Neuron, Volume *109*

Supplemental information

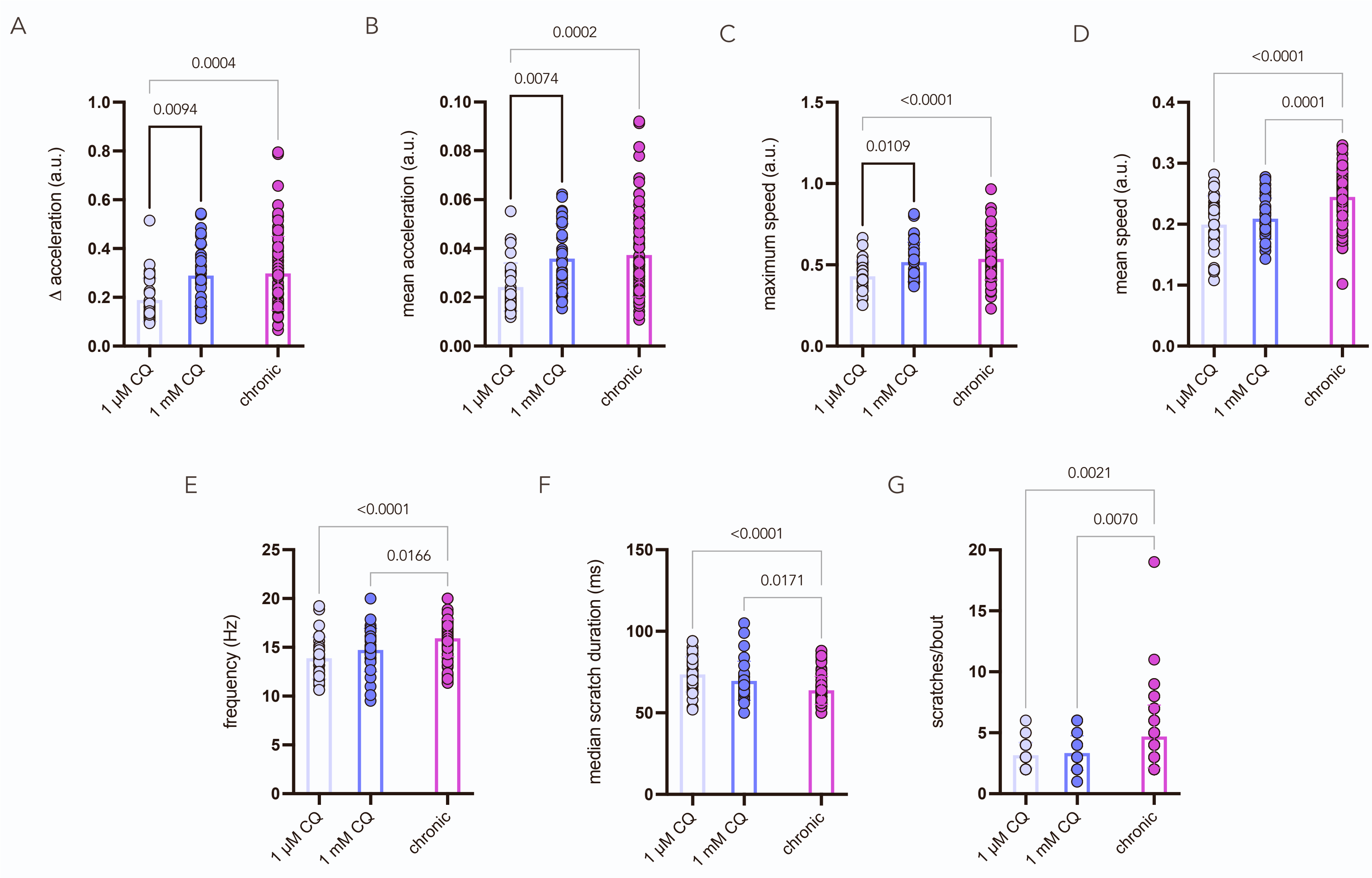
Dissecting the precise nature

of itch-evoked scratching

Nivanthika K. Wimalasena, George Milner, Ricardo Silva, Cliff Vuong, Zihe Zhang, Diana M. Bautista, and Clifford J. Woolf

Supplementary Figure 1: Comparison of scratching parameters between CQ-induced and chronic neck scratching, related to Figure 5 and 7

