**Functions**

**CSIgui\_OpeningFcn(hObject, eventdata, gui, varargin)**

Initializes all GUI settings and creates application data before the figure window is made visible. To improve feedback to user a loading bar, loadBar() is shown during this function. User input, if any, is processed. GUI Color scheme is applied here.

output: gui.ID; gui.version; gui.inp;

input: varargin from input arguments using CSIgui(…);

update: appdata-gui;

other: The input argument tags are used to process user input data: 'data', 'list', 'csi', 'spec', 'image', 'labels', 'mrs', 'filepath', 'filepathi', 'fp', 'fpi'.

functions: CSIgui\_setMenuBar;

**gui = CSIgui\_setMenuBar(gui)**

Set the menu bar of the application GUI.

**gui = CSIgui\_setMenuBar\_JavaScript(gui)**

Set tool tip strings for menu bar entries using java script.

**CSIgui\_OutputFcn(hObject, eventdata, gui)**

Sets the output of CSIgui. Currently set to no output.

**button\_DeleteInput\_Callback(hObj, ~, gui)**

Button to delete user input manually using delete\_UserInput().

**delete\_UserInput(hObj, gui)**

Deletes the field inp from appdata-gui.

**openFile(hObj, ~, ~)**

Opens any file compatible with CSIgui. Checks for field inp in appdata-gui to load input from user. If no input data is present, a file selection GUI is opened. Otherwise the inp struct from the appdata-gui is analyzed: filepath and/or filepathi. If the user input contains both the image and spectroscopy data, the field openThisFile is added to the appdata-gui and used to run openFile twice, loading both data sets. Log is cleared prior to parsing the file(s). Files are handled using their extensions and the corresponding parse-functions is called, for user input this extension is set to *“userinput”*. After parsing, the file name display in the GUI is updated. The appdata-csi and appdata-mri are created during this function using the file specific parse-functions.

output: appdata-csi and/or appdata-mri;

input: -

update: appdata-gui;

other: -

functions: CSI\_ReorderDim\_Auto; CSI\_2D\_Scaling\_calc\_xaxis; CSI\_Log

**success = parse\_listdata(fp, fn, gui)**

Parse and load the list and data file. Creates appdata-csi.

**success = parse\_sdatspar(fp, fn, gui)**

Parse and load the spar and sdat file. Creates appdata-csi.

**success = parse\_text(fp, fn, gui)**

Parse and load a text file with MRS data. Creates appdata-csi.

**success = parse\_mat(fp, fn, gui)**

Parse and load a mat file. The mat-file is expected to contain struct with fields: mri, conv, csi, log, xaxis, ori; Creates appdata-csi, appdata-mri, appdata-ori and/or appdata-conv.

**success = parse\_dicom(fp, fn, gui)**

Parse and load dicom files with image data. Creates appdata-mri.

**success = parse\_parrec(fp, fn, gui)**

Parse and load par and rec files with image data. Creates appdata-mri.

**success = parse\_protocolText(fp, fn, gui)**

Parse and load text file with protocol information. The xaxis and coordinate fields in appdata-csi are created automatically.

**success = parse\_protocolSpar(fp, fn, gui)**

Parse and load spar file with protocol information. The xaxis and coordinate structures in appdata.csi are created automatically.

**parse\_userinput(gui)**

Parse and load user input expecting input labels: 'data', 'list', 'csi', 'spec', 'image', 'mrs', 'labels'. These labels are set as fieldnames in the appdata-gui field inp. Creates either appdata-mri or appdata-csi.

**button\_CSI\_ReadInfo\_Callback(hObj, eventdata, gui)**

Button to read a file with protocol information. Either spar or text files can be opened. The text file should be exported from the scan environment (Philips).

