INTRODUCTION

Alzheimer's disease (AD) is a neurodegenerative disease that progressively destroys brain tissue, leading to irreversible loss of brain mass manifesting in impaired memory and spatial cognition among other things. This report aims to outline the results obtained from completing a Morris water maze experiment on wild-type (WT) mice and transgenic mouse model AD (TG), testing their spatial memory capabilities. Our analysis have focused primarily on the cummulative distance travelled by each mouse in order to reach a hidden platform in the maze; evaluating how the distance travelled differed from WT and TG mice as well as assessing whether any differences were statistically significant.

RESULTS

Data analysis methods

Our data set consists of 16 different video recordings from 16 different trials, each with a different number of frames. In order to detect the mouse, we have binarised the video frames so the mouse appears as a white blob on a black background. Using Otsu's method to determine the binary threshold did not yield a satisfactorily binarised image, therefore we have manually determined the threshold by looking at the video (frame)'s image histogram and setting the threshold as the pixel intensity that roughly separates the two bimodal curves in the histogram. We then normalised this value by dividing it by 255, as the video recordings were determined to be 8 bit.

To evaluate the cummulative distance each mouse travelled in each recording, we have determined the XY pixel coordinates (set as the centre of the mouse blob) in each frame of the recording, and then calculating the distance between the coordinates of each successive frames using Pythagoras' theorem. To convert this pixel distance into metres, we have multiplied the pixel distances by 0.0059, which was calculated by dividing 1 by the width of the video frames in pixels (170).

To assess whether any differences in cummulative distance between WT and TG mice were statistically significant, we first assessed the normality of the data using the Jarque-Bera (JB) test. In our function, the two-sample t-test would run if the data was normaly distributed, and a Mann-Whitney test if the data was not. JB test analysis resulted in a negative result, so the non-parametric Mann-Whitney test was used.

TG mice showed higher cummulative distance travelled than WT mice

As seen in Fig 1., the TG mice travelled a longer cummulative distance to locate the hidden platform in the water maze compared to WT mice (see Fig 2. for an example swim path of a WT and TG mouse). The TG mice had an average cummulative distance of 6.0 m compared to the WT mice's 1.03 m. The TG mice also showed higher inter-individual variability in distance travelled, with a cummulative distance range of 11.89 m compared to the WT mice's range of 0.88 m (see Fig 1).

As seen in the mouse swim-path Figure 2., WT mice appear to remember exactly where the hidden platform is and is able to swim directly to it upon being placed in the maze, whereas it

takes longer for TG mice to locate the platform, characterised by its more wandering swim-path.

Statistical analysis using the Mann-Whitney test resulted in a p-value of 0.0030, indicating that there is strong evidence to suggest that the differences in cummulative distance travelled between WT and TG mice were statistically significant.

SUMMARY CONCLUSION

In conclusion, TG mice travelled a longer cummulative distance to find the water maze's hidden platform, and hence performed significantly worse than WT mice. The result of this evaluation is likely due to the fact that the TG mice had impaired spatial memory due to AD, requiring more time to locate the hidden platform and thus increasing its cummulative distance travelled in the maze. Compare this to the WT mice, who did not have AD and so had little problem memorising the location of the platform, leading to better performance in the water maze. These results provide evidence that AD can lead to impairments in spatial memory.

