Environmental enrichment and epigallocatequingallate normalize aberrant neural activity in a murine model of Down Syndrome

Marta Vilademunt Alcaide
Final Degree Project
Human Biology
Pompeu Fabra University
2018
INDEX

0. Abstract and keywords
1. Introduction ........................................................................................................3
2. Hypothesis and Objectives ..............................................................................8
3. Methodology .......................................................................................................9
4. Results ................................................................................................................13
5. Discussion ..........................................................................................................15
6. Conclusions ........................................................................................................17
7. Acknowledgements ..........................................................................................17
8. Tables and Figures ............................................................................................18
9. Bibliography .......................................................................................................22
10. Appendices: Supplementary data

Abstract:
Down syndrome (DS) is the most common genetic disease underlying intellectual disability. It is characterized by memory impairment and brain morphological and molecular alterations resulting in poor executive performance. Different mouse models have been developed to understand DS, Ts65Dn being the best characterized. Even though a lot is known about genetic, morphological and molecular abnormalities in this model, neural activity alterations are poorly understood. It has been described that environmental enrichment (EE) and the green tea extract epigallocatequine-3-gallate (EGCG) rescue cognitive impairment in Ts65Dn mice; however, the cellular mechanisms underlying their therapeutic action are not well understood. The present work reports the presence of an aberrant prefrontal cortex and hippocampal multi-unit activity (MUA) in Ts65Dn mice when compared to wild-type non-trisomic mice. The results show that EE and EGCG, when administered in combination, normalize this aberrant neural activity of trisomic mice, although, statistically significant differences were only reported in the prefrontal cortex. Therefore, additional data needs to be analysed to enhance statistical robustness and demonstrate further this effect also in the hippocampus. Aside its limitations, this work provides novel and valuable neurophysiological information that may be used to understand DS cognitive normalization both in mouse models and humans.

Keywords:
Down syndrome, Ts65Dn, multi-unit activity, environmental enrichment, epigallocatequine-3-gallate
1. **Introduction**

The main purpose of this project is to understand brain activity patterns in a murine model of Down syndrome and how they are normalized by environmental and pharmacological interventions. Considering this global context, the work presented here needs to be understood as a little contribution to this general aim. Specifically, this work assessed multi-unit activity (MUA) levels in two brain regions crucial for cognition in a mouse model of DS. Hence, it aims to better understand cognitive normalization by pharmacological and environmental interventions.

1.1. *Cognitive function is impaired in Down syndrome*

Down syndrome (DS) is the most common genetic cause leading to intellectual disability. This disorder results from the triplication of genes on human chromosome 21 (Hsa21), affecting 10 of every 10,000 live births in European countries (1). Despite its onset occurs during prenatal development, the main manifestations may not become apparent until adulthood.

Some of these manifestations include different brain morphological alterations (Table 1). Moreover, during ageing most of DS individuals develop symptoms typical to Alzheimer's disease (2). At the cellular level, irregular laminar formation and delayed myelination of cortical fibres has been described. A decrease in neuronal density in different regions including the cochlear nuclei, cerebellum, HPC, basal forebrain, the granular layers of the neocortex and areas of the brainstem have also been reported (2).

Even though a variety of phenotypes have been described in DS, all of them showing different penetrance and severity, a significant level of cognitive disability is always common in all DS individuals (3). Actually, similar IQ range has been described among all these individuals, thus reflecting cognitive deficiency. Examples of this impairment are the disruption in keeping information online, performance of mental computations and storage for a future use on this information (2).

Memory is also impaired in DS individuals, showing poor performance at spatial-simultaneous memory tests. Memory deficits result also in cognitive impairment, including language and vocabulary development and problem-solving skills (2).
1.2. **Electrophysiology as a tool to understand cognitive function**

Brain electrophysiology characterizations are crucial for understanding the cellular mechanisms of learning and memory, especially in diseases underlying cognitive impairment such as DS. It is well-known that executive tasks with high demand require precise communication and synchronization of neural activity across structures. Two particular important ones are the prefrontal cortex (PFC) and hippocampus (HPC). Interestingly, a communication between these two structures has been reported during associative learning and memory acquisition (4–6).

This communication is supported by neural oscillations, described as electrical rhythmic changes generated by thousands of discharging neurons (7). These electric currents generate electric potentials that can be measured with respect to a reference. Differences of electric potentials between two locations can be monitored by simultaneous electrodes with a time resolution of milliseconds, constituting a useful tool to interpret neuronal communication. These differences give rise to an electric field, a vector whose amplitude is measured in Volts per distance. The oldest and widely used way to record this electric activity in the scalp is the electroencephalogram (EEG). When the electrodes are located intracerebrally it is called LFP (Local Field Potential). When compared to EEG, which samples large brain areas, LFPs allow the recording of deeper and more specific areas in the brain (8).

Up until now the golden standard to study information processing in the brain is to measure spiking activity with intracranial electrodes. This method, as compared to others, allows the best spatial and temporal resolution at the same time. The simplest way to understand neural spiking is to analyse the action potentials of a single cell by recording single unit activity (SUA). Even though this method can provide accurate information about a particular neuron, it doesn’t consider communication between neurons and the presence of brain networks. For this reason, measuring multiunit activity (MUA), which is defined as the aggregate activity of several neurons in the vicinity of the electrode, constitutes a better method to understand neural dynamics (9).

In terms of neural network activity, different studies have correlated neuronal oscillations to memory and learning. For example, it has been reported that neuronal oscillations in PFC, especially alpha (8-14 Hz) and beta (15-30 Hz) play an important role in associative learning and memory (10). However, it is very poorly understood how synchronized neural activity is disrupted in DS during cognition. It has recently been reported that DS individuals show a correlation between the severity of cognitive
impairment and global increase in inter-regional synchrony (11). However, brain activity impairment in DS needs better description.

1.3. **Neurophysiological alterations in Down syndrome mouse models**

In order to understand neurological alterations in DS, different mouse models had been studied (12). For an animal model of a human disorder to be considered valid, it has to satisfy different criteria including construct validity. It refers to the similarity between the aetiology of the human and the animal disorder. For example, in the case of DS, it would be the triplication of Hsa21 genes. The long arm of Hsa21 contains approximately 552 genes, in which 166 are orthologous to those localized in specific regions of three mouse chromosomes: Mmu16 (110 orthologous genes), Mmu17 (19 orthologous genes) and Mmu10 (37 orthologous genes). These homologies have been used to develop different mouse models. One example is Tc1 mouse, a mosaic mouse model as only contains a segregate copy of Hsa21 in half of its cells (12).

Despite the existence of several mouse models, this study focused on Ts65Dn, the one most commonly used and best characterized model of DS showing well-documented DS-like deficits. Ts65Dn is trisomic for more than half genes of Hsa21 homologs (12).

Considering cognitive impairment, Ts65Dn mice exhibit deficits in behavioural tasks such as those related to declarative memory (novel object recognition and spontaneous alteration tasks) and the proper encoding and recollection of spatial information (radial arm and Morris water mazes) (12).

At a morphological level, Ts65Dn mice show a reduction in the volume of the cerebellum and HPC, a reduction in the neuronal density of dentate gyrus as well as a decrease in the number of excitatory synapses and synaptic length in the temporal cortex (13–15).

Other morphological deficits described in this mouse model includes fewer excitatory neurons in the neocortex and decreased spine densities with larger spine volumes both in the neocortex and the HPC. A shift of inhibitory synaptic connections away from the dendritic shafts and onto the necks in the dentate granule cells of the HPC has also been described in Ts65Dn mice (16). This is consistent with a failure to inhibit activity which results in the observed hyperactivity in this mouse model (17). Hence, these alterations lead to deficits in learning and operant conditioning paradigms (18).
Finally, when referring to Hippocampal long-term potentiation (LTP), which has been described as the electrophysiological substrate of learning, Ts65Dn mice display reduced LTP in hippocampal CA1 and Dentate Gyrus regions (16). It has been suggested that this alteration is caused by an unbalanced excitatory and inhibitory neurotransmission (19).

This information suggests that assessing MUA in a murine model of DS as Ts65Dn may constitute a promising way to understand its cognitive impairments associated with memory and learning deficits.

1.4. Environmental enrichment normalizes cognitive deficiency in Down syndrome mouse models

Dendritic abnormalities result in alterations of several signalling pathways in Ts65Dn mice. Abnormal synaptic morphology and abnormal membrane properties have been described in this mouse model (20). These alterations of the number and shape of spines have been associated with synaptic plasticity and learning and can cause serious consequences on cognitive abilities.

One of the interventions that enhances dendritic growth is environmental enrichment (EE), which is associated with morphological, physiological and behavioural changes. It has been reported to increase cortical weight and thickness, the number of synapses per neuron, dendritic branching, number and length of spines, size and number of synaptic junctions and neurogenesis in specific areas of the brain (21). Moreover, these changes are associated with changes in the expression of genes involved in neuronal structure, synaptic signalling and plasticity after inducing enrichment (22).

It has been proven that euploid, non-trisomic mice exposed to an enriched environment showed increase in dendritic complexity and in the number of spines. However, this effect could not be demonstrated in Ts65Dn mice (23). Nevertheless, another study demonstrated that EE modulates spatial learning of control and trisomic Ts65Dn female mice. The effect, however, was not observed in males (21). The way EE could modulate brain neural activity in Ts65Dn mice has not yet been reported.

1.5. Chronic epigallocatechin-3-gallate rescues cognitive deficits in Ts65Dn mice

Another approach in the treatment of DS phenotypes consisted in directly targeting the Hsa21-orthologous candidate genes to normalize their expression levels, which are triplicated in DS. One of this target genes is the Dual-specificity tyrosine-(Y)-
phosphorylation regulated kinase 1A gene product (Dyrk1A). It has been proven to interact and phosphorylate a huge variety of neuronal molecular substrates (24). In one recent study, Dyrk1A expression levels were normalized in Ts65Dn mice with a short hairpin RNA showing attenuation of the synaptic plasticity defects (25).

Another strategy to normalize Dyrk1A expression has been a treatment with the flavonol epigallocatechin gallate (EGCG). It is the main active cathechin of green tea and has proved to have health benefits in the prevention of cardiovascular diseases, cancer chemoprevention, and to be a neuroprotector in neurodegenerative diseases. A recent study has reported EGCG to rescue cognitive deficits in both Ts65Dn and transgenic mice overexpressing Dyrk1A. This same work also undergone a pilot study with DS human individuals showing that EGCG have effects on memory recognition, working memory and quality of life. Phase I and II clinical trials of this study show that oral doses of EGCG were safe and well-tolerated (26). Hence, these results prove EGCG to be a safe and potential treatment to restore cognition in both DS mice and humans.

1.6. The double treatment of Environmental Enrichment and Epigallocatechin-3-gallate as a promising approach to cognitive normalization

A recent study performed in TgDyrk1A transgenic mice has found that EE also normalized Dyrk1A expression levels in the HPC, rescuing neurogenesis alterations (27). Moreover, a recent clinical trial report EGCG and cognitive training to improve visual recognition memory, inhibitory control and adaptative memory in DS individuals (28). These data suggest that the double treatment of EE and EGCG can be a potential approach to restore cognitive impairment in DS by normalizing Dyrk1A expression levels.