**openHelp(~,~, ~)**

**openAbout(hObject,~ ,~)**

**userInput = getUserInput(question, defans, clrs)**

**getUserInput\_editKeyPress(H, E)**

**getUserInput\_setOutput(hObject,~,~)**

**userInput = getUserInput\_Popup(popup\_title, popup\_input, clrs)**

**getUserInput\_Popup\_setOutput(hObject,~,~)**

**userInput = getUserInput\_Buttons(qst, button\_text, clrs)**

**getUserInput\_Buttons\_setOutput(hObject,~,~)**

**function userInput = getUserInput\_Radio(question, defans, clrs)**

**function getUserInput\_Radio\_setOutput(hObject,~,~)**

**function button\_CSI\_Average\_Callback(~, ~, gui)**

**function button\_CSI\_Mean\_Callback(~, ~, gui)**

**function [data\_averaged, index] = CSI\_average(data,index)**

**function button\_CSI\_FFT\_Kspace\_Callback(~, ~, gui)**

**function button\_CSI\_FFT\_Callback(~, ~, gui)**

**function button\_CSI\_iFFT\_Callback(~, ~, gui)**

**function button\_CSI\_Apodization\_Kspace\_Callback(~, ~, gui)**

**function button\_CSI\_Apodization\_FID\_Callback(~, ~, gui)**

**function [data, window] = CSI\_filterSpectra(data, ftype, opts)**

**function filter\_window = CSI\_filter(fsz, ftype, opts)**

**function button\_CSI\_setFrequency\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_setLabels\_Callback(~, ~, gui)**

**function button\_CSI\_ReorderDim\_Callback(~, ~, gui)**

**function button\_CSI\_ReorderDim\_Auto\_Callback(hObject, ~, ~)**

**function CSI\_ReorderDim\_Auto(gui)**

**function button\_CSI\_Sum\_Callback(~, ~, gui)**

**function [data\_summed, index] = CSI\_summate(data, index)**

**function button\_T1\_MRS\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_T2\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_ZeroFill\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_AutoPhase\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_Flip\_Callback(~, ~, gui)**

**function button\_CSI\_Rotate\_Callback(~, ~ , gui)**

**function button\_CSI\_Multiply\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_Divide\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_Combine\_Callback(~, ~, gui)**

**function CSI\_Combine\_Manual(hobj, ~)**

**function CSI\_Combine\_WSVD(hobj,~)**

**function button\_CSI\_MaxValue\_Callback(~, ~, gui)**

**function CSI\_Max\_Per\_Slice(~, ~, gui)**

**function CSI\_Max\_In\_3D(~, ~, gui)**

**function CSI\_max3D(data,clrs)**

**function button\_CSI\_SNR\_Callback(~, ~, gui)**

**function button\_ws\_Callback(~, ~, gui)**

**function button\_Info\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_frequencyShift\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_VoxelShift\_Callback(hObject, eventdata, gui)**

