

Symbol 

L2Regularization is an option for TrainSegmentationNetwork. It defines the L2 regularization factor.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`L2Regularization

^

Symbol

LabelTag is an option for PrepareTrainingData. It defines the tag used in the filenames of the label data.

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`LabelTag

^

Symbol i

LearningRate is an option for NetTrain that specifies the rate at which to adjust neural net weights in order to minimize the training loss.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected, ReadProtected}

Full Name System`LearningRate

^

Symbol i

LoadTrainingData is an option for TrainSegmentationNetwork. If set to True the training data is loaded from the disk.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`LoadTrainingData

^

Symbol i

LossFunction is an option for NetTrain that specifies how to compare actual and requested outputs from a neural net.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected, ReadProtected}

Full Name System`LossFunction

^

Symbol i

MaxPatchSize is an option for SegmentData and ApplySegmentationNetwork. Defines the patch size used when segmenting data. Bigger patches are better.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`MaxPatchSize

^

Symbol i

MaxTrainingRounds is an option for NetTrain and related functions that specifies the maximum number of rounds of training to do.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected, ReadProtected}

Full Name System`MaxTrainingRounds

^

Symbol i

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 i

Monitor[*expr*, *mon*] generates a temporary monitor cell in which the continually updated current value of *mon* is displayed during the course of evaluation of *expr*.

Documentation [Local »](#) | [Web »](#)
Attributes {HoldAll, Protected, ReadProtected}
Full Name System`Monitor

^

Symbol i

MonitorCalc is an option for many processing functions. When true the proceses of the calculation is shown.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`GeneralTools`MonitorCalc

^

Symbol i

MonitorInterval is an option for TrainSegmentationNetwork. It defines how often the training is monitored.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`SegmentationTools`MonitorInterval


^

Symbol

NetworkArchitecture is an option for MakeUnet. It defines the architecture of the network. It can be "UNet", "UNet+", or "UNet++".
For "UNet+" or "UNet++" it can also be {arch, i} where i specifies how many of the top layers are connected to the mapping layer.

Attributes {Protected, ReadProtected}
Full Name QMRITools`SegmentationTools`NetworkArchitecture

^

Symbol 

NetworkDepth is an option for MakeUnet. It specifief how deep the UNET will be.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`NetworkDepth

^


Symbol

OutputLabels is an option for PrepareTrainingData. Can be set to a list of integers. The used label number will be replaced by these numbers.

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`OutputLabels

^

Symbol 


PatchesPerSet is an option for GetTrainData. Defines how many random patches per dataset are created within the batch.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`PatchesPerSet

^

Symbol 


PatchNumber is an option for DataToPatches. Can be an integer value ≥ 0 . The larger the number the more overlap the patches have.
The minimal number of patches in each direction is calculated, and then for each dimension the given number is added.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`PatchNumber

^

Symbol 

PatchPadding is an option for DataToPatches. Can be an integer value ≥ 0 . It padds the chosen patch size with the given number.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`PatchPadding

^

Symbol i

PatchSize is an option for TrainSegmentationNetwork. Defines the patch size used in the network training.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`PatchSize

^

Symbol

RescaleMethod is an option for MakeUnet. It specifies how the network rescales. It can be "Conv" or "Pool".

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`RescaleMethod

^

Symbol i

RoundLength is an option for TrainSegmentationNetwork. Defines how many batches will be seen during eacht training round.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`RoundLength

^

Symbol i

SettingSchedule is an option for MakeUnet. It defines the settings for the Unet blocks. If one setting is given it applied to all layers.
If a list of settings is given the settings can be different per layer. The following settings are the default settings.
"Unet": convblock repetitions, 2, "ResNet" -> convblock repetitions, 2, "DenseNet" -> {dense depth, block repetitions}, {4,2},
"Inception" -> {inception width, block repetitions}, {4,2}, "U2Net"-> {Unet depth, downscale}, {5, True}.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SegmentationTools`SettingSchedule

^

Symbol i

TargetDevice is an option for certain functions that specifies on which device the computation should be attempted.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected, ReadProtected}