Altogether, electrophysiology assessment of MUA in Ts65Dn mice subjected to EE and the double treatment of EE and EGCG seems a novel and interesting approach to study cognitive normalization in DS.
2. **Hypothesis and objectives**

The main aim of this project is to assess how EE and EGCG normalize aberrant neural activity in a mouse model of DS. Considering that Ts65Dn has shown to be hyperactive (17), the first hypothesis of this work is that trisomic mice will present an exacerbation in MUA. Based on recent studies showing satisfactory results with EE in trisomic mice (21), another hypothesis of this project is that MUA will decrease in enriched Ts65Dn mice. Finally, other work has reported that EGCG is able to reduce cognitive deficits in trisomic mice (26). Hence, it is also hypothesised in this project that the double treatment of EE and EGCG will have an enhanced effect in restoring normal MUA activity in Ts65Dn mice when compared to their wild type littermates. Considering the above, the specific objectives of this work are:

1) To understand how EE and EGCG normalize cognitive impairment in a mouse model of Down syndrome, by:
   a. Carrying out an adequate background searching in the literature.
   b. Understand previous data acquisition consisting in mouse surgery for electrode implantation and neural activity recordings.

2) To prepare previous acquired data to assess MUA:
   a. Pre-process raw data from electrophysiological recordings in quiet states to extract spiking activity using Python scripts.
   b. Filter the data with an adequate threshold.
   c. Sort spike waveforms of MUA recordings using a spike sorting software.

3) To interpret the data and conduct statistical analyses:
   a. Analyse the filtered data in an efficient way using Python programming language to obtain the firing rate of every recording.
   b. Study the histology in order to report the position of the electrodes and identify possible outliers.
   c. Report the differences between firing rates in the different groups and carry out a deep statistical analysis.
3. Methodology:

In order to accomplish every one of the mentioned objectives the subsequent methodology was followed.

3.1. To understand how EE and EGCG normalize cognitive impairment in a mouse model of Down syndrome

The first step of this project consisted in an exhaustive analysis of the recent literature in order to acquire a proper background. Hence, free access databases as PubMed (https://www.ncbi.nlm.nih.gov/pubmed/) were used.

Once this background was obtained, an understanding of the data acquisition procedure was needed. Due to ethic and protocoly reasons, data could not be directly obtained by the student in this project as it involved animal manipulation. However, data acquisition was observed (both surgical procedures and actual electrophysiological recordings) to a better understanding and knowledge of the whole project.

Ts65Dn mice and their respective wild type littermates were maintained in a 12 hour light-dark cycle (8:00 to 20:00) and controlled environmental conditions of humidity (60%) and temperature (22 +/- 1 ºC) with free access to food and water. After weaning (21 days of age) all the animals were separated by sex and randomly classified in three groups: control, EE and EE + EGCG. Controls animals were reared in conventional cages (20 x 12 x 12 cm height, Plexiglas cage) in groups of 2-3 animals. Alternatively, EE animals were moved at week 3 to spacious Plexigas cages (55 x 80 x 50 cm height) with toys, small houses, tunnels and platforms being changed every 3 days to enhance novelty. Moreover, to stimulate social interactions, 4-5 mice were housed in each cage. In half of the EE cages, mice were administered EGCG in the drinking water (2-3 mg per days).

Surgery of the mice was needed for implantation of intracranial electrodes. Mice were anaesthetized with a mixture of ketamine/xylazine (75:1, 0.10 ml/10 g i.p.) and this state was maintained with isoflurane. Stereotaxic apparatus was used to perform craniotomies above the medial PFC (mPFC) and the HPC. In detail, implantation of tungsten electrodes was undergone in the following coordinates mPFC (AP: 2.0 mm; ML: ± 0.4 mm; DV: -1.5 mm from bregma) and the HPC (AP: -2.5 mm; ML: -2.0 mm; DV: -2.5 mm from bregma). References were set in the lateral ventricle for the HPC.
and corpus callosum for the PFC. Common reference was set either in the corpus callosum or lateral ventricle (Figure 1). They were fixed using micro-screws and dental cement was used to finally assemble the implant. Electrophysiological recordings allowed to properly identify the areas with both rich and noise-free neural activity. A ground was fixed in the cerebellum. An adaptor emerging from the implant facilitated wires connection and recordings. All animals underwent 1-week recovery period after surgery receiving analgesia (buprenorphine) and anti-inflammatory (meloxicam) treatment.

The next step consisted of performing electrophysiological recordings. Neural activity was simultaneously recorded using a multi-channel Open Ephys system (http://www.open-ephys.org/) via tungsten stereotrodes both from the PFC and the HPC during all these different conditions, including quiet wake without movement, active exploration in an open field, Novel Object Recognition test (NOR) and spatial memory task in an operant chamber among others. The signal was recorded at a sampling rate of 30,000 Hz. As the electrode implant was chronic, large amount of data was recorded during 5 months of each mouse life. It is important to mention that only quiet wake recordings of 5 min time duration were analysed in this work due to time constrains.

Mice were classified in six groups depending on three different treatments (control, Environmental Enrichment (EE) and double treated with Environmental Enrichment and Epigallocatequin-3-gallate (EE + EGCG)) and on genotype (Wild Type (WT) and Ts65Dn (TS) mice). The total amount of animals in each group was: WT Control (n=6), WT EE (n=6), WT EE+EGCG (n=8), TS Control (n=6), TS EE (n=6), TS EE+EGCG (n=8). Due to lack of data availability only 22 animals were analysed in this work consisting in 3 animals of every group except for the EE TS group with 6 animals (Table 2).

3.2. To prepare previous acquired data to asses MUA

Continuous neural activity signals previously acquired in the recordings were pre-processed using Python scripts to assess MUA. Raw data was stored in the HDF5 format (Kwik kwd output format from Open Ephys). This format allows to store big amount of data in just one file. An existing Python script called “Puig_lab_analysis.py” was used in order to extract only the channels containing data and store the information into an uncompressed MATLAB format (Mat). Moreover, this script
substituted the bit scale into Volts using a factor conversion to correct previous amplification of the data. Extracted data was then converted to a Nex file compatible with Plexon Offline Sorter Software, the programme used in further steps for spike sorting. An existing Python script called “Extracted_to_nex.py” was used. This script associated each file with the appropriate adaptor allowing a correct channel map of signals, and referenced it to the electrodes showing the best signal-to-noise ratio. References without spiking activity located either in corpus callosum or the third ventricle were used to remove artefacts generated by the own movement of the animal. Moreover, a band-pass filter between 450 and 6000 Hz was applied. This is because spiking activity is found in this frequency range. Therefore, pre-processed data was obtained in a Nex format file (Figure 2).

Spike sorting was performed using the commercial software Plexon Offline Sorter (Plexon, USA). Before sorting, it was needed to apply a voltage threshold to optimize the ratio of false positive (noise or artefact events that cross the threshold) to false negatives (missed spikes). This threshold was set at 3 standard deviations (3σ) from the background noise mean (Figure 3).

Finally, waveforms differing from the physiological ones were invalidated to sort the correct spikes only (Figure 3). Therefore, manually inspecting the waveform shapes was performed in a narrow time window (0.5-1s). Principal Component Analysis (PCA) is used by Plexon Offline Sorter to classify all the waveforms in clusters based on different features of the waveforms (for example, maximum trough and peak, average amplitude, etc.). Hence, this tool was used in this step to manually invalidate the waveforms whose position was far from the cluster of neurons and likely reflected noise or artefacts during the recordings. So, manual clustering was used to assign waveforms to each unit. In some cases, more than one cluster was obtained indicating the presence of different units. The result of spike sorting was saved in a plex file with only containing the sorted spikes (Figure 3). It is important to mention that spike sorting was blind to avoid bias. Therefore, it was conducted without knowing the treatment or either the genotype of the animal recording.

3.3. To interpret the data and conduct statistical analyses

The filtered data was stored in a plex file as previously mentioned. However, processing of this data was needed to analyse MUA. For this reason, a Python script was created named “plx_mua_analysis” (Figure S1). The first step consisted in creating
a dictionary using *Pandas DataFrame* Python library ([https://pandas.pydata.org/](https://pandas.pydata.org/)). This tool allowed to classify every file in different keys within a big dictionary (animal name, channel, condition, treatment, firing rate, total number of spikes and total recorded seconds) (Figure 4).

Animal name indicates the code assigned to each mouse. Channel referred to the corresponding electrode either in PFC and the HPC (1,2 corresponding both to the 2W electrode and 3 to the 1W electrode. See Figure 1). Condition refers to the animal’s state. In this case only quiet state was considered as already mentioned. Total amount of samples referred to the total number of possible spikes identified by Plexon Offline Sorter. The value was extracted using the following Python code:

```python
total_samples = seg.annotations['LastTimestamp']
```

Total recorded seconds were calculated dividing the total amount of samples (samples/s) between the sampling rate (30K samples/s per second):

```python
seconds = seg.annotations['LastTimestamp'] / 30000
```

Finally, the firing rate referring to the action potential discharge rate of the MUA was calculated. In this way, a time window of the last 3 minutes of the recording (that originally lasted 5 min) was used. This was done to discard possible biological noises in the beginning of the recording. This is explained due to mice presenting anxiety in the first minutes of the recording and thus affecting the results. To obtain the firing rates, the samples considered in this time window were divided by the time (3 min or 180 seconds in this case) obtaining a final value in spikes/seconds. The following Python code was used:

```python
mysamples = (total_samples - 5400000) / 30000
myspikes = np.array(seg.spiketrains[element])
spikes_total = len(np.array(seg.spiketrains[element]))
firing_rate = sum(myspikes > mysamples) / 180
```

The last step of the script consisted in obtaining tools to interpret and analyse the data. In this way, Boxplots and Bar charts were obtained using Matplotlib Python library ([https://matplotlib.org/](https://matplotlib.org/)). An Excel file was also created to store and organize all the
information using Xlsxwriter Python library (https://xlsxwriter.readthedocs.io) (Figure 4).

Once the data was analysed the next logical step was to identify possible outliers. In this case, three criteria were considered. First, the mean +/- two times the standard deviation between the different groups was calculated (Table S1). Following this criteria only one sample was considered as an outlier: LJ_21 HPC3.

A second criteria consisted in using the boxplots previously generated by Matplotlib (Figure S2). In these boxplots the outliers were considered as the third Q3 + whis*IQR or Q1 – whis*IQR, where Q3 refers to the third quartile, Q1 refers to the first quartile, IQR refers to the interquartile range (Q3-Q1) and this is a factor determined as 1,5. The following outliers were obtained: LJ_21 HPC2, LJ_8 PFC3, LJ_8 PFC2, LJ_5 PFC3, LJ_7 PFC3.

In addition, histology pictures consisting in mouse brain slices were checked out. Each mice electrode had been previously electrically stimulated at 100 Hz, 2 and 5 mA and between 2 and 10 seconds. The location of the electrodes was assessed to identify possible wrong insertions. In this case, any outlier was identified as all the electrodes were located in the proper position. However, it needs to be mentioned that the number of animals studied in this work was not large enough to definitively exclude any potential outlier.

Finally, statistical analysis was performed. A two-way analysis of variance (ANOVA) was used to determine significant statistical differences in the average firing rates either in HPC and PFC among the six groups (divided in the fixed factors of genotype and treatment). IBM SPSS (https://www.ibm.com/analytics/es/es/technology/spss/) was used and its script was changed to properly adequate the analysis (Figure S3).

4. Results:

In order to assess neural activity, MUA recordings were used to calculate the mean firing rate in spikes per second (spikes/s) for each group (considering treatment and genotype). PFC and HPC firing rate means were separately analysed.
Environmental Enrichment and Epigallocatequin-3-gallate combined significantly normalize aberrant neural activity in PFC of Ts65Dn mice

Two-way ANOVA showed statistical significant differences between the firing rate in PFC when considering the interaction between genotype (WT or TS) and Treatment (control, EE or EE+EGCG) when taken as potential factors (p=0.037) (Figure S3). Subsequently, interactions between groups were analysed.