FIGURES

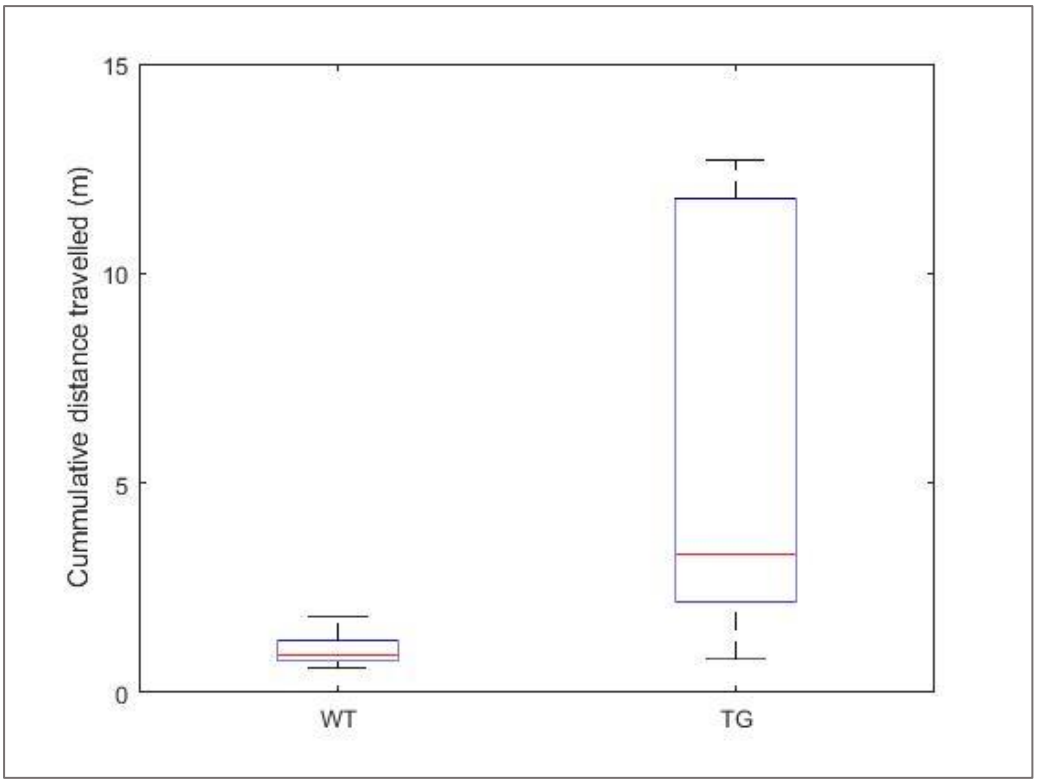

Figure 1: Box plot showing the range of cummulative distances travelled by WT and TG mice (in metres). WT mice had an average cummulative distance of 1.03 m with a min distance of 0.60 m and a max distance of 1.48 m (range = 0.88). TG mice had an average cummulative distance of 60.0 m with a min distance of 0.82 m and a max distance of 12.71 m (range = 11.89).

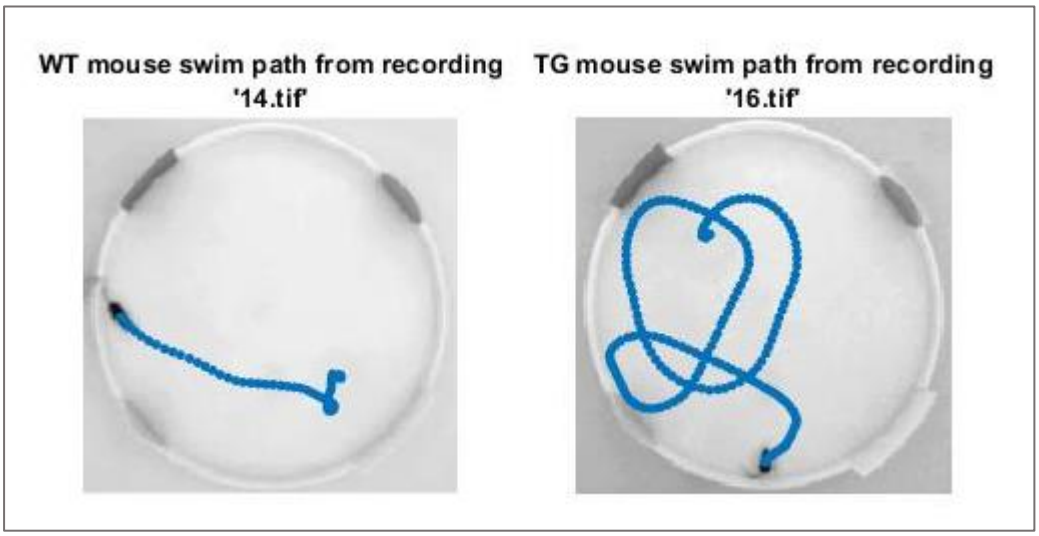

Figure 2: Multi-panel figure showing the swim-path travelled to locate a hidden platform by an example WT mouse (right) and an example TG mouse (left) from recordings 14 and 16 respectively. The swim-path is plotted as blue dots corresponding to the position (XY pixel coordinates) of the centre of the mouse in each frame of the recording with a total of 81 frames in recording 14 and 246 frames in recording 16. The dark coloured blob represents the mouse at the start of the recording.