**function button\_CSI\_FidOrEcho\_Callback(hObject, eventdata, gui)**

**function button\_Log\_DeleteLine\_Callback(hObj, eventdata, gui)**

**function CSI\_dataAsGraph2(data, data\_text, datatag, gui, img)**

**function CSI\_dataAsGraph(data, gui, datatag)**

**function CSI\_dataAsTable(data, datatag)**

**function [sliceArray, sliceIndex] = slices2rowArray(data, split)**

**function indArray = slice2rowIndex(ndimCell)**

**function CSI\_dataTable\_CellSelection(hObj, evt)**

**function CSI\_dataAsTable\_SaveButton(hObj, ~)**

**function button\_CSI\_setCoordinates\_Callback(~, ~, gui)**

**function button\_MRIcoordinates\_Callback(~, ~, gui)**

**function MRI\_coordinates\_DCM(gui, mri)**

**function MRI\_coordinates\_PAR(gui, mri)**

**function button\_MRIconvert2csi\_Callback(~, ~, gui)**

**function [img, img\_all, img\_all\_slice\_range] = MRI\_matchSlices(hObj)**

**function MRI\_to\_CSIspace(gui)**

**function MRI\_saveIMG(hObj,~)**

**function button\_plotCSI\_Callback(hObject, eventdata, gui)**

**function CSI\_2D\_initiate2D(~,~,~)**

**function plot\_par = CSI\_2D\_setFigure(plot\_par, init\_pos, fig\_tag)**

**function plot\_par = CSI\_2D\_getData(plot\_par, gui, data\_volume)**

**function plot\_par = CSI\_2D\_getPlotSettings\_ColorScaling(plot\_par, data\_volume)**

**function plot\_par = CSI\_2D\_getPlotSettings(plot\_par, gui, data\_volume)**

**function plot\_par = CSI\_2D\_plotImages(plot\_par, csiguiObj)**

**function CSI\_2D\_plotVoxels(plot\_par, gui)**

**function CSI\_2D\_grid(target, target\_sz, dim, range, grid\_clr)**

**function CSI\_close2D(hObj, ~)**

**function CSI\_2D\_voxel\_select(hObject, evt, ~)**

**function CSI\_2D\_voxel\_selectMultiple(hObj, ~, ~)**

**function data2D = CSI\_2D\_voxel\_selected\_getData(data2D)**

**function CSI\_2D\_Scaling\_plotColorbar(hObj,~)**

**function scaleby = CSI\_2D\_Scaling\_Color\_Get(gui)**

**function scaleby = CSI\_2D\_Scaling\_Axis\_Get(gui)**

**function CSI\_2D\_Scaling\_Color\_Set(hObject, evt)**

**function CSI\_2D\_Scaling\_Axis\_Set(hObject, evt)**

**function [clrs, clrs\_range] = CSI\_2D\_Scaling\_calc\_ColorOfPlots(data\_limits)**

**function CSI\_2D\_Scaling\_calc\_xaxis(hObject, ~, auto)**

**function CSI\_2D\_Scaling\_Options(gui)**

**function button\_CSI\_DisplayOptions\_Callback(~, ~, gui)**

**function panel\_2D\_DataSliders(~, ~, gui)**

**function panel\_2D\_sliders(hObject, eventdata)**

**function domain = CSI\_getDomain(gui)**

**function CSI\_setDomain(hObj,evt)**

**function data = CSI\_getUnit(data, unit\_str)**

**function [doi, doi\_axis, range] = CSI\_getDataAtPeak(spec, xaxis, range)**

**function range = CSI\_getPeakOfInterest(xaxis, poi\_tag)**

**function [doi, doi\_axis, doi\_range] = CSI\_getDataAtPeak\_Stored(range, gui)**

**function CSI\_1D\_initiateGUI(data1D, varargin)**

**function CSI\_1D\_snap2fig\_set(CSI\_1Dobj, instance)**

**function CSI\_1D\_snap2fig(~, ~)**

**function CSI\_1D\_movePanelToGUI(hObj, ~)**

**function CSI\_1D\_closeGUI(hObj, ~)**

**function CSI\_1D\_displayData(CSI\_1D\_obj)**

**function title\_str = CSI\_1D\_display\_createTitle(index)**

**function [CSI\_1DObj, appdata1D, data] = CSI\_1D\_getData(CSI\_1DObj)**

**function CSI\_1D\_panel(instance)**

**function obj1D = panel\_1D\_getInstanceData(hObj,tag)**

**function panel\_1D\_PhasingMethod(hObj, ~)**

**function panel\_1D\_PhaseCorrection\_Manual(hObj, ~)**

**function