Scratching by individual bout data from Figure 4 (n = 32 bouts for each CQ dose) are replotted alongside chronic neck scratching data from Figure 6 (n = 83 bouts). p-values were calculated using one-way ANOVA; non-significant p-values are not shown. Statistical comparisons between 1 μM and 1 mM are bolded, whereas comparisons between CQ groups and chronic scratching are shown in gray. A) Maximum speed is higher in both the 1 mM CQ and chronic scratching groups compared to 1 μM CQ ($p = 0.0109$, $p < 0.0001$, respectively) B) Mean speed is higher in chronic scratching compared to both 1 μM and 1 mM CQ groups ($p < 0.0001$, $p = 0.0001$, respectively). C) Delta acceleration, representing the difference between the maximum and minimum acceleration, is higher in both the 1 mM CQ and chronic scratching groups compared to 1 μM CQ ($p = 0.0094$, $p = 0.0004$, respectively). D) Mean acceleration is higher in both the 1 mM CQ and chronic scratching groups compared to 1 μM CQ ($p = 0.0074$, $p = 0.0002$, respectively). E) Scratching frequency in Hz is higher in chronic scratching compared to both 1 μM and 1 mM CQ groups ($p < 0.0001$, $p = 0.0166$, respectively). F) Median scratch duration in ms; calculated using the interpeak interval, is higher in chronic scratching compared to both 1 μM and 1 mM CQ groups ($p < 0.0001$, $p = 0.0171$, respectively). G) Number of scratches per bout is higher in chronic scratching compared to both 1 μM and 1 mM CQ groups ($p = 0.0021$, $p = 0.007$, respectively).



A
Δ acceleration (a.u.)
0.0094
0.0004
B
mean acceleration (a.u.)
0.0074
0.0002
C
maximum speed (a.u.)
0.0109
<0.0001
D
mean speed (a.u.)
<0.0001
0.0001
E
frequency (Hz)
<0.0001
0.0166
F
median scratch duration (ms)
<0.0001
0.0171
G
scratches/bout
0.0021
0.0070
1 μM CQ
1 mM CQ
chronic

Supplementary Video Legends

Supplementary Video 1. High-speed video of scratching in response to 1 μM CQ, related to Figure 3.

CQ was injected into the nape of the neck. Video recorded at 500 fps. This video corresponds to Figure **3C, E, G, I.**

Supplementary Video 2. High-speed video of scratching in response to 1 mM CQ, related to Figure 3.

CQ was injected into the nape of the neck. Video recorded at 500 fps. This video corresponds to Figure **3D, F, H, J.**

Supplementary Video 3. High-speed video of an instance of neck scratching, related to Figure 4.

Video recorded at 500 fps. This video corresponds to Figure **4A-I**. The animal scratches the neck four times before licking the hind paw. Forceful contact with the skin is apparent in the second and fourth scratches.

Supplementary Video 4. High-speed video of an instance of face scratching, related to Figure 4.