CODE APPENDIX

```
function [Out]=S1_690056125(dir)
```

```
%INTRODUCTION
%This function returns 3 things:

%1: A struct output containing 2 fields (WT and TG) each containing a
%9x3 cell containing the filename of each recording of the WT/TG mice
%and their corresponsing cummulative distance travelled for the recording
%as well as XY pixel coordinates for the mouse in every frame

%2: A boxplot summarising the range of cummulative distances WT and TG
%travelled throughout the recordings

%3: A multipanel figure showing the swim path of two example WT and TG mice
%(from recordings 14 and 16)

%This function also runs a statistical test to analyze whether there is
%any statistical difference in distance travelled between WT and TG mice.
%If the test finds a significant difference, it will display a character
%array stating so with the associated p-value.

%FUNCTION INPUT/OUTPUT
%Input(dir):a character array of the file directory containing the
%blinding_list excel notebook and all .tif mouse recordings

%Output[Out]:a struct as explained above
```

```
%Setting current directory from input
cd(dir);

%Reading blinding_list table and dividing recordings into WT and TG
tab=readtable('blinding_list.xlsx');
newtab=table(tab.FileName(ismember(tab.Genotype,'WT')),...
    tab.FileName(ismember(tab.Genotype,'TG')));
newtab.Properties.VariableNames={'WT','TG'};

%Finding number of recordings for each mouse genotype
numrec=height(newtab);
```

```
%PREALLOCATING VALUES FOR THE BATCH PROCESSING LOOP

WT=[]; %cell array to contain all video frames from all 8 WT mouse recordings
TG=[]; %cell array to contain all video frames from all 8 TG mouse recordings

Frames=zeros(numrec,2); %array to contain the number of frames for
                        %each WT mouse recording (1st column) and
                        %each TG mouse recording (2nd column)

WTIN=[]; %cell array to contain WT mouse video frames with inverted colours
TGIN=[]; %cell array to contain TG mouse video frames with inverted colours

threshold=200/255; %threshold for the imbinarize function

WTBW=[]; %cell array to contain binarized frames from WT recordings
TGBW=[]; %cell array to contain binarized frames from TG recordings
```

```
WTCor=[]; %cell array to contain XY pixel coordinates of WT mouse
          %in each frame in each recording
TGCor=[]; %cell array to contain XY pixel coordinates of TG mouse
          %in each frame in each recording

WTDist=[]; %cell array to contain pixel distance moved by WT mouse between
           %each frame in each recording
TGDist=[]; %cell array to contain pixel distance moved by TG mouse between
           %each frame in each recording

distances=zeros(numrec,2); %array to contain cummulative pixel distance
                      %travelled by WT mice (1st column) and
                      %TG mice (2nd column)in each recording
```

```
%BATCH PROCESSING VIDEO FILES USING FOR LOOP

%Reading image files using vid subfunction
for n=1:numrec
```

```
    WT{n}=vid(newtab.WT{n});
    TG{n}=vid(newtab.TG{n});

%Finding the number of frames in each  WT and TG recording
    Frames(n,1)=size(WT{n},3);
    Frames(n,2)=size(TG{n},3);

%Inversing image colour (white to black, black to white)
    WTIN{n}=imcomplement(WT{n});
    TGIN{n}=imcomplement(TG{n});

%Filtering/binarizing frames to black & white so mice appear as white blob
%on a black background
    WTBW{n}=imbinarize(WTIN{n},threshold);
    TGBW{n}=imbinarize(TGIN{n},threshold);
```

```
%Finding mouse XY pixel coordinates in each frame
%Coordinates set as the centre of the white blob (the mouse)

corwt=[];
for m=1:Frames(n,1)
    corwt{m}=regionprops(WTBW{n}(:,:,m),'Centroid');
end

cortg=[];
for p=1:Frames(n,2)
    cortg{p}=regionprops(TGBW{n}(:,:,p),'Centroid');
end

WTCor{n}=corwt;
TGCor{n}=cortg;
```

