Supplement C

Outcome measures of the *in vivo* experiments

The following code describes the calculations that were performed to compute the absolute and relative changes in anal pressure of the *in vivo* experiments.

MATLAB code

```
% Written in MATLAB R2018b
% Edited in MATLAB R2020a

function [s_in] = calcOutcome(s,int_names,fs,tx)
%calcOutcome Calculates desired outcomes from selected windows and data
%   Input/output variables:
%   - s: structure array containing information of all selected files
%   - int_names: interval names of interest
%   - fs: sampling frequency
%   - tx: time of which the inhib pressure slope is calculated, in seconds
%   - s_in: modified structure array containing information of all selected
files

s_in = s;   % work with copy of input array
tx(tx==0) = 1; % if tx=0, this value will later cause an error (cannot divide
by 0). So if input tx equals 0, tx will be changed to 1.
slope_fieldname = sprintf('inhib_slope_for_%usec', tx);  % fieldname of slope
that includes the amount of seconds it is calculated from


 for iRow = 1:numel(s_in)    % loop over rows, i.e. selected files
        pressure = s(iRow).pres_adj;    % copy values, but with a simpler name

     %% start/end point index definition
     %---(beware this is the index in the pressure vector, not the
     %corresponding time value in seconds) ---%

       %baseline_avg
        start_base = s_in(iRow).baseline(1) * fs;
        end_base = s_in(iRow).baseline(2) * fs;

       %stimulation_avg
        start_stim = round(s_in(iRow).stimulation(1) * fs);

        if isnan(s_in(iRow).inhibition) % in case of absent inhibition
            end_stim = round(s_in(iRow).stimulation(2) * fs);    % average
stimulation pressure must be calculated from whole stim interval

        else                            % in case of present inhibition
```

```
            end_stim = round(s_in(iRow).inhibition(1) * fs);    % average
stimulation pressure must be calculated from interval start stim to start
inhib

        %inhibition_avg
            start_inhib = round(s_in(iRow).inhibition(1) * fs);
            end_inhib = round(s_in(iRow).inhibition(2) * fs);

        %relative pressure drop
            a = max(pressure(start_stim : end_stim)) -
mean(pressure(start_base : end_base));
            b = pressure(end_inhib) - mean(pressure(start_base : end_base));

        %pressure_drop_slope
            start_slope = round(start_inhib);
            end_slope = round((s_in(iRow).inhibition(1) + tx) * fs);

        %Zoom in on data and set start- and endpoint of onset
            pressure_window = pressure((start_inhib-4*fs):(start_inhib+4*fs));
% select a smaller window of the data for faster analysis of onset, start at
stimulation and end at inhibition + 4 sec's
            pressure_mean = movmean(pressure_window,500);                    %
create a movingmean vector of the pressure data with a window of 500 samples
            changepts = findchangepts(pressure_mean,'Statistic', 'linear',
'MaxNumChanges',20,'MinDistance',50); % find 20 changing points in slope and
average with a minimal distance of 50 samples inbetween
            changepts = changepts + (start_inhib-4*fs)-1;
% add the start_simulation to rewrite from window to full data set
            start_onset = dsearchn(changepts,start_inhib);                   %
find the point in changepts closest to the start_inhib value
            end_onset = dsearchn(changepts,
(find(pressure_window==max(pressure_window))+(start_inhib-4*fs)-1)); % find
the point in changepts closest to the maximum value
            mean_baseline = mean(pressure(start_base:end_base));             %
take the mean pressure of the baseline phase
        end


      %% calculation of outcome values
        s_in(iRow).baseline_avg = mean(pressure(start_base : end_base));
        s_in(iRow).stimulation_avg = mean(pressure(start_stim : end_stim));
        s_in(iRow).stimulation_startvalue = pressure(start_stim);
        s_in(iRow).stimulation_peak = max(pressure(start_stim : end_stim)); %
this does include the bias by fluctuating pressure values (even after low pass
filter is applied)

        if isnan(s_in(iRow).inhibition) % in case of absent inhibition
            s_in(iRow).inhibition_avg = NaN;
            s_in(iRow).inhibition_startvalue = NaN;
            s_in(iRow).inhibition_endvalue = NaN;
            s_in(iRow).rel_pressure_drop = NaN;
            s_in(iRow).(slope_fieldname) = NaN;
            s_in(iRow).onset_height = NaN;
            s_in(iRow).onset_increase = NaN;
        else                            % in case of present inhibition
```

```
            s_in(iRow).inhibition_avg = mean(pressure(start_inhib :
end_inhib));
            s_in(iRow).inhibition_startvalue = pressure(start_inhib);
            s_in(iRow).inhibition_endvalue = pressure(end_inhib);
%             s_in(iRow).rel_pressure_drop = (s_in(iRow).inhibition_avg -
s_in(iRow).stimulation_avg) ...
%                                              / (s_in(iRow).stimulation_avg -
s_in(iRow).baseline_avg ) * 100;
            s_in(iRow).rel_pressure_drop = ( b - a ) / a * 100;
            s_in(iRow).(slope_fieldname) = (pressure(end_slope)-
pressure(start_slope)) / tx;

            if end_onset-start_onset > 0        % in case of present onset
                if (pressure(changepts(end_onset))-mean_baseline)-
(pressure(changepts(start_onset))-mean_baseline)> 0
                    s_in(iRow).onset_height = (pressure(changepts(end_onset))-
mean_baseline)-(pressure(changepts(start_onset))-mean_baseline); %remove
baseline pressure and calculate the difference between start and end onset
                    s_in(iRow).onset_increase =
(pressure(changepts(end_onset))-
mean_baseline)/(pressure(changepts(start_onset))-mean_baseline); % remove
baseline pressure and calculate the increase percentage of the pressure
                else
                    s_in(iRow).onset_height = NaN;
                    s_in(iRow).onset_increase = NaN;
                end
            else                                % in case of absent onset
                s_in(iRow).onset_height = NaN;
                s_in(iRow).onset_increase = NaN;
            end

       end



  end
```