Appendix

Appendix A

A script for analysis of CEST images

Analysis of CEST images

A script written in Matlab software for calculation of CEST maps

```
function varargout = CEST_processing(varargin)
% CEST_PROCESSING M-file for CEST_processing.fig
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @CEST_processing_OpeningFcn, ...
                   'gui_OutputFcn',  @CEST_processing_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function CEST_processing_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = CEST_processing_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% The pushbutton opens an image acquired at the frequency offset of the
CEST agent
function pushbutton1_Callback(hObject, eventdata, handles)
axes(handles.axes1);
delete(get(handles.axes1,'Children'));

%the function ,,load_image_cest,, shows
the image acquired at the frequency offset of the CEST agent into axes1
the image acquired at the opposite frequency offset into axes2
the image acquired without saturation into axes3
and loads the following parameters:
%handles.matica1/2/3 - the image acquired at the frequency offset of the
CEST agent
%handles.maximum1/2/3 - the maximum signal intensity of the image
%handles.minimum1/2/3 - the minimum signal intensity of the image
%handles.nslice1/2/3 - number of slices,
%handles.mtrx1/2/3 - matrix size in x-axis
%handles.mtry1/2/3 - matrix size in y-axis
%handles.thk1/2/3 - slice thickness
%handles.fovx1/2/3 - field of view in x-axis
%handles.fovy1/2/3 - field of view in y-axis

global matica1
[handles.matica1, handles.maximum1, handles.minimum1, handles.nslice1,
handles.mtrx1, handles.mtry1, handles.thk1, handles.fovx1,
handles.fovy1]=load_image_cest(hObject, handles, handles.axes1);
matica1=handles.matica1;
guidata(hObject, handles);
```

```
% The pushbutton opens an image acquired at the opposite frequency offset
to the CEST agent
function pushbutton2_Callback(hObject, eventdata, handles)
axes(handles.axes2);
delete(get(handles.axes2,'Children'));

[handles.matica2, handles.maximum2, handles.minimum2, handles.nslice2,
handles.mtrx2, handles.mtry2, handles.thk2, handles.fovx2,
handles.fovy2]=load_image_cest(hObject, handles, handles.axes2);
matica2=handles.matica2;
guidata(hObject, handles);

% The pushbutton opens an image acquired without saturation
function pushbutton3_Callback(hObject, eventdata, handles)
axes(handles.axes3);
delete(get(handles.axes3,'Children'));

[handles.matica3, handles.maximum3, handles.minimum3, handles.nslice3,
handles.mtrx3, handles.mtry3, handles.thk3, handles.fovx3,
handles.fovy3]=load_image_cest(hObject, handles, handles.axes3);
matica3=handles.matica3;
guidata(hObject, handles);

%% Calculation of the MTRasym values
function pushbutton4_Callback(hObject, eventdata, handles)
matica1=handles.matica1; %an image acquired at the frequency offset of the
CEST agent (+ppm)
matica2=handles.matica2; %an image acquired at the opposite frequency
offset to the CEST agent (-ppm)
matica3=handles.matica3; %an image acquired without saturation (or at the
opposite frequency offset)

%Calculation of the MTRasym values in each pixel of the image
%MTRasym = (I(-)-I(+))/I(0)*100%
global matica_cest
matica4=matica2-matica1;
matica_cest=matica4./matica3*100; %final matrix of the MTRasym map

%the minimum and maximum values of MTRasym in the final MTRasym map
minimum5=min(min(matica_cest));
maximum5=max(max(matica_cest));

%Choosing of the region of interest (ROI) mask
%maska - a global variable for the chosen ROI mask
global maska
if minimum5==maximum5
   imshow(matica_cest(:,:,1));colormap(jet); %checking whether the matrix
is empty or not (if empty, all values =1)
else
    figure; imagesc(matica_cest', [0 30]);
    maska=ones(size(matica_cest));
    maska(:,:)=roipoly; %a function for choosing of the ROI

vysledok %vysledok opens the GUI with the final MTRasym map

end
handles.matica_cest=matica_cest;
guidata(hObject, handles);

%% Calculation of the difference image
```

```
function pushbutton5_Callback(hObject, eventdata, handles)
matica1=handles.matica1; %an image acquired at the frequency offset of the
CEST agent (+ppm)
matica2=handles.matica2; %an image acquired at the opossite frequency
offset to the CEST agent (-ppm)

%subtraction of the images
global matica5
matica5=matica2(:,:)-matica1(:,:);
if minimum5==maximum5
   imshow(matica5(:,:,1));colormap(jet);
else
    vysledok %opens the GUI with the final difference image
end
handles.matica5=matica5;
guidata(hObject, handles);
```

