Ion Channels project - Milestone report

Berenice Dethier

Summary The goal of this Kaggle competition is to predict the number of open channels based on an electrophysiological signal measured by bioassay. In this document, we discuss how we explored and cleaned the signal, preparing for the next step: build models with ensemble algorithms and neural networks.

1 *Overview*

The Kaggle competition hosted by the University of Liverpool was a good fit for my next project: a deep learning approach is possible, it involves signal processing, is related to biochemistry, and has enough incentives to keep me motivated. The format was excellent for learning, as more experienced scientists were pitching in and sharing some of their findings, opening topics I wouldn't have had time to familiarize myself with. The goal of the competition was to predict the number of open channels based on the electric signal measured by a bioassay. An example of the signal and the number of open channels is presented Figure 1. The relevance of identifying the number of open channels is summarized by the host on the competition page:

Details about the competition can be found on Kaggle - Ion Switching Competition.

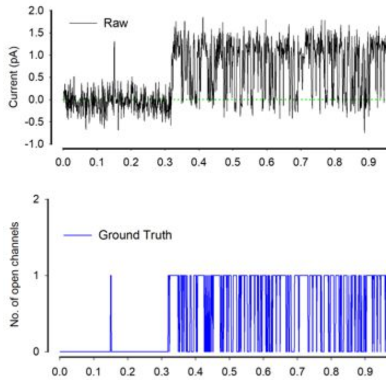

Figure 1: Example of a signal and number of open channels provided by the competition host, the University of Liverpool.

> "Many diseases, including cancer, are believed to have a contributing factor in common. Ion channels are pore-forming proteins present in animals and plants. They encode learning and memory, help fight infections, enable pain signals, and stimulate muscle contraction. If scientists could better study ion channels, which may be possible with the aid of machine learning, it could have a far-reaching impact.
>
> When ion channels open, they pass electric currents. Existing methods of detecting these state changes are slow and laborious. Humans must supervise the analysis, which imparts considerable bias, in addition to being tedious. These difficulties limit the volume of ion channel current analysis that can be used in research. Scientists hope that technology could enable rapid automatic detection of ion channel current events in raw data.
>
> The University of Liverpool's Institute of Ageing and Chronic Disease is working to advance ion channel research. Their team of scientists have asked for your help. In this competition, you'll use ion channel data to better model automatic identification methods. If successful, you'll be able to detect individual ion channel events in noisy raw signals. The data is simulated and injected with real world noise to emulate what scientists observe in laboratory experiments.
>
> Technology to analyze electrical data in cells has not changed significantly over the past 20 years. If we better understand ion channel activity, the research could impact many areas related to cell health and migration."

We started the exploratory data analysis (section 2) by looking at the signal and number of open channels in a few different situations, then took steps to clean the signal (section 3). Most of the work was performed with Jupyter Notebook IDE and Python 3.7, but the Kalman filter tuning was done with R (R Studio IDE).

The final report will include the models and their performances, as well as our score and conclusions. The project notebooks and other information can be found in the project GitHub repository.

2 *Exploratory Data Analysis*

Two data sets are intitially available, a training set consisting of 5,000,000 observations of the variables 'time', 'signal', and 'open channels', and a public test set consisting of 2,000,000 observations of 'time' and 'signal' only. A private test set is to be released on 05/18/2020. The code for the EDA was inspired by this notebook from Chris Deotte, and this notebook from Eunho Lee.

The documentation states:

> " While the time series appears continuous, the data is from discrete batches of 50 seconds long 10 kHz samples (500,000 rows per batch). In other words, the data from 0.0001 - 50.0000 is a different batch than 50.0001 - 100.0000, and thus discontinuous between 50.0000 and 50.0001."

We will therefore look at the batches as separate experiments.