We first compared the neural activity between WT and DS genotypes in PFC (Figure 5). Ts65Dn (TS) mice presented an aberrant increased firing rate in PFC when compared to WT (TS 60.48 spikes/s ± 3.88 SEM against WT 46.89 spikes/s ± 4.18 SEM) (p=0.162). This aberrant activity was reduced to wild-type like values in TS enriched mice (TS 53.36 spikes/s ± 4.29 SEM compared to WT 51.37 ± 4.84 SEM) (p=0.893). When EGCG treatment was applied in enriched mice the situation was reversed as TS mice showed a much lower firing rate than WT mice (TS 45.4 spikes/s ± 3.65 SEM compared to WT 68.37 spikes/s ± 6.34 SEM). In the latter case differences were statistical meaningful (p=0.024).

Considering differences of genotype, WT mice showed an increased average firing rate both after EE (8.73% ± 0.65 SEM) or double treatment, EE+EGCG (31.43% ± 0.21 SEM) (Figure 6a). However, only in the double treatment the differences were statistically significant (p=0.033).

Referring to TS mice, the ones that were enriched showed a decrease in the firing rate mean when compared to control mice (-13.35% ± 1.72 SEM) (p=0.423). Double treated mice showed also a stronger reduction of the average firing rate (-33.23% ± 2.01 SEM) (p=0.124) compared to control mice. Hence, the results showed that when EGCG is administered to enriched mice, the firing rate decrease is enhanced compared to enrichment therapy alone. Statistical significance was absent likely because of the small number of samples (Figure 6b).

Environmental enrichment but not Epigallocatequin-3-gallate significantly normalize aberrant neural activity in HPC of TS mice

Two-way ANOVA did not report statistical significant differences between the firing rate mean in HPC considering the two fixed factors genotype and treatment. However, treatment factor appears to be significant independently of the genotype (p=0.022) (Figure S3). Interactions between groups were also analysed in this case.
We found (Figure 7) that TS mice also showed an excessive firing rate in HPC when compared to WT mice (107.64 spikes/s ± 5.18 SEM compared to 78.44 spikes/s ± 7.36 SEM). These differences were almost statistically significant (p=0.05). This aberrant activity was normalized in enriched TS mice with respect to WT, both showing similar firing rate means (TS 65.88 spikes/s ± 6.16 SEM compared to WT 62.94 spikes/s ± 10.36 SEM) (p=0.806). However, when EGCG was applied to enriched mice, the double treatment did not result in neural activity normalization. TS mice showed elevated firing rate mean instead, when compared to WT mice (TS 82.57 spikes/s ± 9.68 SEM and WT 68.2 spikes/s ± 9.68 SEM) (p=0.332).

Referring the activity recorded in WT after the different treatments (Figure 8a), enriched WT mice showed a decrease in firing rate when compared to control mice (-24.62% ± 7.57 SEM) (p=0.267). Moreover, when EGCG treatment was administered to WT mice, it reduced the average firing rate as compared to control mice (-15.01% ± 4.84 SEM) (p=0.490).

Considering TS mice (Figure 8b), enriched animals presented a decrease in the mean firing rate as compared to control mice (-63.4% ± 6.78 SEM). This difference was statistically significant (p=0.003). Administration of EGCG to TS enriched mice resulted in a reduction in the mean firing rate (-30.36% ± 9.87 SEM with respect to control mice) (p=0.093). Thus, both treatments resulted in a decrease in firing rate mean in HPC of TS mice with respect to control mice.

5. Discussion
The results obtained in this work show that EE and EGCG, when administered in combination, significantly normalise aberrant neural activity in the PFC of Ts65Dn mice. Even though this effect is not statistically significant in the HPC, trisomic mice show reduced exacerbation of the firing rate in this area after both interventions.

Aberrant neural activity in Ts65Dn mice described in this work is consistent with other findings and provide reliable electrophysiological data to understand neural abnormalities in DS. Several studies have reported genetical, morphological and molecular alterations in Ts65Dn mice (12-18). However, electrophysiological abnormalities in this mouse model are poorly understood. In this work, we describe an exacerbated increase in MUA activity in Ts65Dn mice when compared to controls. This is consistent with previous work which found an increase of power at low EEG frequencies in DS human individuals (29) and with similar results found in mouse
models (30). A reported excitatory/inhibitory imbalance in adult Ts65Dn mice (19), together with other alterations such as hyperactivity (17), may be a possible explanation for these neural abnormalities.

This work provides novel electrophysiological data suggesting EE as a suitable therapy to rescue DS neurophysiological aberrations. Here we have shown that EE reduced exacerbated MUA activity in trisomic mice as compared to their WT littermates, both in the HPC and the PFC. This is consistent with other work reporting EE to rescue dendritic abnormalities in Ts65Dn mice (21). Other studies have demonstrated beneficial effects of enrichment, such as preventing cognitive impairment in epileptic rats (30) or mitigating cognitive deficits in a mouse model of Alzheimer’s disease (31). This suggest that brain activity may also be assessed in other pathologies to better understand cognitive normalization.

Combined therapy of EE and EGCG may constitute a promising approach in rescuing cognitive alterations in DS. Pharmacological interventions using the green tea extract EGCG have shown to restore Dyrk1A overexpression in Ts65Dn mice and, therefore, rescued cognitive deficits (26). Moreover, EE has also proved to normalize Dyrk1A expression levels in TgDyrk1A transgenic mice (27). This is consistent with our results as we show that the double treatment of EE and EGCG significantly normalize aberrant neural activity in the PFC and has an enhanced effect when compared to enrichment intervention alone. Actually, previous studies have proved this double treatment to restore learning deficits in Ts65Dn mice (32), and a recent clinical trial showed positive results in humans when combining cognitive training and EGCG (28). Therefore, this work provides valuable electrophysiological data reinforcing the idea of this combined therapy as a promising treatment to normalize cognitive alterations in DS both in humans and mice.

Data analysed in this work was not enough to ensure statistical robustness due to time constrains. Statistically significant difference in the HPC firing rate mean between double treated mice and their WT littermates was not found. This may be caused by an unequal distribution of the number of mice in every group as well as little number of animals being studied due to lack of available data and time limitations. Considering that, more electrophysiological data needs to be done and carefully analysed to demonstrate further this effect also in the HPC.
Interestingly, when the double treatment of EE and EGCG was applied to WT mice, their MUA activity in the PFC increased, thus showing aberrant neuronal activity that is typical of TS mice. Hence, potential adverse effects of EGCG need to be considered (33).

In conclusion, this study has been successful in identifying aberrant MUA in a trisomic mouse model during quiet wakefulness and assessing its normalization by two different interventions, one environmental and one pharmacological. Even though MUA has been described as a reliable and robust method to study neural activity, little is known of how MUA is disrupted in DS. Accordingly, this study provides novel and valuable data to unravel the mechanisms underlying cognitive impairment and its normalization in DS.

However, some limitations need to be solved for further work. First, data include small groups of mice and unequal distribution among the groups, thus resulting in statistical bias. Moreover, further experiments with a group treated only with EGCG need to be performed to increase statistical validity and robustness of this project. Finally, a thorough analysis of neural oscillations and MUA during the performance of the novel object recognition task are underway by a graduate student in the Puig laboratory and are out of the scope of this project.

6. Conclusions
   1) Excessive MUA is present in the PFC and HPC of TS mice compared to their WT littermates.
   2) EE probes to be a suitable therapy to rescue neural activity abnormalities in TS mice both in the PFC and the HPC.
   3) Combining enrichment with a green tea extract (EGCG) potentiates the effect mediated by enrichment therapy alone, constituting a promising therapeutic approach in normalizing DS neurophysiological abnormalities.

7. Acknowledgements
To the support and guidance of Dra Maria Victoria Puig, the helpful advices and comments of Dr Thomas Gener as well as Maria Alemany, who provided all the data, and Cristina Delgado and specially to the patience of Pau Nebot in resolving all my doubts. Finally, to Dr Fernando Giraldez who accepted to be my DCEXS tutor. This work is supported by Fondation Jérome Lejeune.
8. Tables and Figures

**Table 1**

<table>
<thead>
<tr>
<th>Brain region</th>
<th>Newborns</th>
<th>Adults (20–50 years of age)*</th>
<th>Elderly individuals (&gt;50 years of age)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole brain</td>
<td>Almost normal weight</td>
<td>Reduction in weight, brachycephalic</td>
<td>Smaller overall cerebral volumes</td>
</tr>
<tr>
<td>Prefrontal cortex</td>
<td>Reduction in volume</td>
<td>Reduction in volume</td>
<td>Reduction in volume</td>
</tr>
<tr>
<td>Parietal cortex</td>
<td>Normal or reduction in volume</td>
<td>Reduction in volume</td>
<td>Unknown</td>
</tr>
<tr>
<td>Temporal cortex</td>
<td>Narrow superior temporal gyrus</td>
<td>Reduction in volume of right middle or superior temporal gyrus</td>
<td>Decreased grey matter volume in posterior cingulate and entorhinal cortex</td>
</tr>
<tr>
<td>Hippocampus</td>
<td>Unknown</td>
<td>Reduction in volume</td>
<td>Unknown</td>
</tr>
<tr>
<td>Parahippocampal region</td>
<td>Unknown</td>
<td>Increase in size of the parahippocampal gyrus</td>
<td>Reduction in volume</td>
</tr>
<tr>
<td>Amygdala</td>
<td>Reduction in volume</td>
<td>Reduction in volume</td>
<td>Reduction in volume</td>
</tr>
<tr>
<td>Cerebellum</td>
<td>Reduction in volume</td>
<td>Reduction in volume</td>
<td>Reduction in volume</td>
</tr>
<tr>
<td>Brain stem</td>
<td>Reduction in volume</td>
<td>Increase in grey matter volume</td>
<td>Degeneration of locus coeruleus</td>
</tr>
<tr>
<td>Basal prosencephalon</td>
<td>Almost normal size</td>
<td>Normal</td>
<td>Degeneration of basal prosencephalon cholinergic nuclei (nucleus of Meynert)</td>
</tr>
</tbody>
</table>

*Adulthood begins later in people with trisomy 21 than in individuals without intellectual disability, although people with Down syndrome are considered to be elderly once they have reached 50 years of age.

**Table 2**

<table>
<thead>
<tr>
<th>Treatment/ genotype</th>
<th>Control</th>
<th>EE</th>
<th>EE + EGCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>WT</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>TS</td>
<td>3</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

**Figure 1**

Electrodes stereotaxic coordinates inserted in mice brain during surgery. Numbers indicate the following positions: 1) PFC 1W (AP 0.21, ML 0.02, DV 1.7) 2) PFC 2W (AP 0.154, ML 0.06, DV 1.7) 3) PFC reference, corpus callosum (AP 0.1, ML 0.1, DV 1.25) 4) Common reference 1W, corpus callosum or lateral ventricle (AP 0.02, ML 0.08, DV 1.4) 5) Hippocampus reference, lateral ventricle (AP -0.1, ML 0.175, DV 1.5) 6) Hippocampus 2W, (AP -0.18, ML 0.13, DV 1.15) and 7) Hippocampus 1W (AP -0.25, ML 0.23, DV 1.25). All coordinates measured in mm and indicating its position from Bregma.
Work-flow showing raw-data pre-processing procedure. Continuous LFP recordings were stored in the HDF5 format. After channel extraction and substitution of the bit scale to Volts data was stored in a Mat format. Finally, referencing and band-pass filter of the data between 450 and 6000 Hz resulted in pre-processed data stored in Nex file format.

Scheme representing spike sorting procedure. Plexon Offline Sorter was used to define a threshold of 3 times the standard deviation. Then, manually invalidating of the waveforms was undergone deleting non-physiological ones. PCA was used to manually invalidate waveforms away from the cluster unit. Finally, filtered data was stored as a Plex file. PCA refers to Principal Component Analysis.
Figure 4

Workflow representing the steps followed to analyse the filtered data. Python language with Pandas DataFrame Python library was used to create a Dictionary sorting every file in 7 different keys. Firing rate was calculated only considering the last three minutes of the recording. Data Analyses consisted in creating boxplots and bar charts using Matplotlib Python library and an excel file with all the data being stored using Xlsxwriter Python library.