```
%Finding distance travelled between each frame using norm function
%to calculate Eucledian distance between 2 XY coordinates using Pythagoras'
%theorem

diswt=[];
for k=1:(numel(WTCor{n})-1)
    diswt{k}=norm(WTCor{n}{0+k}(1).Centroid-WTCor{n}{1+k}(1).Centroid);
end
```

```
distg=[];
for j=1:(numel(TGCor{n})-1)
    distg{j}=norm(TGCor{n}{0+j}(1).Centroid-TGCor{n}{1+j}(1).Centroid);
end

WTDist{n}=diswt;
TGDist{n}=distg;
```

```
%Finding cummulative distance travelled for mice in each recording
distances(n,1)=sum(cell2mat(WTDist{n})); %distance for WT mice
distances(n,2)=sum(cell2mat(TGDist{n})); %distance for TG mice
```

```
end

%%Converting pixel distance to metres
px=1/width(WTBW{1}); %1 pixel = 1/170 (0.0058...) metres
distancesmetre=distances.*px; %new matrix containing distance travelled in m
```

```
%CREATING BOX PLOT TO SUMMARISE DISTANCE TRAVELLED BETWEEN WT AND TG
figure
boxplot(distancesmetre,'Labels',{'WT','TG'})
ylabel('Cummulative distance travelled (m)')
ylim([0 15])
```

```
%STATISTICAL TEST
%Assesing the normality of the data using Jarque-Bera test
norm1=jbtest(distancesmetre(:,1)); %assesing normality for WT data
norm2=jbtest(distancesmetre(:,2)); %assesing normality for TG data

%Code will run a two-sample t-test if both norm1 & norm2 are normally
%distributed (ie. = 1) and will run a Mann-Whitney ranksum test otherwise

if norm1==1 && norm2==1
    [~,p]=ttest2(distancesmetre(:,1),distancesmetre(:,2));
else
    p=ranksum(distancesmetre(:,1),distancesmetre(:,2));
end


if p<0.05
    disp(['There is evidence showing that there is a difference in'...
        ' cummulative distance travelled between WT and TG mice'...
        ' with a p-value of ' num2str(p)])
else
    disp(['There is no evidence showing that there is a difference in'...
        ' cummulative distance travelled between WT and TG mice'])
end
```

```
%CREATING THE STRUCT OUTPUT

tt=table2cell(newtab);
Out.WT=cell(tt(1:end,1)); %1st column in field contains filenames of each
                          %recording
Out.TG=cell(tt(1:end,2));


for n=1:numrec
    Out.WT{n,2}=distancesmetre(n,1); %2nd column contains cummulative
```

```
                                    %distance travelled
    Out.WT{n,3}=WTCor{n}; %3rd column contains XY coordinates of mouse
                          %in each frame

    Out.TG{n,2}=distancesmetre(n,2);
    Out.TG{n,3}=TGCor{n};
end

%Creating headers in each struct fields
header={'Filename','Cummulative distance travelled (m)',...
    'Mouse XY pixel coordinates per frame'};
Out.WT=[header;Out.WT];
Out.TG=[header;Out.TG];
%headers show as the first row in each field
```

```
%CREATING MULTIPANEL FIGURE SHOWING BEHAVIOUR OF A WT AND TG MOUSE
%WT mouse from recording 14 and TG mouse from recording 16

figure
ax1=subplot(1,2,1);imshow(WT{8}(:,:,1));
hold (ax1,"on")
for a=1:81 %number of frames in recording 14
plot(corwt{a}.Centroid(1),corwt{a}.Centroid(2),'o','Color','#0072BD',...
    'MarkerFaceColor','#0072BD',"MarkerSize",3)
end
hold (ax1,"off")

ax2=subplot(1,2,2);imshow(TG{8}(:,:,1));
hold (ax2,"on")
for b=1:246 %number of frames in recording 16
    plot(cortg{b}.Centroid(1),cortg{b}.Centroid(2),'o','Color',...
        '#0072BD','MarkerFaceColor','#0072BD',"MarkerSize",3)
end


title(ax1,["WT mouse swim path from recording"; "'14.tif'"])
title(ax2,["TG mouse swim path from recording"; "'16.tif'"])
```

```
%VID SUBFUNCTION
%Info: This function outputs a 3D matrix containing data for each video frame
%using the filename of the video as the input

    function [video]=vid(filename)
        vdata=imfinfo(filename);
        numframes=length(vdata);

        video=[];
        for x=1:numframes
            v=imread(filename,x);
            video=cat(3,video,v);
        end
    end
```

```
end
```

Published with MATLAB® R2020b