Full Name System`TargetDevice

^

Symbol
TestRun is an option for PrepareTrainingData. If set to True the data is not saved only analyzed.
Attributes {Protected, ReadProtected}
Full Name QMRITools`SegmentationTools`TestRun
^

SimulationTools

Functions

Symbol i
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 i
BlochSeries[vectorIn, deltata, freqRange, B1] performs a Bloch simulation of an RF pulse.
Documentation Local »
Attributes {Protected, ReadProtected}
Full Name QMRITools`SimulationTools`BlochSeries
^

Symbol i
CalculateGfactor[factors, sensitivity, wMat] calculates a gfactor for given sensitivity maps and noise corralltion 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 i

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 i

ErnstAngle[] shows Ernst angle plot for t1 = 1400ms and tr = 15 ms.

ErnstAngle[t1] shows Ernst angle plot for t1 and tr = 15 ms.

ErnstAngle[t1, tr] shows Ernst angle plot for t1 and tr.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`ErnstAngle

^

Symbol i

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 i

GetPulseProfile[excitation, refocus] gives the pulsed angle profiles for the excitation and refocussing pulses.

a pulse is defined as {"name", flipangle, {G_strnth, Dur, BW}}.

GetPulseProfile[{"name", flipangle, {G_strnth, Dur, BW}}] gives detailed slice profile information of one pulse.

output is {ex_angle_profile, ref_angle_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 i

GfactorSimulation[sensitivity, cov, {dir,sense}] calculates the gfactor maps for given sensitivity maps and noise correlation 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 gfactor maps for given sensitivity maps and noise correlation 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 i

PlotSimulation[pars, xval, true, label, color] plots the pars (output from Parameters). Using label as PlotLabel and xval as x axis. 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 i

Pulses[name] gives the pulse shape of some predefined Philips pulse shapes.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`Pulses

^

Symbol i

Signal[par, tr, te] calculates the MRI signal at a given tr and te. Par is defined as {pd, t1, t2}.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`Signal

^

Symbol i

SimAngleParameters[tens,vec] calculates 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 i

SimParameters[tens] calculates 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 i

SimulateDualTR[] simulates the signal of a Dual tr t1 map.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[SimulateDualTR] = {ArgumentsPattern → {...}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SimulateDualTR

^

Symbol i

SimulateSliceEPG[excitation, refocus, {{t1, t2}, {nEcho, echoSp}, b1}] gives a simulated slice profile and EPG signal plot.
excitation and refocus are generated by `GetPulseProfile`.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[SimulateSliceEPG] = {ArgumentsPattern → {...}, OptionsPattern[]}`

Options `ReportFits → False`

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SimulateSliceEPG

^

Symbol i

Tensor[{l1, l2, l3}] creates a diffusion 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 diffusion tensor with vectors $\{e1, e2, e3\}$ and eigenvalues $\{l1, l2, l3\}$.

Tensor[{l1, l2, l3}, "Random"] creates a diffusion tensor with random orthogonal eigenvectors $\{e1, e2, e3\}$ and eigenvalues $\{l1, l2, l3\}$.

Tensor[{l1, l2, l3}, "RandomZ"] creates a diffusion 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 diffusion 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 i

FatFieldStrength is an option for GetPulseProfile. If the value >0 it will calculate the shift of the fat refocussing pulse compared to the fat excitation pulse.
The shift is in SliceRangeSamples steps.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`SimulationTools`FatFieldStrength

^

Symbol i

GOutput is an option for GfactorSimulation. can be "Grid" or "List".

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`SimulationTools`GOutput

^

Symbol i

GRegularization is an option for CalculateGfactor and GfactorSimulation.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`SimulationTools`GRegularization

^

Symbol i

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 i

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 i

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 i

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 i

Reject is an option for EigenvalCalc. If True then voxels with negative eigenvalues are rejected and set to 0.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`Reject

^

Symbol i

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 i

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 i

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 i

SortVecs is an option for PlotSimulationVec.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SimulationTools`SortVecs

^

Symbol i

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 i

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 i

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 i

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 i

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 i

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 i

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 i

ChangeDwellTimeFid[fid, dt, dtnew] changes the sampling 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 i

CompareFidFitPlot[time, fidPlot, fitPlot] function not done.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SpectroTools`CompareFidFitPlot`

^

Symbol i

CompareSpectraFitPlot[ppmPl, specPlot, fitPlot] function not done.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`SpectroTools`CompareSpectraFitPlot`

^

Symbol i

CorrectTEFid[fid, dw, te] corrects the fid for 1st order phase by extrapolating the missing FID samples in the TE using Henkel matrix SVD ansalsis.