![Workflow diagram](image)

Figure 5

Environmental enrichment and epillogatechin-3-gallate normalize aberrant PFC neural activity in TS mice compared to WT mice. Bar chart representing firing rate mean differences in PFC between control TS (N=3) and WT (N=3) mice, environmental enriched (EE) TS (N=6) and WT (N=3) mice and double treated (EE+EGCG) WT (N=3) and TS (N=3) mice. Error bars represent the Standard Deviation of the Mean (SEM). Firing rate is calculated in spikes/second with a time window of the last 3 min of the recording. Statistical significance across genotype and treatment groups was determined using a two-way Analysis of Variance (ANOVA). P value <0.05, which indicates a statistically significant difference among relevant groups, is designated with an asterisk.

![Bar chart](image)
Environmental enrichment and epillogatechin-3-gallate reduce PFC aberrant neural activity in TS mice while having a reverted effect in WT mice. Figure A represents a bar chart showing firing rate mean differences in PFC between enriched (EE) WT mice (N=3) and double treated with EE and EGCG (EE+EGCG) WT mice (N=3) compared to control (N=3). Figure B represents a bar chart showing firing rate mean differences in PFC between enriched (EE) TS mice (N=6) and double treated with EE and EGCG (EE+EGCG) TS mice (N=3) compared to control (N=3). Error bars represent the Standard Deviation of the Mean (SEM). Firing rate is calculated in spikes/second with a time window of the last 3 min of the recording. Statistical significance across genotype and treatment groups was determined using a two-way Analysis of Variance (ANOVA). P value <0.05, which indicates a statistically significant difference among relevant groups, is designated with an asterisk.

Environmental enrichment but not epillogatechin-3-gallate normalize HPC aberrant neural activity in TS mice compared to WT mice. Bar chart representing firing rate mean differences in HPC between control TS (N=3) and WT (N=3) mice, environmental enriched (EE) TS (N=6) and WT (N=3) mice and double treated (EE+EGCG) WT (N=3) and TS (N=3) mice. Error bars represent the Standard Deviation of the Mean (SEM). Firing rate is calculated in spikes/second with a time window of the last 3 min of the recording. Statistical significance across genotype and treatment groups was determined using a two-way Analysis of Variance (ANOVA). P value <0.05, which indicates a statistically significant difference among relevant groups, is designated with an asterisk.
Environmental enrichment and epilogatechin-3-gallate decrease HPC neural activity both in WT and TS mice compared to control. Figure A represents a bar chart showing firing rate mean differences in HPC between enriched (EE) WT mice (N=3) and double treated with EE and EGCG (EE+EGCG) WT mice (N=3) compared to control (N=3). Figure B represents a bar chart showing firing rate mean differences in PFC between enriched (EE) TS mice (N=6) and double treated with EE and EGCG (EE+EGCG) TS mice (N=3) compared to control (N=3). Error bars represent the Standard Deviation of the Mean (SEM). Firing rate is calculated in spikes/second with a time window of the last 3 min of the recording. Statistical significance across genotype and treatment groups was determined using a two-way Analysis of Variance (ANOVA). P value <0.05, which indicates a statistically significant difference among relevant groups, is designated with an asterisk.

9. Bibliography:


9. Supèr H, Roelfsema PR. Chronic multiunit recordings in behaving animals: Advantages and


10. Appendices: Supplementary data

Supplementary Table 1.
Table showing firing rate mean ± two times standard deviation (spikes/second) between the different mouse groups considering treatment and genotype. PFC and HPC were separately analysed. PFC refers to Prefrontal Cortex, HPC refers to Hippocampus, EE refers to Environmental Enrichment, EGCG refers to Epigallocatequin-3-gallate. WT refers to Wild Type mice and TS to Ts65Dn mice.

### Supplementary Table 1.
Table showing firing rate mean ± two times standard deviation (spikes/second) between the different mouse groups considering treatment and genotype. PFC and HPC were separately analysed. PFC refers to Prefrontal Cortex, HPC refers to Hippocampus, EE refers to Environmental Enrichment, EGCG refers to Epigallocatequin-3-gallate. WT refers to Wild Type mice and TS to Ts65Dn mice.

#### PFC

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>EE</th>
<th>EE+EGCG</th>
<th>Genotype</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>min</strong></td>
<td>21,7538</td>
<td>23,98891</td>
<td>32,50511</td>
<td>WT</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>72,02028</td>
<td>78,7597</td>
<td>104,2449</td>
<td>WT</td>
</tr>
<tr>
<td><strong>min</strong></td>
<td>38,53552</td>
<td>17,92098</td>
<td>23,49292</td>
<td>TS</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>82,4367</td>
<td>88,79863</td>
<td>67,30708</td>
<td>TS</td>
</tr>
</tbody>
</table>

#### HPC

<table>
<thead>
<tr>
<th></th>
<th>Control</th>
<th>EE</th>
<th>EE+EGCG</th>
<th>Genotype</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>min</strong></td>
<td>36,78514</td>
<td>8,101547</td>
<td>10,11895</td>
<td>WT</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>120,0871</td>
<td>117,7746</td>
<td>126,2736</td>
<td>WT</td>
</tr>
<tr>
<td><strong>min</strong></td>
<td>78,32769</td>
<td>16,55679</td>
<td>51,9313</td>
<td>TS</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>136,9626</td>
<td>115,1967</td>
<td>113,2118</td>
<td>TS</td>
</tr>
</tbody>
</table>

**Supplementary Figure 1.**
Script using Python programming language being used to analyse the filtered data. Pandas Dataframe, Matplotlib and Xlsxwriter Python libraries were used.

```python
# -*- coding: utf-8 -*-

""
Created on Mon Apr 27 2018

@Author: pnebot and mvilademunt
""
import neo
import numpy as np
import tkinter
from tkinter import filedialog
import datetime
import xlsxwriter
from scipy.io import savemat
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
from statsmodels.graphics.factorplots import interaction_plot
import matplotlib.pyplot as plt
from scipy import stats

def file_selector_gui(title='Choose files', initialdir='', filetype=()):
    """
    Tkinter GUI to select one or more files.
    """
    Parameters
    --------
    title : str, optional
        A string that will appear as the title of the window. Default is: 'Choose files'.
    initialdir : str, optional
        The first path that will be explored by the GUI. Default is: '', the directory where the script is located or the one where the last file was selected with this function.
    filetypes : array_like, optional
        A vector containing description-extension pairs. Leaving this parameter
```
This function is in order to save matlab files.

```python
def plexon_spiketrain_extractor_channels():
    ""
    This function returns you a dictionary named data_dict, in which the input
data (obtained using file_selector_gui function) is being organized
following these conditions:
    - the NAME of the file is the first object in the filename.
    - A dictionary is going to be created for every NAME in which we will
        find the following keys:
        - ANIMAL: animal name
        - CONDITION: state condition extracted from the filename
        - GENOTYPE: animal genotype extracted from a previous
            stored list (list_TS or list_WT).
        - TREATMENT: animal treatment extracted also from a previous list.
        - SPIKES TOTAL: number of the total amount of spikes obtained with
            the seg.spikestrains function.
        - FIRING_RATE: spiking frequency obtained by considering only the
            spikes in the last 3 min and dividing by the time.
        - CHANNEL: channel being considered from the filename
        - SECONDS: number of seconds the register lasts obtained from
            dividing the total samples between the sampling rate (30000 hz).
    ""
    data_dict = {} #we create an empty dictionary to store all the data
    file_list = file_selector_gui() #here we use the function already defined
```

#This function is in order to save matlab files.

```python
def plexon_spiketrain_extractor_channels():
    ""
    This function returns you a dictionary named data_dict, in which the input
data (obtained using file_selector_gui function) is being organized
following these conditions:
    - the NAME of the file is the first object in the filename.
    - A dictionary is going to be created for every NAME in which we will
        find the following keys:
        - ANIMAL: animal name
        - CONDITION: state condition extracted from the filename
        - GENOTYPE: animal genotype extracted from a previous
            stored list (list_TS or list_WT).
        - TREATMENT: animal treatment extracted also from a previous list.
        - SPIKES TOTAL: number of the total amount of spikes obtained with
            the seg.spikestrains function.
        - FIRING_RATE: spiking frequency obtained by considering only the
            spikes in the last 3 min and dividing by the time.
        - CHANNEL: channel being considered from the filename
        - SECONDS: number of seconds the register lasts obtained from
            dividing the total samples between the sampling rate (30000 hz).
    ""
    data_dict = {} #we create an empty dictionary to store all the data
    file_list = file_selector_gui() #here we use the function already defined
```

for filename in file_list:
    # we open a for loop for every file
    name = filename.split('/'-1).split('.')[0]
    # from the name we already stored we extract the name of our file
    data_dict[name] = {}
    # we open a dictionary inside every file with this name
    reader = neo.io.PlexonIO(filename=filename)
    # we open the file
    seg = reader.read_segment(lazy=False, cascade=True)
    # we extract the name of the channel, animal and condition from the filename
    channel = filename.split('/'-1).split('.')[-1].split('_')[0].split('_')[-1]
    animal = '_'.join((filename.split('/'-1).split('_')[0:2]))
    condition = filename.split('/'-1).split('.')[-1].split('_')[2]
    # we now fill the dictionary with this data
    data_dict[name]['animal'] = animal
    data_dict[name]['condition'] = condition
    number_channels = range(len(seg.spiketrains))

    if animal in list_TS:
        data_dict[name]['genotype'] = 'TS'
    else:
        data_dict[name]['genotype'] = 'WT'

    if animal in list_EE:
        data_dict[name]['treatment'] = 'EE'
    elif animal in list_EE_EGCG:
        data_dict[name]['treatment'] = 'EE+EGCG'
    else:
        data_dict[name]['treatment'] = 'control'

    for element in number_channels:  # we analyse every channel
        unit_name = seg.spiketrains[element].annotations['unit_name']
        if channel == unit_name:  # we select only the channel of our file
            seconds = seg.annotations['LastTimestamp']
            total_samples = seg.annotations['LastTimestamp']
            # this is the total amount of samples
            mysamples = (total_samples - 5400000) / 30000
            # we measure the first sample in our time window (3 min)
            myspikes = np.array(seg.spiketrains[element])
            spikes_total = len(np.array(seg.spiketrains[element]))
            # total amount of spikes
            firing_rate = sum(myspikes > mysamples) / 180
            # we extract the spikes in the last 3 min and we divide for the time
            # we fill the dictionary with all the data
            data_dict[name]['spikes_total'] = spikes_total
            data_dict[name]['firing_rate'] = firing_rate
            data_dict[name]['channel'] = channel

            if animal in list_EE:
                data_dict[name]['treatment'] = 'EE'
            elif animal in list_EE_EGCG:
                data_dict[name]['treatment'] = 'EE+EGCG'
            else:
                data_dict[name]['treatment'] = 'control'
data_dict[name]["seconds"] = seconds

return(data_dict)

def boxplots():
    
    # This function is used to create boxplots for every condition we considered:
    # - TS and WT HPC
    # - TS and WT PFC
    # - Different treatments in PFC TS and WT
    # - Different treatments in HPC TS and WT
    