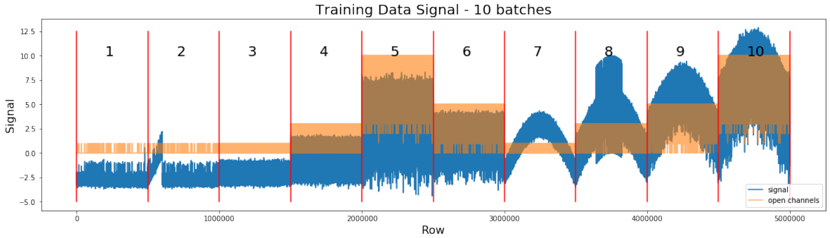

Figure 2: The signal for the train set consists of 10 batches of 50 s measurements. The sampling rate is 10 kHz, so each batch contains 500,000 observations. When compared to the number of open channels (Figure 3-6), we notice the signal is correlated with the number of open channels. Batches have different amplitudes and baselines. There is drift in multiple batches.

When the signal and number of open channels are plotted over time (Figure 2), we notice the following things:

- Four batches have a small amplitude and small baseline (batches 1-3 and 7), four batches have an intermediate amplitude and baseline (batches 4, 6, 8 and 9), and two have a larger amplitude and a baseline slightly larger, too (batches 5 and 10). These caracteristics of the signal are correlated with the number of open channels: 0-1 open channels for batches 1, 2, 3 and 7, 0-3 open channels for batches 4 and 8, 1-5 open channels for batches 6 and 9, and 2-10 open channels for batches 5 and 10.

- Among the batches with 0/1 open channels, two have a low frequency of opening (batches 1 and 2), and two have a high frequency of opening (batches 3 and 7).
- There is drift in five of the ten batches: batch 2 (first part only), and 7-10 (parabolic drift).
- There seems to be noise in the form of a thick baseline when no channels are open (batch 1, for example) and in the form of spikes in signal (see btach 8, for example).

We can identify 5 behaviors in ion channels opening thanks to these observations:

1. **Type 1s**: batch 1 and batch 2. There is up to one open channel, and the rate for opening and closing is slow (Figure 3).
2. **Type 1f**: batch 3 and batch 7. There is up to one open channel, and the rate for opening and closing is fast (Figure 4).
3. **Type 3**: batch 4 and batch 8. There are between 0 and 3 channels open (Figure 5).
4. **Type 5**: batch 6 and batch 9. There are between 1 and 5 channels open (Figure 6, left).
5. **Type 10**: batch 5 and batch 10. There are between 2 and 10 channels open (Figure 6, right).

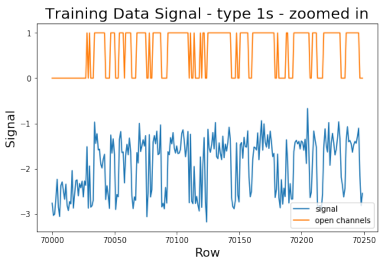

Figure 3: Example of signal and number of open channels typical of type **1s** signal.

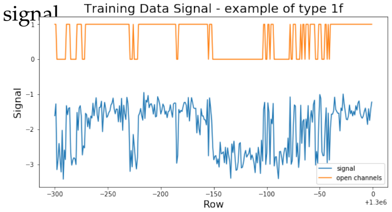

Figure 4: Example of a signal and number of open channels typical of **1f** signal.

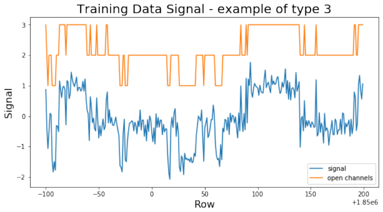

Figure 5: Example of a signal and number of open channels typical of type **3** signal.