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 ansalsis. 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 i

CorrectTESpec[spectra, dw, te] corrects the spectra for 1st order phase by extrapolating the missing FID samples in the TE using Henkel matrix SVD ansalsis.

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 ansalsis. 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 i

CSIIInterface[] opens the CSI interface. Function not done.

CSIIInterface[te, bw] opens the CSI interface with known te and bw.

CSIIInterface[file] opens the CSI interface with the data from file loaded.

CSIIInterface[file, {tei, bwi}] opens the CSI interface with the data from file loaded with known te and bw.

Documentation [Local »](#)


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

Options {SpectraFieldStrength → 7, SpectraNucleus → 31P}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`CSIIInterface

^

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 i

GetGyro[nuc, field] geth the gyromagnetic ratio with field the field strength in Tesla and nuc the nucleus available in GyromagneticRatio.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`GetGyro

^

Symbol i

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 available 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 i

GetSpectraBasisFunctions[{met1, ..., metn}] generates a list of spectra basis functions with names met1 to metn. The names are strings and are the metabolites available in GetSpinSystem.

GetSpectraBasisFunctions[{{props1}, ..., {propsn}}] generates a list of spectra basis 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 available 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 increase 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 increase 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 values.

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 before 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 → {_, dw, te, gyro, ppmRan}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PhaseCorrectSpectra



Symbol



PhaseShiftSpectra[spectra, phi0] applies the 0th order phase phi0 to the spectra.

PhaseShiftSpectra[spectra, ppm, gyro, phi1] applies 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}] applies 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 i

PlotFid[fid, dwell] plots the fid assuming dwell as the sampling 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 i

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 ppm range of the spectra which can be obtained with GetPpmRange.

Documentation [Local](#) »

Default Definitions `SyntaxInformation[PlotSpectra] = {ArgumentsPattern -> {_, _}, OptionsPattern[]}`

Options » PlotRange -> Full ... (12 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`PlotSpectra

^

Symbol i

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 i

ShiftSpectra[spec, {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 i

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 i

TimeShiftEcho[fid, time, gam] applies a linebroadening with linewidth gam and a Voigt lineshape to the fid. The time can be obtained using GetTimeRange.

TTimeShiftEcho[fid, time, {gam, f}] applies a linebroadening with linewidth gam and a custom lineshape f to the fid (f=0, "Gaussian", f=1 "Lorentzian").

TTimeShiftEcho[fid, time, gyro, {gam, eps}] applies 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}] applies 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 i

TimeShiftFid[fid, time, gam] applies a linebroadening with linewidth gam and a Voigt lineshape to the fid. The time can be obtained using GetTimeRange.

TimeShiftFid[fid, time, {gam, f}] applies 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}] applies 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}] applies 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 i

TimeShiftFidV[fid, time, gam] applies a linebroadening with linewidth gam and a Voigt lineshape to the fid. The time can be obtained using GetTimeRange.

TimeShiftFidV[fid, time, {gamL, gamG}] applies a linebroadening with linewidth gamG "Gaussian" and gamL "Lorentzian".

TimeShiftFidV[fid, time, gyro, {gam, eps}] applies 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.

TimeShiftFidV[fid, time, gyro, {{gamL, gamG}, eps}] applies a linebroadening with linewidth linewidth gamG "Gaussian" and gamL "Lorentzian" 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 i

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 i

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 i

BasisSequence is an option for GetSpectraBasisFunctions and specifies which sequence to use.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`SpectroTools`BasisSequence

^

Symbol i

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

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`JcouplingTools`CenterFrequency

^

Symbol i

Filling is an option for ListPlot, Plot, Plot3D, and related functions that specifies what filling to add under points, curves, and surfaces.