Load_image_cest (a script written in Matlab for opening of the images)

```
%the functions loads images into the axes in the GUI
function [image, maximum, minimum, nslice, mtrx, mtry, thickness, fovx,
fovy] = load_image_cest(hObject, handles, os)
cd;
delete(get(os,'Children'));
cd(pathname)
cd ../..
path=strcat(cd);

%the path to the 2dseq folder is saved
            cestakmri=[path filesep 'pdata' filesep '1' filesep '2dseq'];
            disp(path)
%reading acqp to cells
            path2acqp=[path filesep 'acqp'];
            [acqp]=textread(path2acqp, '%s', 100000);
%loading number of slices and saving into variable nslice
            riadokspoctomvrstiev=1+strmatch('##$ACQ_read_offset=(',acqp);
            nslice=str2num(char(acqp(riadokspoctomvrstiev,:)));
%reading of slice thickness
            riadokshrubkou=strmatch('##$ACQ_slice_thick=', acqp);
            [thickness1] = strread(char(acqp(riadokshrubkou)),'%*s %4c
','delimiter','= ');
            thickness = str2num(thickness1);
%loading of field of vies
            riadoksfovx=3+strmatch('##$ACQ_fov=(', acqp);
            riadoksfovy=riadoksfovx+1;
            fovx=str2num(char(acqp(riadoksfovx,:)));
            fovy=str2num(char(acqp(riadoksfovy,:)));
%loading of reco to cells
            cd(path);
            path2reco=[path filesep 'pdata' filesep '1' filesep 'reco'];
            [reco]=textread(path2reco, '%s', 100000);
%loading of the matrix size
            radeksmtrx=3+strmatch('##$RECO_size=(',reco);
            radeksmtry=radeksmtrx+1;
            mtrx=str2num(char(reco(radeksmtrx,:)));
            mtry=str2num(char(reco(radeksmtry,:)));
%loading of number of acquisitions (nacq)
            clear pomnacq
            radeksnacq=strmatch('##$NA=',acqp);
                for inacq=7:size(acqp{radeksnacq},2)
```

```
                    pomnacq(inacq)=char(acqp{radeksnacq}(inacq));
                end;
            nacq(1)=str2num(pomnacq);
%loading of parameters "offset" and "slope"
            radeksoffset=2+strmatch('##$RECO_map_offset=(',reco);
            radekseslope=2+strmatch('##$RECO_map_slope=(',reco);
            for ivrst=1:nslice
              offset(1,ivrst)=str2num(char(reco(radeksoffset+ivrst,:)));
              slope(1,ivrst)=str2num(char(reco(radekseslope+ivrst,:)));
            end;
%reading of the images
            matica11=zeros(mtrx,mtry,nslice,'double');
            fid=fopen(cestakmri,'r', 'l');
            for i=1:nslice
                matica11(:,:,i)=fread(fid, [mtrx mtry], 'uint16', 'ieee-
le');
            end
            fclose(fid);
            matica(:,:,1)=(matica11/slope(1,1)+offset(1,1))/rg(1)/nacq(1);

%loading of the images and size control
matica1=(matica(:,:,1));
maximum=max(max(matica(:,:,1)));
minimum=min(min(matica(:,:,1)));
mri_obr=imagesc(matica1(:,:)', [minimum maximum]); colormap(gray);
set(os,'dataaspectratio',[1 1 1]);
set(os,'UserData',[mtrx mtry]);
axis off
image = matica(:,:,:);
guidata(hObject, handles);
```