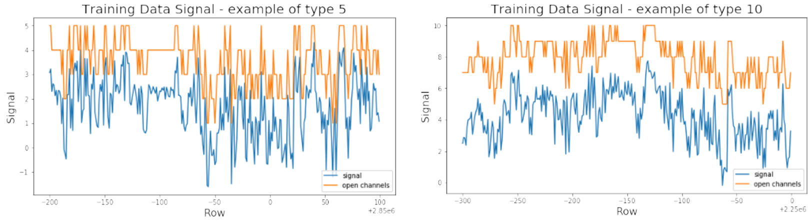

Figure 6: Example of a signal and number of open channels typical of type **5** signal (left) and of type **10** signal (right).

As a consequence, the following steps will need to be applied to the signal of the train and test set:

1. Remove the drift (areas to be processed will have to be identified visually)
2. Try applying filters to reduce the noise.

The distribution of observations depending on the number of open channels (Figure 7) indicates that zero open channels is the most common configuration (24.8% of the observations in that configuration) , followed by 1 (19.7%), 3 (13.4%), 2 (11.1%), 4 (8.1%), 5 (5.6%), 7 (5.3%), 6 (3.8%), 8 (4.9%), 9 (2.7%) then 10 open channels (0.7%).

We investigated the openings and closings of channels. A change in number of open channels will be called "event", and the number of timesteps the system remains in a certain state between two events will be called "interval". The distribution of intervals per number of open channels is summarized Figure 8. Three (3) open channels is the most common type of interval, then 2, 4, 7, 8, 1, 5, 6, 9, 0 and 10.

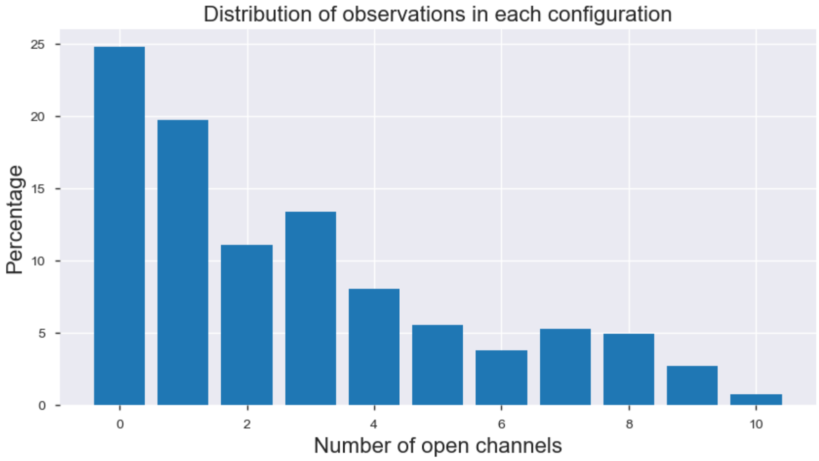

Figure 7: Zero open channels is the most common state, then 1, 3, 2, 4, 5, 7, 6, 8, 9, then 10 open channels.

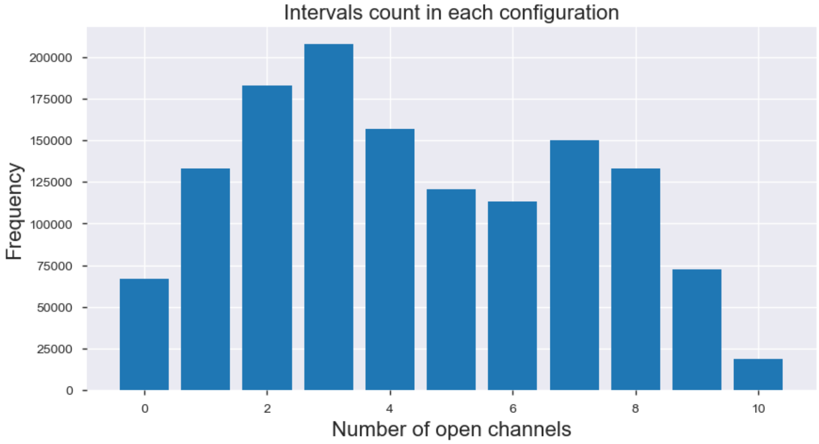

Figure 8: Three (3) open channels has the highest count of intervals, followed by 2, 4, 7, 8, then 1 open channels. There are less intervals for configurations when 5, 6, 9, 0 then 10 channels are open.