Documentation [Local »](#) | [Web »](#)
Attributes {Protected}
Full Name System`Filling


^

Symbol i

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 i

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 i

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 i

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 i

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 i

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 i

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 i

ReadoutType is an option for FitSpectra and padding and apodization fuctions. Value can be "Fid" or "Echo".

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`ReadoutType

^

Symbol i

SparID is an option for ExportSparSdat.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SparID

^

Symbol i

SparName is an option for ExportSparSdat.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SparName

^

Symbol i

SparOrientation is an option for ExportSparSdat.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SparOrientation

^

Symbol i

SpectraBandwidth is an option for GetSpectraBasisFunctions and sets the bandwidth of the spectra.

Documentation [Local »](#)
Attributes {Protected, ReadProtected}
Full Name QMRITools`SpectroTools`SpectraBandwidth

^

Symbol i

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 i

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 i

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 i

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 i

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 i

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 i

SplineSpacingFactor is an option for FitSpectra and defines the distance between the bsplien points relative the the mean linewidthd of the peaks.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`SpectroTools`SplineSpacingFactor

^

TaggingTools

Functions

Symbol i

AnnalyzeTagging[gridC] ...

Documentation [Local »](#)

Options {HistoryWeighting → 0.7, MonitorTagging → True}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TaggingTools`AnnalyzeTagging

^

Symbol i

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



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 i

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 throughtout 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 i

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 i

EigensysCalc[tensor] calculates the eigensystem for the given tensor.

Documentation [Local »](#)

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

Options {RejectMap → False, Reject → True, PerformanceGoal → Speed}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`EigensysCalc

^

Symbol i

EigenvalCalc[tensor] caculates the eigenvalues for the given tensor.

Documentation [Local »](#)

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

Options {RejectMap → False, Reject → True, PerformanceGoal → Speed}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`EigenvalCalc

^

Symbol i

EigenvecCalc[tensor] caculates the eigenvectors for the given tensor.

Documentation [Local »](#)

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

Options `{RejectMap → False, Reject → True, PerformanceGoal → Speed}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TensorTools`EigenvecCalc`

^

Symbol i

ExpTensor[tensor] transforms the tensor from LogEuclidian space.

ExpTensor[] is based on DOI: 10.1109/42.963816.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[ExpTensor] = {ArgumentsPattern → {_}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TensorTools`ExpTensor`

^

Symbol i

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 i

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 i

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 i

LogTensor[tensor] transforms the tensor to LogEuclidian space.

LogTensor[] is based on DOI: 10.1109/42.963816.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`LogTensor

^

Symbol i

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 mm²/s).

Documentation [Local »](#)

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

Options {Reject → False, PerformanceGoal → Speed}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`ParameterCalc

^

Symbol i

Removelsolimages[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[Removelsolimages] = {ArgumentsPattern → {_, _ ...}}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`Removelsolimages

^

Symbol i

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 i

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 i

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 $\{\{x_1, y_1, z_1\}, \dots, \{x_n, y_n, z_n\}\}$ without the unweighted gradient direction.

bvalue is a single 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 $\{\{x_1, y_1, z_1\}, \dots, \{x_n, y_n, z_n\}\}$ 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 $\text{mm}^2/2$.

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 i

TransformTensor[tensor, disp, vox] corrects the tensor with voxel size vox based on the displacementfield disp. The displacementfield is the displacement in mm for each voxel location in x, y and z.

TransformTensor[] is based on DOI: 10.1109/42.963816.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`TransformTensor

^

Symbol i

WestinMeasures[eigenvalues] calculates the westin measures.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`WestinMeasures

^

Options

Symbol i

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 i


FilterShape is an option for SigmaCalc. Can be "Gaussian" or "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 many processing functions. When true the proceses of the calculation is shown.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`GeneralTools`MonitorCalc

^

Symbol 

NormalizeSignal is an option for DriftCorrect.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`NormalizeSignal

^

Symbol i

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

Definitions

```
Parallelize[Parallel`Kernels`Private`args$____] := (Parallel`Protected`doAutolaunch[TrueQ[Parallel`Static`$enableLaunchFeedback]] ;
Parallelize[Parallel`Kernels`Private`args$])
```

Documentation [Local »](#) | [Web »](#)