Vysledok (a script written in Matlab for adjusting and saving of CEST images)

```
function varargout = vysledok(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @CEST_processing_OpeningFcn, ...
                   'gui_OutputFcn',  @CEST_processing_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function vysledok_OpeningFcn(hObject, eventdata, handles, varargin)
cd
global matica_cest
global maska

axes(handles.axes1);
handles.maska=maska;

%loading of the MTRasym map/difference image into axes1
```

```
    matica=matica_cest;
    minimum=min(min(matica))
    maximum=max(max(matica))
    mri_obr=imagesc(matica(:,:,1)', [0 30]);
    colormap(jet);
    set(mri_obr, 'Parent', handles.axes1);
    set(handles.axes1,'xtick',[],'ytick',[]);
    colorbar('peer', handles.axes1);
    ylabel(colorbar, 'CEST[%]');
    h=colorbar;
    set(h,'fontsize',20);

%the scale of the image is set from 0 to 30% (MTRasym)
handles.matica=matica;
set(handles.slider1,'min',0,'max',maximum,'value',0);
set(handles.slider2,'min',0,'max',maximum,'value',30);
handles.minimum_aktual=0;
handles.maximum_aktual=30;

handles.output = hObject;
guidata(hObject, handles);

%slider1 serves for adjustment of minimum value of the image
function slider1_Callback(hObject, eventdata, handles)
[handles.minimum_aktual]=upravjas_minimum(hObject,handles,handles.axes1,han
dles.slider1,handles.slider2,1);
guidata(hObject, handles);

%slider2 serves for adjustment of maximum value of the image
function slider2_Callback(hObject, eventdata, handles)
[handles.maximum_aktual]=upravjas_maximum(hObject,handles,handles.axes1,han
dles.slider1,handles.slider2,1);
guidata(hObject, handles);

% Choosing colormap by pop-up menu 1
Function pushbutton1_Callback(hObject, eventdata, handles)
matica=handles.matica;
minimum_actual=get(handles.slider1, 'Value');
maximum_actual=get(handles.slider2, 'Value');

colormapa1 = get(handles.popupmenu1, 'Value');
switch colormapa1
    case 1
        c1=gray;
    case 2
        c1=jet;
    case 3
        c1=hot;
    case 4
        c1=cool;
    case 5
        c1=spring;
    case 6
        c1=summer;
end
disp(c1)
handles.b = imagesc(matica', [minimum_actual maximum_actual]);
colormap(c1);
set(handles.axes1,'xtick',[],'ytick',[]);
colorbar('peer', handles.axes1);
```

```
ylabel(colorbar, 'CEST[%]');
guidata(hObject, handles);

%deleting the background outside the chosen ROI
function pushbutton5_Callback(hObject, eventdata, handles)
handles.matica=handles.matica.*handles.maska';
mri_obr=imagesc(handles.matica(:,:,1)', [handles.minimum_aktual
handles.maximum_aktual]);
colormap(jet);
    set(mri_obr, 'Parent', handles.axes1);
    set(handles.axes1,'xtick',[],'ytick',[]);
    colorbar('peer', handles.axes1);
    ylabel(colorbar, 'CEST[%]');
    h=colorbar;
    set(h,'fontsize',20);
guidata(hObject, handles);

%saving of the final image
function pushbutton2_Callback(hObject, eventdata, handles)
imageData = screencapture(gcf, [42 160 392 365]);
figure, imshow(imageData);

[filename, pathname, filterindex] = uiputfile( ...
       {'*.png','PNG Images (*.png)'; ...
       '*.*',  'All Files (*.*)'}, ...
        'Save as', 'vysledok.png');
    imwrite((imageData),[pathname filename],'png')

saveas( handles.axes1 , 'mysavedfig.jpg' )
guidata(hObject, handles);
```

Appendix B

Gálisová A et al. *Magnetic resonance visualization of pancreatic islets labeled by PARACEST contrast agents at 4.7 T. Molecular Imaging and Dynamics 2016; 6:121*

The original article is a part of the printed version of the thesis and it can be found online: https://www.omicsonline.org/peer-reviewed/magnetic-resonance-visualization-of-pancreatic-islets-labeled-by-paracest-contrast-agents-at-47-t-68387.html

Appendix C

Krchová T, Gálisová A et al. *Ln(III)-complexes of a DOTA analogue with an ethylenediamine pendant arm as pH-responsive PARACEST contrast agents. Dalton Transactions 2016; 45(8):3486-96*

The original article is a part of the printed version of the thesis and it can be found online:
http://pubs.rsc.org/en/content/articlelanding/2016/dt/c5dt04443j#!divAbstract

Appendix D

Gálisová A et al. *Multimodal imaging reveals improvement of blood supply to an artificial cell transplant site induced by bioluminescent mesenchymal stem cells. Molecular Imaging and Biology 2017; 19(1):15-23*

The original article is a part of the printed version of the thesis and it can be found online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5209399/

Appendix E

Gálisová A et al. *The optimal timing for pancreatic islet transplantation into subcutaneous scaffolds assessed by multimodal imaging. Contrast Media and Molecular Imaging 2017. Article ID 5418495*

The original article is a part of the printed version of the thesis and it can be found online: https://www.hindawi.com/journals/cmmi/2017/5418495/

Appendix F

Fábryová E, Jirak D, Girman P, Zacharova K, **Gálisová A et al.** *Effect of Mesenchymal Stem Cells on the Vascularization of the Artificial Site for Islet Transplantation in Rats. Transplantation Proceeding 2014; 46:1963-1966*

The original article is a part of the printed version of the thesis and it can be found online: https://www.sciencedirect.com/science/article/pii/S0041134514004229

Appendix G

Herynek V, Gálisová A et al. *Pre-Microporation Improves Outcome of Pancreatic Islet Labeling for Optical and ^{19}F MR Imaging. Biological Procedures Online. 2017; 19:6*

The original article is a part of the printed version of the thesis and it can be found online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5488379/