    # Here we will create boxplots with matplotlib to identify outliers:
    mydata = [[key,
                data_dict[key]["animal"],
                data_dict[key]["channel"],
                data_dict[key]["condition"],
                data_dict[key]["genotype"],
                data_dict[key]["spikes_total"],
                data_dict[key]["firing_rate"],
                data_dict[key]["seconds"],
                data_dict[key]["treatment"]] for key in data_dict.keys()]
    mydata_clean = pd.DataFrame(mydata)[[1,2,3,4,5,6,7,8]]
    mydata_clean.columns = ["animal", "channel", "condition", "genotype", "spikes_total", "firing_rate", "seconds", "treatment"]

    mydata_HPC = mydata_clean[mydata_clean["channel"].isin(["HPC1", "HPC2", "HPC3")]) # we select only the HPC

    mydata_PFC = mydata_clean[mydata_clean["channel"].isin(["PFC1", "PFC2", "PFC3")]) # we select only the PFC

    TS_HPC = mydata_HPC[mydata_HPC["treatment"] == "TS"]
    WT_HPC = mydata_HPC[mydata_HPC["treatment"] == "WT"]

    TS_PFC = mydata_PFC[mydata_PFC["treatment"] == "TS"]
    WT_PFC = mydata_PFC[mydata_PFC["treatment"] == "WT"]

    #TS HPC boxplots:
    TS_HPC_con = TS_HPC[TS_HPC["treatment"] == "control"]["firing_rate"]
    TS_HPC_EE = TS_HPC[TS_HPC["treatment"] == "EE"]["firing_rate"]
    TS_HPC_EE_EGCG = TS_HPC[TS_HPC["treatment"] == "EE+EGCG"]["firing_rate"]

    BP_TS_HPC = [TS_HPC_con, TS_HPC_EE, TS_HPC_EE_EGCG]
    labels = ["Control", "EE", "EE+EGCG"]

    fig, ax = plt.subplots()
    bp_TS_HPC = ax.boxplot(BP_TS_HPC, labels = labels)
    outliers_TS_HPC = bp_TS_HPC["fliers"][0].get_ydata()

    #WT HPC boxplots:
    WT_HPC_con = WT_HPC[WT_HPC["treatment"] == "control"]["firing_rate"]
    WT_HPC_EE = WT_HPC[WT_HPC["treatment"] == "EE"]["firing_rate"]
    WT_HPC_EE_EGCG = WT_HPC[WT_HPC["treatment"] == "EE+EGCG"]["firing_rate"]

    BP_WT_HPC = [WT_HPC_con, WT_HPC_EE, WT_HPC_EE_EGCG]
    labels = ["Control", "EE", "EE+EGCG"]

    fig, ax = plt.subplots()
    bp_WT_HPC = ax.boxplot(BP_WT_HPC, labels = labels)
    outliers_WT_HPC = bp_WT_HPC["fliers"][0].get_ydata()

    #TS PFC boxplots:
    TS_PFC_con = TS_PFC[TS_PFC["treatment"] == "control"]["firing_rate"]
    TS_PFC_EE = TS_PFC[TS_PFC["treatment"] == "EE"]["firing_rate"]
    TS_PFC_EE_EGCG = TS_PFC[TS_PFC["treatment"] == "EE+EGCG"]["firing_rate"]
BP_TS_PFC = [TS_PFC_con, TS_PFC_EE, TS_PFC_EE_EGCG]
labels = ['Control', 'EE', 'EE+EGCG']

fig, ax = plt.subplots()
bp_TS_PFC = ax.boxplot(BP_TS_PFC, labels = labels)
outliers_TS_PFC = bp_TS_PFC['fliers'][0].get_ydata()

# WT PFC boxplots:
WT_PFC_con = WT_PFC[WT_PFC.treatment == 'control']['firing_rate']
WT_PFC_EE = WT_PFC[WT_PFC.treatment == 'EE']['firing_rate']
WT_PFC_EE_EGCG = WT_PFC[WT_PFC.treatment == 'EE+EGCG']['firing_rate']

BP_WT_PFC = [WT_PFC_con, WT_PFC_EE, WT_PFC_EE_EGCG]
labels = ['Control', 'EE', 'EE+EGCG']

fig, ax = plt.subplots()
bp_WT_PFC = ax.boxplot(BP_WT_PFC, labels = labels)
outliers_WT_PFC = bp_WT_PFC['fliers'][0].get_ydata()

# con PFC
labels_Genotype = ['WT', 'TS']
BP_con_PFC = [WT_PFC_con, TS_PFC_con]
fig, ax = plt.subplots()
bp_con_PFC = ax.boxplot(BP_con_PFC, labels = labels_Genotype)

# EE PFC
BP_EE_PFC = [WT_PFC_EE, TS_PFC_EE]
fig, ax = plt.subplots()
bp_EE_PFC = ax.boxplot(BP_EE_PFC, labels = labels_Genotype)

# EE+EGCG PFC
BP_EE_EGCG_PFC = [WT_PFC_EE_EGCG, TS_PFC_EE_EGCG]
fig, ax = plt.subplots()

bp_EE_EGCG_PFC = ax.boxplot(BP_EE_EGCG_PFC, labels = labels_Genotype)

# con HPC
labels_Genotype = ['WT', 'TS']
BP_con_HPC = [WT_HPC_con, TS_HPC_con]
fig, ax = plt.subplots()
bp_con_HPC = ax.boxplot(BP_con_HPC, labels = labels_Genotype)

# EE HPC
BP_EE_HPC = [WT_HPC_EE, TS_HPC_EE]
fig, ax = plt.subplots()
bp_EE_HPC = ax.boxplot(BP_EE_HPC, labels = labels_Genotype)

# EE+EGCG HPC
BP_EE_EGCG_HPC = [WT_HPC_EE_EGCG, TS_HPC_EE_EGCG]
fig, ax = plt.subplots()
bp_EE_EGCG_HPC = ax.boxplot(BP_EE_EGCG_HPC, labels = labels_Genotype)

def creating_excel():
    """
    This function uses xlsxwriter and pandas to create an excel file organizing the data already stored in a dictionary in the previous function.
    - Different sheets will be created for PFC, HPC and both (total).
    - Histograms plots will be created to have a general idea of the data.
    - Means for HPC, PFC and total will be created in which the mean for every group will be calculated, as well as the standard deviation and the standard error of the mean (this data will be used in another function to plot).
    """

    # we first create a dataframe with all our data
mydata = [[key, 
data_dict[key]['animal'], 
data_dict[key]['channel'], 
data_dict[key]['condition'], 
data_dict[key]['genotype'], 
data_dict[key]['spikes_total'], 
data_dict[key]['firing_rate'], 
data_dict[key]['seconds'], 
data_dict[key]['treatment']] for 
key in data_dict.keys()] 

mydata_clean = 
pd.DataFrame(mydata)[[1,2,3,4,5,6,7,8]] 

mydata_clean.columns = ['animal', 'channel', 'condition', 'genotype', 'spikes_total','firing_rate', 'seconds', 'treatment']

mydata_HPC = mydata_clean[mydata_clean['channel'].isin(['HPC1', 'HPC2', 'HPC3'])] #we select only the HPC 
mydata_PFC = mydata_clean[mydata_clean['channel'].isin(['PFC1', 'PFC2', 'PFC3'])] #we select only the PFC

#we create the excel file with three sheets: total, HPC, PFC. 
writer = pd.ExcelWriter('results_sorting.xlsx')
workbook = writer.book
mydata_clean.to_excel(writer, 'total') #we now stored the dataframes in an excel file in the sheet name 'total' 

mydata_HPC.to_excel(writer, 'HPC') #we store the HPC data in an excel sheet named HPC 
mydata_PFC.to_excel(writer, 'PFC') #we store the PFC data in an excel sheet named PFC

#Now we will create a chart for the total mean 
worksheet_total = writer.sheets['total']
#First, we calculate the mean of HPC and PFC 
mean_HPC = mydata_HPC['firing_rate'].mean(axis = 0) 
worksheet_total.write('J2', 'HPC') 
worksheet_total.write('J3', mean_HPC)

mean_PFC = mydata_PFC['firing_rate'].mean(axis = 0) 
worksheet_total.write('K2', 'PFC')

mydata_tidy.columns = ['animal', 'genotype', 'treatment']
tidy_dict = {}

#Now we fill the new dict we already created (tidy_dict) with the FR mean of HPC and PFC of every animal 
for element in mydata_HPC['animal']: 
tidy_dict[element] = {}

PFC_FR = mydata_PFC[mydata_PFC['animal'] == element].firing_rate.mean() 
HPC_FR = mydata_HPC[mydata_HPC['animal'] == element].firing_rate.mean() 

tidy_dict[element]['PFC_FR'] = PFC_FR

tidy_dict[element]['HPC_FR'] = HPC_FR 

FR = [[key, 
tidy_dict[key]['PFC_FR'], 
tidy_dict[key]['HPC_FR']] for 
key in tidy_dict.keys()]

FR_data = pd.DataFrame(FR)[[1,2]]
FR_data.columns = ['PFC_FR', 'HPC_FR']

#We concatenate the two dataframes and we store the result in the excel file. 
mydata_FR = pd.concat([mydata_tidy, FR_data], axis=1) 
mydata_FR.to_excel(writer, 'Total_means')

#Now we will create a new data frame with the desired data and store in a new worksheet 
#we first create a DataFrame only with animal, genotype and treatment and we delete the repeating animals. 
mydata_tidy = pd.DataFrame(mydata)[[1,4,8]]
mydata_tidy = mydata_tidy.drop_duplicates().reset_index(drop=True)
worksheet_total.write('K3', mean_PFC)

    # Now we calculate the standard deviation and the error
    sdev_HPC = mydata_HPC['firing_rate'].std()
    worksheet_total.write('J4', sdev_HPC)
    sdev_PFC = mydata_PFC['firing_rate'].std()
    worksheet_total.write('J4', sdev_PFC)
    sem_HPC = mydata_HPC['firing_rate'].sem()
    worksheet_total.write('K4', sem_HPC)
    sem_PFC = mydata_PFC['firing_rate'].sem()
    worksheet_total.write('K4', sem_PFC)

    # Now we create the chart
    chart_mean = workbook.add_chart({'type': 'column'})
    chart_mean.add_series({  # We add the data to the chart
        'values': '=total!$J$3:$K$3',
        'categories': '=total!$J$2:$K$2',
        'y_error_bars': {  # We add the custom error bars
            'type': 'custom',
            'plus_values': '=total!$J$4:$K$4',
            'minus_values': '=total!$J$4:$K$4',
        },
    })
    chart_mean.set_x_axis({  # We add the name of the x axis
        'name_font': {'size': 11, 'bold': True},
        'num_font': {'italic': True},
    })
    chart_mean.set_y_axis({  # We add the name of the y axis
        'name': 'Mean of spikes',
        'name_font': {'size': 11, 'bold': True},
        'num_font': {'italic': True},
    })
    chart_mean.set_title({'name': 'PFC and HPC mean',
                           'name_font': {'name': 'Calibri',
                                         'color': 'black',}})
    worksheet_total.insert_chart('K6', chart_mean)