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

Attributes {HoldFirst, Protected}

Full Name System`Parallelize

^

Symbol i

PerformanceGoal is an option for various algorithmic and presentational functions that specifies what aspect of performance to try to optimize with Automatic settings for options.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected}

Full Name System`PerformanceGoal

^

Symbol i

Reject is an option for EigenvalCalc. If True then voxels with negative eigenvalues are rejected and set to 0.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`Reject

^

Symbol i

RejectMap is an option for EigenvalCalc. If Reject is True and RejectMap is True both the eigenvalues as well as a map showing the rejected values is returned.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`RejectMap

^

Symbol i

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 i

RobustFitParameters is an option for TensorCalc. gives the threshold for stopping the iterations and the kappa for the outlier marging, {tr,kappa}.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TensorTools`RobustFitParameters

^

Symbol i

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 i

UseMask is a function for MeanSignal and DriftCorrect.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`ProcessingTools`UseMask

^

TractographyTools

Functions

Symbol i

ExportTracts[file, tracts, vox, dim, seeds] exports the tracts, vox, dim and seeds to *.trk file.


Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`ExportTracts

^

Symbol 

FiberLength[tracts] calculates the length of each tract.


Documentation [Local »](#)

Default Definitions `SyntaxInformation[FiberLength] = {ArgumentsPattern -> {...}}`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`FiberLength`

^

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 stopping 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** -> {20, 500} ... (10 total)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`FiberTractography`

^

Symbol i

FilterTracts[tracts, vox, {select..}] filters the tracts based on the list of select criteria.

Select criteria are defined as {"logic", {"how", criteria}}.

The "logic" parameter can be "and", "or" and "not".

The "how" parameter can be:

- "x", "y", or "z" for slice selection, here criteria is a slice number
- "through" for selecting tract that go through a roi, here criteria is a 3D mask.
- "within" for selecting tract that fit fully within the roi, here criteria is a 3D mask.
- "partwithin" for selecting the part of the tracts that fall within the roi, here criteria is a 3D mask.

Any number of select criteria can be listed.

Documentation [Local »](#)

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

Options FiberLengthRange → {20, 500}

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FilterTracts

^

Symbol i

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 → {20, 500} ... (7 total)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FindTensorPermutation

^

Symbol i

FitTracts[tract] fits a tract or a list of tracts, each defined as a list of {x, y, z} coordinates with a polynomial function.

FitTracts[tract, vox, dim] does the same but constrains all the tract coordinates to the volume defined by dim.

Documentation [Local »](#)

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

Options `FittingOrder → 3`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`FitTracts`

^

Symbol i

GetTractValues[tracts, parameter, vox] gets the value of the parameter map at each tract coordinate.

Documentation [Local »](#)

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

Options `InterpolationOrder → 1`

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`GetTractValues`

^

Symbol i

ImportTracts[file] imports a *.trk file. It can contain {tracts, vox, dim, seeds}.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`ImportTracts`

^

Symbol i

MoveTracts[tracts, off] moves the tract coordinates by off, which is {x, y, z}.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`MoveTracts`

^

Symbol i

PlotSegmentedTracts[tracts, segments, dim, vox] plots the tracts after segmenting each segments.

PlotSegmentedTracts[tracts, segments, bones, dim, vox] plots the tracts after segmenting each segments also rendering a bone volume.

Documentation [Local »](#)

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

Options ▸ `MaxTracts` → 5000 ... (6 total)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`PlotSegmentedTracts`

^

Symbol i

PlotTracts[tracts, vox] plots the tracts assuming an Boxratio based on vox.

PlotTracts[tracts, vox, dim] plots the tracts assuming a Boxratio based on vox with a PlotRange spanning the full dim.

Documentation [Local »](#)

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

Options ▸ `MaxTracts` → 2000 ... (10 total)

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`PlotTracts`

^

Symbol i

ResampleTracts[tracts, n] resample each Tract to exactly n vertices.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`ResampleTracts`

^

Symbol i

RescaleTracts[tracts, sc] scales the tract coordinates by 1/sc, which is {x, y, z} or single number.