Video recorded at 500 fps. This video corresponds to Figure **4J-R**. The animal scratches the face 19 times before licking the hind paw.

```
function [max_s, mean_s,median_s,a_diff, mean_a, median_a, frequencyHz, Number_of_peaks,↙
median_interval] = statistics(data_path,filename)
%% Extract, filter, and smooth trajectories
data = readtable(data_path);
x = data(:,1);
y = data(:,2);
x = table2array(x);
y = table2array(y);
y = 1000 - y; %the y axis values from image_J run from top to bottom so need to be↙
flipped
x_filt= medfilt1(x); %median filter, 1 dimensional, 3rd order
y_filt = medfilt1(y); %median filter, 1 dimensional, 3rd order
x_filt = smoothdata(x_filt,'gaussian',5); %gaussian smoothing
y_filt = smoothdata(y_filt,'gaussian',10); %gaussian smoothing
x_norm = zscore(x_filt);
y_norm = zscore(y_filt);


%% Speed calculations
delta_x= diff(x_norm); %differential of x_norm
delta_y= diff(y_norm); %differential of y_norm
s = ((delta_x.^2 + delta_y.^2).^0.5); %pythagoras calculating speed

max_s = max(s); %maximum speed
min_s = min(s); %minimum speed
mean_s = mean(s); %mean speed
median_s = median(s); %median speed

%% Acceleration calculations
accel = diff(s); %differential of speed to calculate acceleration
accel_time = size(accel);

max_a = max(accel); %maxiumum acceleration
min_a = min(accel); %minimum acceleration
a_diff = max_a - min_a; %delta acceleration, accounting for positive and negative values

mean_a = mean(abs(accel)); %mean of the absolute value of acceleration
median_a = median(abs(accel)); %median of the absolute value of acceleration


%% Gradient colour plot
% to generate plot of x,y trajectory overlayed with speed represented as a color gradient
totalframes = size(s);
x_norm(totalframes(1),:) = [];
y_norm(totalframes(1),:) = [];
figure; scatter(x_norm,y_norm, 75, s, 'filled');
ylim ([-2.5 2])
xlim ([-3 3.5])
hold on
plot(x_norm,y_norm,'k',...
    'LineWidth',0.01);
c = colorbar;
set(c, 'ylim', [0 0.8])
xlabel('normalized x position')
ylabel('normalized y position')
```

```
%% Plotting speed and acceleration overtime

figure;plot(s); %plotting speed
xlabel('time (frame number)');
ylabel('speed (a.u.)')

figure;plot(accel); %plotting acceleration
xlabel('time (frame number)');
ylabel('acceleration (a.u.)')



%% Peak and trough analysis on y values

%peaks
PeakCutoff = max_y*0.7; %the cutoff for what is defined as a peak is taken as 70% of the↙
maximum speed
hi= numel(findpeaks(y_norm)); %How many peaks are there overall in the curve
Number_of_peaks = numel(findpeaks(y_norm,'MinPeakDistance',3,'MinPeakHeight',↙
PeakCutoff,'MinPeakProminence', 0.05)); % this counts peaks in the speed graph that reach↙
the threshold for height, are 6 or more points away from each other)
[peaks, x_peak] = (findpeaks(y_norm,'MinPeakDistance',3,'MinPeakHeight',↙
PeakCutoff,'MinPeakProminence', 0.05)); %generates x and y values for peaks in a matrix
Peaks = [peaks, x_peak];


%troughs
y_norminverted = -y_norm; %invert the data and find peaks in order to find troughs
max_trough = max(y_norminverted); %identify maximum trough value to calculate trough↙
cutoff
TroughCutoff = max_trough*-1.2;
[troughs, x_trough] = findpeaks(y_norminverted,'MinPeakDistance',3,'MinPeakHeight',↙
TroughCutoff,'MinPeakProminence',0.05); %criteria for troughs
Troughs = [-troughs, x_trough];


%% Plot peaks on y_norm

frames = (1:totalframes(1));
time = frames*2; %given 500 frames per second framerate
x_peak_time = x_peak*2;
x_trough_time = x_trough*2;
figure;plot(y_norm); %plot y values over time

hold
plot(x_peak,peaks,'o') %denote peaks
plot(x_trough,-troughs,'o') %denote troughs
xlabel('time (frame number)')
ylabel('normalized y position')

%% Caluclate interval and frequncy

median_interval = median(diff(x_trough)); %the median distance between troughs; useful to↙
use median if some peaks are not detected
frequencyHz = 1000/(median_interval*2); %frequency calulated based on the median interval
```

```
%% File open loop

    files = dir ('/Users/Nivanthika/Desktop/Matlab/data/chronic_scratch only/*.csv');
    folder_location = ('/Users/Nivanthika/Desktop/Matlab/data/chronic_scratch only/');
    N = length(files);
    a = [ ];
    names = string.empty(0,N);

for i = 1:N
    filename = files(i).name;
    file_location = strcat(folder_location,filename);
    [max_s, mean_s,median_s, a_diff, mean_a, median_a, frequencyHz, Number_of_peaks,↙
median_interval] = statistics(file_location,filename);
  a(:,i) = [max_s, mean_s,median_s, a_diff, mean_a, median_a, frequencyHz,↙
Number_of_peaks, median_interval];
  filename;
  names(i) = filename;
  close all
end

split_name = split(names, '.');
new_name = split_name(:,:,1);
New_name = string(new_name);
RowNames = {'Max Speed', 'Mean Speed','Median Speed','Accel Diff','Mean Accel', 'Median↙
Accel', 'Frequency (Hz)', 'Number of Peaks', 'Median Interpeak Interval'};
Names = cellstr(names);
Comparison = array2table(a, 'RowNames', RowNames);
```