    # Now we create a chart with the HPC results.
    chart_HPC = workbook.add_chart({'type': 'column'})
    worksheet_HPC = writer.sheets['HPC']
    i = len(mydata_HPC)  # We establish a variable i that will be the amount of elements we have in the HPC data
    chart_HPC.add_series({  # We add the data to the chart
        'categories': ['HPC', 1, 6, i, 6],
        'values': ['HPC', 1, 1, i, 1],
    })
    chart_HPC.set_x_axis({  # We add the name of the x axis
        'name': 'Number of spikes',
        'name_font': {'size': 11, 'bold': True},
        'num_font': {'italic': True},
    })
    chart_HPC.set_y_axis({  # We add the name of the y axis
        'name': 'HPC results',
        'name_font': {'name': 'Calibri',
                      'color': 'black',}}
    })
    chart_HPC.set_title({'name': 'HPC results',
                          'name_font': {'name': 'Calibri',
                                        'color': 'black',}})
# Now we do the same for the PFC results.
chart_PFC = workbook.add_chart({'type': 'column'})
worksheet_PFC = writer.sheets['PFC']
e = len(mydata_PFC)
chart_PFC.add_series({'values': ['PFC', 1, 6, e, 6],
                     'categories': ['PFC', 1, 1, e, 1]})
chart_PFC.set_x_axis({'name_font': {'size': 11, 'bold': True},
                      'num_font': {'italic': True}})
chart_PFC.set_legend({'none': True})
chart_PFC.set_y_axis({'name': 'Number of spikes',
                      'name_font': {'size': 11, 'bold': True},
                      'num_font': {'italic': True}},
                     'name': 'PFC results',
                     'name_font': {'name': 'Calibri',
                                   'color': 'black',
                                   'black'}})
worksheet_PFC.insert_chart('J3', chart_PFC)

# Now we calculate the means of the firing rate for each group.
worksheet_means1 = workbook.add_worksheet('Means_HPC')
worksheet_means1.write(0, 0, 'HPC')
worksheet_means1.write(0, 1, 'Control')
worksheet_means1.write(0, 2, 'EE')
worksheet_means1.write(0, 3, 'EE+EGCG')
worksheet_means1.write(0, 4, 'Total')
worksheet_means1.write(1, 0, 'WT')
worksheet_means1.write(2, 0, 'TS')
worksheet_means1.write(3, 0, 'Total')
worksheet_means1.write(6, 0, 'STD')
worksheet_means1.write(6, 1, 'EE')
worksheet_means1.write(6, 2, 'EE+EGCG')

TS = mydata_HPC[mydata_HPC.genotype == 'TS']
WT = mydata_HPC[mydata_HPC.genotype == 'WT']
TS[TS.treatment == 'control'].firing_rate.mean()
worksheet_means1.write(1, 1, WT[WT.treatment == 'control'].firing_rate.mean())
worksheet_means1.write(2, 1, TS[TS.treatment == 'control'].firing_rate.mean())
worksheet_means1.write(1, 2, WT[WT.treatment == 'EE'].firing_rate.mean())
worksheet_means1.write(2, 2, TS[TS.treatment == 'EE'].firing_rate.mean())
worksheet_means1.write(1, 3, WT[WT.treatment == 'EE+EGCG'].firing_rate.mean())
worksheet_means1.write(2, 3, TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())
worksheet_means1.write(1, 4, WT[WT.treatment == 'EE+EGCG'].firing_rate.mean())
worksheet_means1.write(2, 4, TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())
worksheet_means1.write(3, 1, mydata_clean[mydata_clean.treatment == 'control'].firing_rate.mean())
worksheet_means1.write(3, 2, mydata_clean[mydata_clean.treatment == 'EE'].firing_rate.mean())
worksheet_means1.write(3, 3, mydata_clean[mydata_clean.treatment == 'EE+EGCG'].firing_rate.mean())

# Now we do the same to calculate the std:
worksheet_means1.write(6, 0, 'STD')
worksheet_means1.write(6, 1, 'Control')
worksheet_means1.write(6, 2, 'EE')
<table>
<thead>
<tr>
<th>Worksheet</th>
<th>WT</th>
<th>TS</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```python
### Worksheet 1

```python
worksheet_means1.write(6, 3, 'EE+EGCG')
worksheet_means1.write(6, 4, 'Total')
worksheet_means1.write(7, 0, 'WT')
worksheet_means1.write(8, 0, 'TS')
worksheet_means1.write(9, 0, 'Total')

worksheet_means1.write(7, 1, WT['WT.treatment == 'control'].firing_rate.std())
worksheet_means1.write(8, 1, TS['TS.treatment == 'control'].firing_rate.std())
worksheet_means1.write(7, 2, WT['WT.treatment == 'EE'].firing_rate.std())
worksheet_means1.write(8, 2, TS['TS.treatment == 'EE'].firing_rate.std())
worksheet_means1.write(7, 3, WT['WT.treatment == 'EE+EGCG'].firing_rate.std())
worksheet_means1.write(8, 3, TS['TS.treatment == 'EE+EGCG'].firing_rate.std())
worksheet_means1.write(7, 4, WT['firing_rate.std()'])
worksheet_means1.write(8, 4, TS['firing_rate.mean()'])
worksheet_means1.write(9, 1, mydata_clean['mydata_clean.treatment == 'control'].firing_rate.std())
worksheet_means1.write(9, 2, mydata_clean['mydata_clean.treatment =='EE'].firing_rate.std())
worksheet_means1.write(9, 3, mydata_clean['mydata_clean.treatment =='EE+EGCG'].firing_rate.std())

# Now we do the same for the PFC

worksheet_means2 = workbook.add_worksheet('Means_PFC')
worksheet_means2.write(0, 0, 'PFC')
worksheet_means2.write(0, 1, 'Control')
worksheet_means2.write(0, 2, 'EE')
worksheet_means2.write(0, 3, 'EE+EGCG')
worksheet_means2.write(0, 4, 'Total')
worksheet_means2.write(15, 0, 'WT')
```
**Worksheet Means and Standard Deviation**

<table>
<thead>
<tr>
<th></th>
<th>WT</th>
<th>TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EE+EGCG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Worksheet Means**

```python
worksheet_means2.write(0, 3, 'EE+EGCG')
worksheet_means2.write(0, 4, 'Total')
worksheet_means2.write(1, 0, 'WT')
worksheet_means2.write(1, 1, WT[WT.treatment == 'control'].firing_rate.mean())
worksheet_means2.write(1, 2, WT[WT.treatment == 'EE'].firing_rate.mean())
worksheet_means2.write(1, 3, WT[WT.treatment == 'EE+EGCG'].firing_rate.mean())
worksheet_means2.write(1, 4, WT.firing_rate.mean())
worksheet_means2.write(2, 0, 'TS')
worksheet_means2.write(2, 1, TS[TS.treatment == 'control'].firing_rate.mean())
worksheet_means2.write(2, 2, TS[TS.treatment == 'EE'].firing_rate.mean())
worksheet_means2.write(2, 3, TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())
worksheet_means2.write(2, 4, TS.firing_rate.mean())
worksheet_means2.write(3, 0, 'Total')
worksheet_means2.write(3, 1, mydata_clean[mydata_clean.treatment == 'control'].firing_rate.std())
worksheet_means2.write(3, 2, mydata_clean[mydata_clean.treatment == 'EE'].firing_rate.std())
worksheet_means2.write(3, 3, mydata_clean[mydata_clean.treatment == 'EE+EGCG'].firing_rate.std())
```

**Worksheet Std**

```python
worksheet_means2.write(6, 0, 'STD')
worksheet_means2.write(6, 1, 'Control')
worksheet_means2.write(6, 2, 'EE')
worksheet_means2.write(6, 3, 'EE+EGCG')
worksheet_means2.write(6, 4, 'Total')
worksheet_means2.write(7, 0, 'WT')
worksheet_means2.write(8, 0, 'TS')
worksheet_means2.write(9, 0, 'Total')
worksheet_means2.write(10, 0, 'STD')
worksheet_means2.write(10, 1, 'Control')
worksheet_means2.write(10, 2, 'EE')
worksheet_means2.write(10, 3, 'EE+EGCG')
```

**Worksheet SEM**

```python
worksheet_means2.write(14, 0, 'SEM')
worksheet_means2.write(14, 1, 'Control')
worksheet_means2.write(14, 2, 'EE')
worksheet_means2.write(14, 3, 'EE+EGCG')
```

---

**Table 1:**

<table>
<thead>
<tr>
<th></th>
<th>WT</th>
<th>TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EE+EGCG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2:**

<table>
<thead>
<tr>
<th></th>
<th>WT</th>
<th>TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EE+EGCG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Code:**

```python
TS = mydata_PFC[mydata_PFC.genotype == 'TS']
WT = mydata_PFC[mydata_PFC.genotype == 'WT']
```
worksheet_means2.write(14, 4, 'Total')
worksheet_means2.write(15, 0, 'WT')
worksheet_means2.write(16, 0, 'TS')
worksheet_means2.write(17, 0, 'Total')

worksheet_means2.write(15, 1, WT[WT.treatment == 'control'].firing_rate.sem())
worksheet_means2.write(16, 1, TS[TS.treatment == 'control'].firing_rate.sem())
worksheet_means2.write(15, 2, WT[WT.treatment == 'EE'].firing_rate.sem())
worksheet_means2.write(16, 2, TS[TS.treatment == 'EE'].firing_rate.sem())
worksheet_means2.write(15, 3, WT[WT.treatment == 'EE+EGCG'].firing_rate.sem())
worksheet_means2.write(16, 3, TS[TS.treatment == 'EE+EGCG'].firing_rate.sem())
worksheet_means2.write(15, 4, WT.firing_rate.sem())
worksheet_means2.write(16, 4, TS.firing_rate.sem())
worksheet_means2.write(17, 1, mydata_clean[mydata_clean.treatment == 'control'].firing_rate.sem())
worksheet_means2.write(17, 2, mydata_clean[mydata_clean.treatment == 'EE'].firing_rate.sem())
worksheet_means2.write(17, 3, mydata_clean[mydata_clean.treatment == 'EE+EGCG'].firing_rate.sem())

writer.save()