Documentation [Local »](#)

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

Attributes `{Protected, ReadProtected}`

Full Name `QMRITools`TractographyTools`RescaleTracts`

^

Symbol i

SeedDensityMap[seeds, vox, dim] makes a seed density map based on the seed loactions.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`TractographyTools`SeedDensityMap`

^

Symbol i

SegmentTracts[tracts, segs, vox, dim] segments the tracts based on segs.

Documentation [Local »](#)

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

Options {FiberLengthRange → {15, 500}, OutputForm → Joined, FitTractSegments → True}

Attributes {Protected, ReadProtected}

Full Name `QMRITools`TractographyTools`SegmentTracts`

^

Symbol i

TractAngleMap[tracts, vox, dim] makes a tract angle map based on the tracts angles with the z-plane.

Documentation [Local »](#)

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

Attributes {Protected, ReadProtected}

Full Name `QMRITools`TractographyTools`TractAngleMap`

^

Symbol i

TractDensityMap[tracts, vox, dim] makes a tract density map based on the tracts vertices.

Documentation [Local »](#)

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

Options `NormalizeDensity → True`

Attributes {Protected, ReadProtected}

Full Name `QMRITools`TractographyTools`TractDensityMap`

^

Symbol i

TractLengthMap[tracts, vox, dim] makes a tract length map based on the tracts lengths.

Documentation [Local »](#)

Default Definitions `SyntaxInformation[TractLengthMap] = {ArgumentsPattern -> {_, _, _}}`

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`TractLengthMap

^

Options

Symbol i

Boxed is an option for Graphics3D that specifies whether to draw the edges of the bounding box in a three-dimensional picture.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected}

Full Name System`Boxed

^

Symbol i

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 i

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 i


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 FitTracts. It specifies the polinomial order of the function to fit the tract.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FittingOrder

^


Symbol

FitTractSegments is an option for SegmentTracts. If set True the segmented tracts are fitted with FitTracts.

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`FitTractSegments

^

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 seedpoints to be used.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`MaxSeedPoints

^

Symbol 

MaxTracts is an option for PlotTracts. It specifies how many tracts are plotted.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`MaxTracts

^

Symbol i

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 i

Monitor*[expr, mon]* generates a temporary monitor cell in which the continually updated current value of *mon* is displayed during the course of evaluation of *expr*.

Documentation [Local »](#) | [Web »](#)

Attributes {HoldAll, Protected, ReadProtected}

Full Name System`Monitor

^

Symbol i

NormalizeDensity is an option for TractDensityMap. If set True the tractdensity is normalized, if False then it is the true tract count.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`NormalizeDensity

^

Symbol i

OutputForm*[expr]* prints as a two-dimensional representation of *expr* using only keyboard characters.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected}

Full Name System`OutputForm

^

Symbol i


PerformanceGoal is an option for various algorithmic and presentational functions that specifies what aspect of performance to try to optimize with Automatic settings for options.

Documentation [Local »](#) | [Web »](#)

Attributes {Protected}

Full Name System`PerformanceGoal

^

Symbol 


StepSize is an option for FiberTractography and defines the tractography step size.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`StepSize

^

Symbol 


StopThreshold is an option for FiberTractography and defines the stop threshold which is a value between 0 and 1.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`StopThreshold

^

Symbol 


TensorFilps is an option for FiberTractography and speciefies if the tensor orientation is flipped, 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

^

Symbol i

TractColoring is an option for PlotTracts and sets how the tracts are colored. Values can be "Direction", "Length", "Angle", {par}, or RGBColor[].

For "Length", "Angle", {par} it can be defined in the form {..., {min, max}} where the {min, max} specifies the range of the color function.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`TractColoring

^

Symbol i

TractReduction is an option for PlotTracts. Value can be an Integer > 0, which determines with which factor the tract coordinates are subsampled.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`TractReduction

^

Symbol i

TractScaling is an option for PlotTracts. The value can be "World" or "Voxel", if the value is "Wold" the tracts are in mm else in voxel coordinates.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`TractScaling

^

Symbol i

TractSize is an option for PlotTracts. When tubes are used it specifies the tube withd.

Documentation [Local »](#)

Attributes {Protected, ReadProtected}

Full Name QMRITools`TractographyTools`TractSize

^