panel\_1D\_PhaseCorrection\_ApplyToAll(hObj,~)**

**function panel\_1D\_PhaseCorrection\_Auto(hObj, ~)**

**function panel\_1D\_ZeroFill(hObj, ~)**

**function panel\_1D\_SNR(hObj, ~)**

**function panel\_1D\_FWHM(hObj, ~)**

**function panel\_1D\_Baseline(hObj,~)**

**function panel\_1D\_SaveToMain(hObj, ~)**

**function panel\_1D\_FFT(hObj, ~)**

**function panel\_1D\_iFFT(hObj, ~)**

**function panel\_1D\_Apodization(hObj,~)**

**function panel\_1D\_DataDisplay(hObj, ~)**

**function panel\_1D\_Export(hObj, ~)**

**function CSI\_Log(varargin)**

**function CSI\_Log\_deleteLine(hObj)**

**function CSI\_viewNoise(hObject, ~, ~)**

**function CSI\_saveData(hObject, ~, ~)**

**function CSI\_saveSDAT(hObject, ~, ~, varargin)**

**function CSI\_saveTXT(hObject, ~, ~, varargin)**

**function CSI\_saveMAT(hObject, ~, ~,varargin)**

**function CSI\_saveFig(hObject, ~, ~)**

**function CSI\_saveLog(hObject, ~, ~)**

**function button\_setBackup\_Callback(~, ~, gui)**

**function button\_getBackup\_Callback(~, ~, gui)**

**function CSI\_backupSet(gui, info\_str)**

**function CSI\_backupGet(gui, askUser)**

**function CSIgui\_Undo(hObj,~,~)**

**function CSI\_backupManager(gui)**

**function button\_plotIMG\_Callback(hObj, ~, ~)**

**function button\_MRIcontrast\_Callback(hObject, eventdata, gui)**

**function CSIgui\_main\_CloseFromMenu(hObject, ~)**

**function CSIgui\_main\_CloseRequestFcn(hObject, ~, ~)**

**function setGUIcolor\_bymenu(hObj, evt, ~)**

**function setGUIcolor(hObject)**

**function setGUIcolor\_custom(hObj,~,~)**

**function colors = setGUIcolor\_custom\_read()**

**function setGUIcolor\_custom\_pickColor(hObj,evt,~)**

**function setGuiColor\_custom\_Done(hObj,~, ~)**

**function loadBar(perc, info)**

**function listbox\_CSIinfo\_Callback(hObject, eventdata, gui)**

**function listbox\_CSIinfo\_CreateFcn(hObject, eventdata, gui)**

**function listbox\_MRIinfo\_Callback(hObject, eventdata, gui)**

**function listbox\_MRIinfo\_CreateFcn(hObject, eventdata, gui)**

**function popup\_plotUnit\_Callback(hObject, eventdata, gui)**

**function popup\_plotUnit\_CreateFcn(hObject, eventdata, gui)**

**function popup\_plotIMG\_Callback(hObject, eventdata, gui)**

**function popup\_plotIMG\_CreateFcn(hObject, eventdata, gui)**

**function button\_TestSomething\_Callback(hObj, eventdata, gui)**

**function CSI\_showAllImages\_Something(hObj)**

**function CSI\_MergeVoxels\_Initiate(hObj, gui)**

**function button\_CSI\_MergeVoxels\_Callback(hObj, ~, gui)**

**function CSI\_MergeVoxels\_plotVoxels(hObj, gui, plot\_par)**

**function selected = CSI\_MergeVoxels\_getSelected(hObj)**

**function CSI\_MergeVoxels\_Button\_Merge(hObj,~)**

**function CSI\_MergeVoxels\_Button\_Save(hObj,~)**

**function CSI\_MergeVoxels\_SaveData(hObj, fp, fn)**

**function CSI\_MergeVoxels\_SaveFig(hObj, fp, fn)**

**function CSI\_MergeVoxels\_Average(sel)**

**function [poi, pex] = CSI\_MergeVoxels\_POI(xaxis)**

**function sel = CSI\_MergeVoxels\_SNR\_Filter(sel)**

**function snr = CSI\_MergeVoxels\_SNR(sel, poi\_tag)**

**function sel = CSI\_MergeVoxels\_Weighted(sel)**

**function sel = CSI\_MergeVoxels\_Align(sel)**

**function button\_MRI\_plotImage\_Grid\_Callback(hObj, ~, gui)**

**function button\_Flip092\_Callback(~, ~, gui)**

**function button\_CSI\_AutoProcessing\_Callback(hObj, evt, gui)**

**function CSI\_Normalize(gui)**

**function button\_Normalize\_Callback(~ , ~, gui)**