def creating_plots():
    ""
    This function is used to create barplots for every condition we considered using matplotlib:
    - TS and WT HPC
    - TS and WT PFC
    - Different treatments in PFC TS and WT
    - Different treatments in HPC TS and WT
    Error bars were considered using the sem already calculated in the previous function.
    ""
    # First we create a plot comparing WT treatment in PFC:
    TS = mydata_PFC[mydata_PFC.genotype == 'TS']
    WT = mydata_PFC[mydata_PFC.genotype == 'WT']
    n_groups = 2
    fig, ax = plt.subplots()
    index = np.arange(n_groups)
    bar_width = 0.25
    opacity = 0.7
    means_PFC_WT_con = (WT[WT.treatment == 'control'].firing_rate.mean(), WT[WT.treatment == 'control'].firing_rate.mean())
    means_PFC_WT_EE_EGCG = (WT[WT.treatment == 'EE'].firing_rate.mean(), WT[WT.treatment == 'EE'].firing_rate.mean())
    means_PFC_WT_con_sem = (WT[WT.treatment == 'control'].firing_rate.sem(), WT[WT.treatment == 'control'].firing_rate.sem())
    means_PFC_WT_EE_EGCG_sem = (WT[WT.treatment == 'EE'].firing_rate.sem(), WT[WT.treatment == 'EE'].firing_rate.sem())
    plot1 = ax.bar(index, means_PFC_WT_con, bar_width, alpha=opacity, color='black', label='control', yerr=means_PFC_WT_con_sem, error_kw = {'capsize':2})
    plot2 = ax.bar(index+bar_width, means_PFC_WT_EE_EGCG, bar_width, alpha=opacity,
Final Degree Project
Vilademunt, 2018

```python
color='grey',
yerr=means_PFC_WT_EE_EGCG_sem,
    error_kw =
    {'capsize':2})

plt.ylabel('Firing rate mean (spikes/second)')
plt.xlabel('Treatment')
plt.title('PFC neural activity in WT mice')
plt.xticks(index+bar_width/2., ('EE', 'EE+EGCG'))
plt.legend()

plt.tight_layout()
plt.show()

#Now we do the same for TS treatment in PFC:

TS = mydata_PFC[mydata_PFC.genotype == 'TS']
fig, ax = plt.subplots()
index = np.arange(n_groups)
bar_width = 0.25
opacity = 0.7

means_PFC_TS_con =
    (TS[TS.treatment == 'control'].firing_rate.mean(),
    TS[TS.treatment == 'control'].firing_rate.mean())
means_PFC_TS_EE_EGCG =
    (TS[TS.treatment == 'EE'].firing_rate.mean(),
    TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())

means_PFC_TS_con_sem =
    (TS[TS.treatment == 'control'].firing_rate.mean(),
    TS[TS.treatment == 'control'].firing_rate.mean())
means_PFC_TS_EE_EGCG_sem =
    (TS[TS.treatment == 'EE'].firing_rate.mean(),
    TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())

plot1 = ax.bar(index,
    means_PFC_TS_con, bar_width,
    alpha=opacity,
    color='black',
    label='control',
yerr=means_PFC_TS_con_sem,
    error_kw =
    {'capsize':2})

plot2 =
    ax.bar(index+bar_width,
    means_PFC_TS_EE_EGCG, bar_width,
    alpha=opacity,
    color='grey',
yerr=means_PFC_TS_EE_EGCG_sem,
    error_kw =
    {'capsize':2})

plt.ylabel('Firing rate mean (spikes/second)')
plt.xlabel('Treatment')
plt.title('PFC neural activity in TS mice')
plt.xticks(index+bar_width/2., ('EE', 'EE+EGCG'))
plt.legend()

plt.tight_layout()
plt.show()

#Now we compare the three treatments in PFC:

n_groups = 3
TS = mydata_PFC[mydata_PFC.genotype == 'TS']
fig, ax = plt.subplots()
index = np.arange(n_groups)
bar_width = 0.25
opacity = 0.7

means_PFC_WT =
    (WT[WT.treatment == 'control'].firing_rate.mean(),
    WT[WT.treatment == 'EE'].firing_rate.mean(),
    WT[WT.treatment == 'EE+EGCG'].firing_rate.mean())
means_PFC_TS =
    (TS[TS.treatment == 'control'].firing_rate.mean(),
    TS[TS.treatment == 'EE'].firing_rate.mean(),
    TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())
means_PFC_TS =
    (TS[TS.treatment == 'control'].firing_rate.mean(),
    TS[TS.treatment == 'EE'].firing_rate.mean(),
    TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())

plot1 = ax.bar(index,
    means_PFC_WT, bar_width,
    alpha=opacity,
    color='black',
    label='control',
yerr=means_PFC_WT_sem,
    error_kw =
    {'capsize':2})

plot2 =
    ax.bar(index+bar_width,
    means_PFC_TS, bar_width,
    alpha=opacity,
    color='grey',
yerr=means_PFC_TS_sem,
    error_kw =
    {'capsize':2})

plt.ylabel('Firing rate mean (spikes/second)')
plt.xlabel('Treatment')
plt.title('PFC neural activity in TS mice')
plt.xticks(index+bar_width/2., ('EE', 'EE+EGCG'))
plt.legend()

plt.tight_layout()
plt.show()
```

12
'EE'].firing_rate.mean(),
TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())

means_PFC_WT_sem = (WT[WT.treatment == 'control'].firing_rate.sem(),
WT[WT.treatment == 'EE'].firing_rate.sem(),
WT[WT.treatment == 'EE+EGCG'].firing_rate.sem())
means_PFC_TS_sem = (TS[TS.treatment == 'control'].firing_rate.sem(),
TS[TS.treatment == 'EE'].firing_rate.sem(),
TS[TS.treatment == 'EE+EGCG'].firing_rate.sem())

plot1 = ax.bar(index, means_PFC_WT, bar_width, alpha=opacity, color='black', label='WT', yerr=means_PFC_WT_sem, error_kw = {'capsize':2})
plot2 = ax.bar(index+bar_width, means_PFC_TS, bar_width, alpha=opacity, color='grey', label = 'TS', yerr=means_PFC_TS_sem, error_kw = {'capsize':2})

plt.ylabel('Firing rate mean (spikes/second)')
plt.xlabel('Treatment')
plt.title('PFC neural activity')
plt.xticks(index+bar_width/2., ('Control', 'EE', 'EE+EGCG'))
plt.legend()
plt.tight_layout()
plt.show()

#Now we do the same for HPC:

n_groups = 2
TS = mydata_HPC[mydata_HPC.genotype == 'TS']
WT = mydata_HPC[mydata_HPC.genotype == 'WT']

means_HPC_WT_con = (WT[WT.treatment == 'control'].firing_rate.mean(),
WT[WT.treatment == 'EE'].firing_rate.mean(),
WT[WT.treatment == 'EE+EGCG'].firing_rate.mean())
means_HPC_WT_EE_EGCG = (WT[WT.treatment == 'control'].firing_rate.mean(),
WT[WT.treatment == 'EE'].firing_rate.mean(),
WT[WT.treatment == 'EE+EGCG'].firing_rate.mean())

plot1 = ax.bar(index, means_HPC_WT_con, bar_width, alpha=opacity, color='black', label='control', yerr=means_HPC_WT_con_sem, error_kw = {'capsize':2})
plot2 = ax.bar(index+bar_width, means_HPC_WT_EE_EGCG, bar_width, alpha=opacity, color='grey', label = 'EE+EGCG', yerr=means_HPC_WT_EE_ERGCG_sem, error_kw = {'capsize':2})

plt.xlabel('Treatment')
plt.ylabel('Firing rate mean (spikes/second)')
plt.title('HPC neural activity')
plt.xticks(index+bar_width/2., ('Control', 'EE', 'EE+EGCG'))
plt.legend()
plt.tight_layout()
plt.show()
plt.ylabel('Firing rate mean (spikes/second)')
plt.xlabel('Treatment')
plt.title('HPC neural activity in WT mice')
plt.xticks(index+bar_width/2., ('EE', 'EE+EGCG'))
plt.legend()

plt.tight_layout()
plt.show()

#Now we do the same for TS treatment in HPC:

TS = mydata_HPC[mydata_HPC.genotype == 'TS']
fig, ax = plt.subplots()
index = np.arange(n_groups)
bar_width = 0.25
opacity = 0.7

means_HPC_TS_con = (TS[TS.treatment == 'control'].firing_rate.mean(),
                      TS[TS.treatment == 'control'].firing_rate.mean())
means_HPC_TS_EE_EGCG = (TS[TS.treatment == 'EE'].firing_rate.mean(),
                       TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())

means_HPC_TS_con_sem = (TS[TS.treatment == 'control'].firing_rate.sem(),
                        TS[TS.treatment == 'control'].firing_rate.sem())
means_HPC_TS_EE_EGCG_sem = (TS[TS.treatment == 'EE'].firing_rate.sem(),
                           TS[TS.treatment == 'EE+EGCG'].firing_rate.sem())

plot1 = ax.bar(index, means_HPC_TS_con, bar_width,
                alpha=opacity,
                color='black',
                label='control',
                yerr=means_HPC_TS_con_sem,
                error_kw = {'capsize':2})

plot2 = ax.bar(index+bar_width, means_HPC_TS_EE_EGCG, bar_width,
                alpha=opacity,
                color='grey',
                yerr=means_HPC_TS_EE_EGCG_sem,
                error_kw = {'capsize':2})

plt.ylabel('Firing rate mean (spikes/second)')
plt.xlabel('Treatment')
plt.title('HPC neural activity in TS mice')
plt.xticks(index+bar_width/2., ('EE', 'EE+EGCG'))
plt.legend()
plt.tight_layout()
plt.show()

#Now we compare the three treatments in HPC:

n_groups = 3
TS = mydata_HPC[mydata_HPC.genotype == 'TS']
fig, ax = plt.subplots()
index = np.arange(n_groups)
bar_width = 0.25
opacity = 0.7

means_HPC_WT = (WT[WT.treatment == 'control'].firing_rate.mean(),
                 WT[WT.treatment == 'EE'].firing_rate.mean(),
                 WT[WT.treatment == 'EE+EGCG'].firing_rate.mean())
means_HPC_TS = (TS[TS.treatment == 'control'].firing_rate.mean(),
                TS[TS.treatment == 'EE'].firing_rate.mean(),
                TS[TS.treatment == 'EE+EGCG'].firing_rate.mean())

means_HPC_WT_sem = (WT[WT.treatment == 'control'].firing_rate.sem(),
                    WT[WT.treatment == 'EE'].firing_rate.sem(),
                    WT[WT.treatment == 'EE+EGCG'].firing_rate.sem())
means_HPC_TS_sem = (TS[TS.treatment == 'control'].firing_rate.sem(),
                    TS[TS.treatment == 'EE'].firing_rate.sem(),
                    TS[TS.treatment == 'EE+EGCG'].firing_rate.sem())

plot1 = ax.bar(index, means_HPC_WT, bar_width,
               alpha=opacity,
               color='red',
               label='WT',
               yerr=means_HPC_WT_sem,
               error_kw = {'capsize':2})

plot2 = ax.bar(index+bar_width, means_HPC_TS, bar_width,
               alpha=opacity,
               color='black',
               label='TS',
               yerr=means_HPC_TS_sem,
               error_kw = {'capsize':2})

plt.ylabel('Firing rate mean (spikes/second)')
plt.xlabel('Treatment')
plt.title('HPC neural activity in WT and TS mice')
plt.xticks(index+bar_width/2., ('EE', 'EE+EGCG'))
plt.legend()
plt.tight_layout()
plt.show()
WT[WT.treatment == 'EE+EGCG'].firing_rate.sem())
    means_HPC_TS_sem =
(TS[TS.treatment == 'control'].firing_rate.sem(),
TS[TS.treatment == 'EE'].firing_rate.sem(),
TS[TS.treatment == 'EE+EGCG'].firing_rate.sem())

    plot1 = ax.bar(index,
means_HPC_WT, bar_width,
alpha=opacity,
    label='WT',
    yerr=means_HPC_WT_sem,
    error_kw =
    {'capsize':2})

    plot2 = ax.bar(index+bar_width,
means_HPC_TS, bar_width,
alpha=opacity,
    color='grey',
    label = 'TS',
    yerr=means_HPC_TS_sem,
    error_kw =
    {'capsize':2})

plt.ylabel('Firing rate mean (spikes/second)')
plt.xlabel('Treatment')
plt.title('HPC neural activity')
plt.xticks(index+bar_width/2., ('Control', 'EE', 'EE+EGCG'))
plt.legend()
plt.tight_layout()
plt.show()
Supplementary figure 2.
Boxplots representing firing rate mean (in spikes/second) for every mice group considering treatment and genotype. Dots represent outliers considered as the third Q3 + whis*IQR or Q1 – whis*IQR, where Q3 refers to the third quartile, Q1 refers to the first quartile, IQR refers to the interquartile range (Q3-Q1) and whis is a factor determined as 1.5. Figure A represents PFC neural activity in control (N=3), environmental enriched (EE) (N=6) or double treated (EE+EGCG) (N=3) Ts65Dn (TS) mice. Figure B represents PFC neural activity in control (N=3), environmental enriched (EE) (N=3) or double treated (EE+EGCG) (N=3) wild type (WT) mice. Figure C represents PFC neural activity comparing WT and TS mice between the different treatments. Figure D represents HPC neural activity in control (N=3), environmental enriched (EE) (N=6) or double treated (EE+EGCG) (N=3) Ts65Dn (TS) mice. Figure E represents HPC neural activity in control (N=3), environmental enriched (EE) (N=3) or double treated (EE+EGCG) (N=3) wild type (WT) mice. Figure F represents HPC neural activity comparing WT and TS mice between the different treatments.
Supplementary Figure 3. Two-way analyses of variance (ANOVA) results. Statistical significance across the two factors genotype and treatment was determined for PFC firing rate and HPC firing rate (spikes/seconds) in two different analyses (corresponding to Figure A for PFC and B for HPC). Statistical significance was considered when the P value had a value under 0.05. The script of the programme needed to be changed in the beginning of the analyses as indicated. IBM SPSS (https://www.ibm.com/analytics/es/es/technology/spss/) was used.

