% Patch clamp traces analyzed by Clampex are capped at eight channels, with

% all events involving more than eight channels being reassigned to i = 8.

% This script extrapolates from Clampex data to assign the correct

% activation levels to each event. The file input is an Excel document

% containing event analysis files as provided by Clampex and described in

% Methods.

[status,sheets] = xlsfinfo('final.xlsx');

finalsheetaverages = zeros(100,length(sheets));

finalsheetnumbers = zeros(100,length(sheets));

% Read each tab in Excel, and extract assigned event level

% (event\_amplitudes), dwell time (event\_durations) and current amplitude

% (event\_amptrue) into an array. These are typically on the third, ninth

% and seventh columns of the Clampex file.

for primary = 1:length(sheets);

try

F = readtable('final.xlsx','Sheet',primary);

event\_amplitudes = table2array(F(:,3));

event\_durations = table2array(F(:,9));

event\_amptrue = table2array(F(:,7));

event\_array = [event\_amplitudes event\_durations event\_amptrue];

% For i = 0 through 7, find the average current amplitude occurring at

% this level as determined by Clampex. Use this value to calculate the

% current amplitude for a single channel opening event.

for counter\_a = 0:min(max(event\_amplitudes),7);

indices = find(event\_amplitudes==counter\_a);

meanamp(1,counter\_a+1) = mean(event\_amptrue(indices));

end

for counter\_b = 1:length(meanamp)-1;

meanamptrue(1,counter\_b) = (meanamp(1,counter\_b+1) - meanamp(1,1))/counter\_b;

end

amplitude\_equalizer = mean(meanamptrue);

% An alternative method to calculate single-channel current using events

% at i = 5, 6 and 7. This can be more accurate due to the noise present

% at lower levels.

% if max(event\_amplitudes) == 8;

% amplitude\_equalizer = mean(meanamptrue(5:7));

% end

% Check the entire array for events assigned to i = 8 in Clampex, and

% reassign them using the single-channel current amplitude calculated

% previously.

for counter\_c = length(event\_array):-1:1;

if event\_array(counter\_c,1) == 8;

event\_array(counter\_c,1) = round(event\_array(counter\_c,3)./amplitude\_equalizer);

end

end

% Delete any event lasting more than 0.05 ms. These 'events' are almost

% invariably caused by noise.

for counter\_d = length(event\_array):-1:1;

if event\_array(counter\_d,2)<0.05;

event\_array(counter\_d,1)=event\_array(counter\_d-1,1);

end

end

for hm = length(event\_array):-1:2;

if event\_array(hm,1) == event\_array(hm-1,1);

event\_array(hm-1,2) = event\_array(hm-1,2) + event\_array(hm,2);

event\_array(hm,:) = [];

end;

end;

% Optional code for deleting a level if there are less events in it than a

% specified filter value, as these events may be misassigned and not

% represent the true number of channels (n) in the patch.

% maximum\_amplitude = max(event\_amplitudes);

% for i = 0:maximum\_amplitude

% filter = sum(event\_amplitudes(:) == i);

% if filter < 10

% event\_array(any(event\_array==i,2),:) = [];

% end

% end

% Check the entire array for the number, total time spent at level, and

% average dwell time of events at each i. Assign these to an internal

% array, then move this array to the final data sheet containing dwell time

% and event number information for each trace.

event\_amplitudes = event\_array(:,1);

event\_durations = event\_array(:,2);

maximum\_amplitude\_new = max(event\_amplitudes);

array\_of\_dwell\_times = zeros(maximum\_amplitude\_new+1,1);

array\_of\_event\_numbers = zeros(maximum\_amplitude\_new+1,1);

array\_of\_averages = zeros(maximum\_amplitude\_new+1,1);

for q = 0:maximum\_amplitude\_new;

indices = find(event\_array(:,1)==q);

event\_filtered = event\_array(indices,2);

array\_of\_dwell\_times(q+1,1) = nansum(event\_filtered(:,1));

end

for d = 0:maximum\_amplitude\_new;

number\_of\_events\_at\_level = sum(event\_amplitudes(:) == d);

array\_of\_event\_numbers(d+1,1) = number\_of\_events\_at\_level;

array\_of\_averages(d+1,1) = array\_of\_dwell\_times(d+1,1)./number\_of\_events\_at\_level;

end

final = zeros(100,2);

final2 = zeros(maximum\_amplitude\_new+1,2);

corrector = 100 - length(array\_of\_averages);

correctorarray = zeros(corrector,1);

array\_of\_event\_numbers = vertcat(array\_of\_event\_numbers,correctorarray);

array\_of\_averages = vertcat(array\_of\_averages,correctorarray);

finalsheetaverages(:,primary) = array\_of\_averages;

finalsheetnumbers(:,primary) = array\_of\_event\_numbers;

clearvars -except primary finalsheetaverages finalsheetnumbers status sheets

fprintf("Completed sheet %d\n\r",primary)

end

end