A Two-way ANOVA results for PFC firing rate mean:

```
DATASET ACTIVATE Conjunto_de_datos0.
UNIANOVA PFC_FR BY Genotype Treatment
  /METHOD=STYPE(3)
UNIANOVA HPC_FR BY Genotype Treatment
  /METHOD=STYPE(3)
  /INTERCEPT=INCLUDE
  /POSTHOC=Treatment(TUKEY BONFERRONI)
  /PLOT=PROFILE(Treatment*Genotype)
  /EMMEANS=TABLES(Genotype*Treatment) COMPARE(Treatment) ADJ(LSD)
  /EMMEANS=TABLES(Genotype) COMPARE ADJ(LSD)
  /PRINT=DESCRIPTIVE PARAMETER
  /CRITERIA=ALPHA(0.05)
  /DESIGN=Genotype Treatment Genotype*Treatment.
```

A Two-way ANOVA results for PFC firing rate mean:

```
DATASET ACTIVATE Conjunto_de_datos0.
UNIANOVA PFC_FR BY Genotype Treatment
  /METHOD=STYPE(3)
UNIANOVA HPC_FR BY Genotype Treatment
  /METHOD=STYPE(3)
  /INTERCEPT=INCLUDE
  /POSTHOC=Treatment(TUKEY BONFERRONI)
  /PLOT=PROFILE(Treatment*Genotype)
  /EMMEANS=TABLES(Genotype*Treatment) COMPARE(Treatment) ADJ(LSD)
  /EMMEANS=TABLES(Genotype) COMPARE ADJ(LSD)
  /PRINT=DESCRIPTIVE PARAMETER
  /CRITERIA=ALPHA(0.05)
  /DESIGN=Genotype Treatment Genotype*Treatment.
```

Pruebas de efectos inter-sujetos

<table>
<thead>
<tr>
<th>Variable dependiente: PFC_FR</th>
<th>Tipo III de suma de cuadrados</th>
<th>gl</th>
<th>Cuadrático promedio</th>
<th>F</th>
<th>Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modelo corregido</td>
<td>1156.464^a</td>
<td>5</td>
<td>231.293</td>
<td>1.777</td>
<td>.178</td>
</tr>
<tr>
<td>Intercepción</td>
<td>58844.653</td>
<td>1</td>
<td>58844.653</td>
<td>452.111</td>
<td>.000</td>
</tr>
<tr>
<td>Genotipo</td>
<td>40.249</td>
<td>1</td>
<td>40.249</td>
<td>.069</td>
<td>.896</td>
</tr>
<tr>
<td>Tratamiento</td>
<td>53.602</td>
<td>2</td>
<td>26.801</td>
<td>.208</td>
<td>.816</td>
</tr>
<tr>
<td>Genotipo * Tratamiento</td>
<td>1074.704</td>
<td>2</td>
<td>537.352</td>
<td>4.129</td>
<td>.037</td>
</tr>
<tr>
<td>Error</td>
<td>1952.933</td>
<td>16</td>
<td>122.516</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>65777.585</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total corregido</td>
<td>81085.710</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a. R al cuadrado = .372 (R al cuadrado ajustado = .163)

Comparaciones por parejas

<table>
<thead>
<tr>
<th>Treatment (I) Genotype (J) Genotype</th>
<th>Diferencia de medias (I-J)</th>
<th>Error estándar</th>
<th>Sig b</th>
<th>95% de intervalo de confianza para diferencia^b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Límite inferior</td>
</tr>
<tr>
<td>control TS WT</td>
<td>13.701</td>
<td>9.315</td>
<td>.162</td>
<td>-33.556</td>
</tr>
<tr>
<td>WT TS</td>
<td>-13.701</td>
<td>9.315</td>
<td>.162</td>
<td>-33.556</td>
</tr>
<tr>
<td>EE TS WT</td>
<td>1.106</td>
<td>8.067</td>
<td>.893</td>
<td>-16.089</td>
</tr>
<tr>
<td>WT EE TS</td>
<td>-1.106</td>
<td>8.067</td>
<td>.893</td>
<td>-18.089</td>
</tr>
<tr>
<td>EE+EGCG TS WT</td>
<td>-22.397^a</td>
<td>9.315</td>
<td>.024</td>
<td>-43.251</td>
</tr>
<tr>
<td>WT EE+EGCG TS</td>
<td>22.397^a</td>
<td>9.315</td>
<td>.024</td>
<td>3.542</td>
</tr>
</tbody>
</table>

Se basa en medias marginales estimadas

* La diferencia de medias es significativa en el nivel .05.

b. Ajuste para varias comparaciones: menor diferencia significativa (equivalente a sin ajustes).
Two-way ANOVA results for HPC firing rate mean:

```plaintext
DATASET ACTIVATE Conjunto_de_datos0.
UNIANOVA PFC_FR BY Genotype Treatment
/ METHOD=SSTYPE(3)
UNIANOVA HPC_FR BY Genotype Treatment
/ METHOD=SSTYPE(3)
/INTERCEPT=INCLUDE
/POSTHOC=Treatment(TUKEY BONFERRONI)
/ PLOT=PROFILE(Treatment*Genotype)
/EMMEANS=TABLES(Genotype*Treatment) COMPARE(Genotype) ADJ(LSD)
/EMMEANS=TABLES(Genotype) COMPARE ADJ(LSD)
/PRINT=DESCRIPTIVE PARAMETER
/CRITERIA=ALPHA(.05)
/DESIGN=Genotype Treatment Genotype*Treatment.
```

Pruebas de efectos inter-sujetos

<table>
<thead>
<tr>
<th>Origen</th>
<th>Tipo III de sumas de cuadrados</th>
<th>gl</th>
<th>Cuadrático promedio</th>
<th>F</th>
<th>Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modelo corregido</td>
<td>4342,367a</td>
<td>5</td>
<td>868,473</td>
<td>3,113</td>
<td>.040</td>
</tr>
<tr>
<td>Intercepción</td>
<td>116146,266</td>
<td>1</td>
<td>116146,266</td>
<td>416,327</td>
<td>.000</td>
</tr>
<tr>
<td>Genotipo</td>
<td>1111,853</td>
<td>1</td>
<td>1111,853</td>
<td>3,985</td>
<td>.064</td>
</tr>
<tr>
<td>Tratamiento</td>
<td>2797,971</td>
<td>2</td>
<td>1393,906</td>
<td>4,997</td>
<td>.022</td>
</tr>
<tr>
<td>Genotipo*Treatment</td>
<td>559,894</td>
<td>2</td>
<td>279,497</td>
<td>1,003</td>
<td>.390</td>
</tr>
<tr>
<td>Error</td>
<td>4194,975</td>
<td>15</td>
<td>278,978</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>127337,756</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total corregido</td>
<td>8527,042</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*a. R al cuadrado = .509 (R al cuadrado ajustado = .340)
### Comparaciones por parejas

**Variable dependiente: HPC FR**

<table>
<thead>
<tr>
<th>Treatment</th>
<th>(I) Genotype</th>
<th>(J) Genotype</th>
<th>Diferencia de medias (I-J)</th>
<th>Error estándar</th>
<th>Sig.</th>
<th>95% de intervalo de confianza para diferencia a</th>
<th>Límite inferior</th>
<th>Límite superior</th>
</tr>
</thead>
<tbody>
<tr>
<td>control</td>
<td>TS</td>
<td>WT</td>
<td>28,515</td>
<td>13,638</td>
<td>.054</td>
<td>-57,583</td>
<td>-55,53</td>
<td>57,583</td>
</tr>
<tr>
<td></td>
<td>WT</td>
<td>TS</td>
<td>-28,515</td>
<td>13,638</td>
<td>.054</td>
<td>-57,583</td>
<td>-55,53</td>
<td>57,583</td>
</tr>
<tr>
<td>EE</td>
<td>TS</td>
<td>WT</td>
<td>2,958</td>
<td>11,811</td>
<td>.806</td>
<td>-22,215</td>
<td>-26,132</td>
<td>22,215</td>
</tr>
<tr>
<td>EE+EGCG</td>
<td>TS</td>
<td>WT</td>
<td>13,675</td>
<td>13,638</td>
<td>.332</td>
<td>-42,743</td>
<td>-15,393</td>
<td>42,743</td>
</tr>
<tr>
<td></td>
<td>WT</td>
<td>TS</td>
<td>-13,675</td>
<td>13,638</td>
<td>.332</td>
<td>-42,743</td>
<td>-15,393</td>
<td>42,743</td>
</tr>
</tbody>
</table>

Se basa en medias marginales estimadas

*a. Ajuste para varias comparaciones: menor diferencia significativa (equivalente a sin ajustes).*

### Comparaciones por parejas

**Variable dependiente: HPC FR**

<table>
<thead>
<tr>
<th>Genotype</th>
<th>(I) Treatment</th>
<th>(J) Treatment</th>
<th>Diferencia de medias (I-J)</th>
<th>Error estándar</th>
<th>Sig.</th>
<th>95% de intervalo de confianza para diferencia a</th>
<th>Límite inferior</th>
<th>Límite superior</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td>control</td>
<td>EE</td>
<td>41,281</td>
<td>11,811</td>
<td>.003</td>
<td>66,455</td>
<td>16,103</td>
<td>66,455</td>
</tr>
<tr>
<td></td>
<td>EE+EGCG</td>
<td>24,433</td>
<td>13,638</td>
<td>.063</td>
<td>-4,565</td>
<td>53,551</td>
<td>-4,565</td>
<td>53,551</td>
</tr>
<tr>
<td></td>
<td>EE+EGCG</td>
<td>-16,798</td>
<td>11,811</td>
<td>.175</td>
<td>-31,972</td>
<td>8,375</td>
<td>-16,108</td>
<td>31,972</td>
</tr>
<tr>
<td>EE</td>
<td>16,796</td>
<td>11,811</td>
<td>.175</td>
<td>-8,375</td>
<td>41,972</td>
<td>-8,375</td>
<td>41,972</td>
<td></td>
</tr>
<tr>
<td>WT</td>
<td>control</td>
<td>EE</td>
<td>15,726</td>
<td>13,638</td>
<td>.267</td>
<td>41,972</td>
<td>-13,343</td>
<td>44,793</td>
</tr>
<tr>
<td></td>
<td>EE+EGCG</td>
<td>9,643</td>
<td>13,638</td>
<td>.490</td>
<td>-19,425</td>
<td>38,711</td>
<td>-19,425</td>
<td>38,711</td>
</tr>
<tr>
<td>EE</td>
<td>control</td>
<td>-15,726</td>
<td>13,638</td>
<td>.267</td>
<td>38,711</td>
<td>-44,793</td>
<td>13,343</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EE+EGCG</td>
<td>-6,822</td>
<td>13,638</td>
<td>.662</td>
<td>-35,150</td>
<td>22,988</td>
<td>-35,150</td>
<td>22,988</td>
</tr>
<tr>
<td>EE+EGCG</td>
<td>control</td>
<td>-6,822</td>
<td>13,638</td>
<td>.662</td>
<td>-22,988</td>
<td>19,425</td>
<td>-22,988</td>
<td>19,425</td>
</tr>
<tr>
<td>EE</td>
<td>6,822</td>
<td>13,638</td>
<td>.662</td>
<td>-35,150</td>
<td>22,988</td>
<td>-35,150</td>
<td>22,988</td>
<td></td>
</tr>
</tbody>
</table>

Se basa en medias marginales estimadas

*a. La diferencia de medias es significativa en el nivel 0,05*

*b. Ajuste para varias comparaciones: menor diferencia significativa (equivalente a sin ajustes).*