To understand how the intervals are distributed, we plotted the count for each interval length on Figure 9. The ECDF is presented

Figure 10.

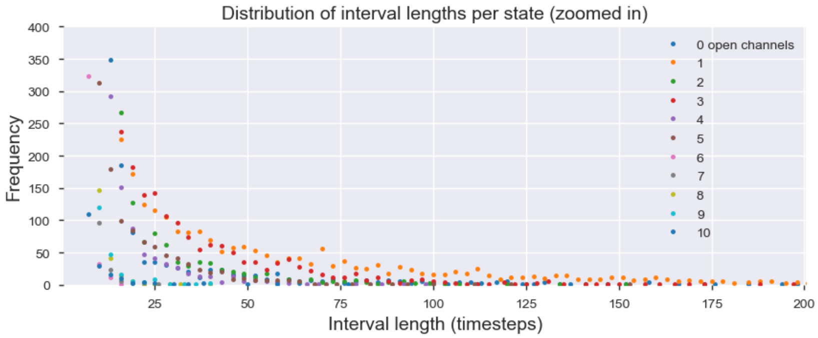

Figure 9: Most of the time, the system remains in one state for a short time (higher frequency for the short durations). There is more variability and longer intervals in the two following states: one channel open (orange trace) and three channels open (red trace).

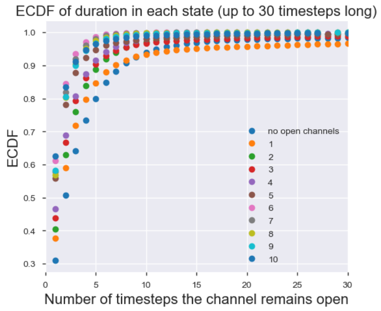

Figure 10: The observations are the same as for Figure 9: usually the intervals are short, and there are longer intervals for systems with one channel open (orange trace) and three channels open (red trace).

Then, we looked at the boxplots (Figure 11). We see that 75% of intervals last between one and 5 timesteps (0 open channels), between one and 4 timesteps (1 open channel), between one and 3 timesteps (2-4 open channels), or between one and 2 timesteps (5-10 open channels). There are numerous outliers, with relatively high values (especially for 0 open channels: intervals were as long as 24000 timesteps).

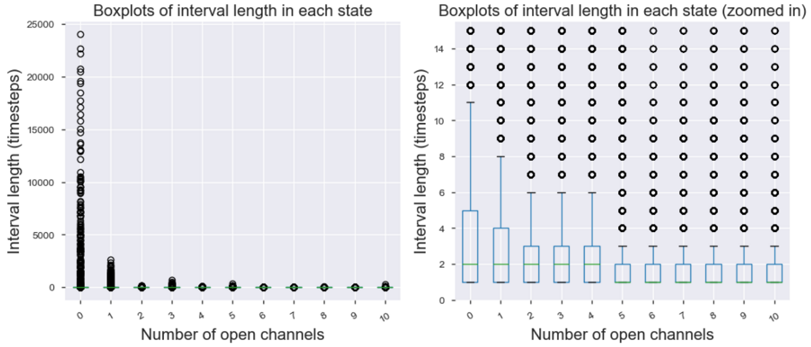

Figure 11: Boxplots of the intervals distribution per configuration.

Other statistics regarding the intervals are presented Table 1. These observations are consistent with the profiles of each of the five batches presented Figure 3-6. Batches 1s, 1f and 3, with low numbers of open channels, have long intervals. Batches 5 and 10, with the higher numbers of open channels, have shorter intervals/more frequent events.

Finally, the events type and frequency were plotted (Figure 12) to see which transitions were most common. Most timesteps (72.8%) are not events (difference in state is zero), and the distribution is very symetrical: -1 and 1 transitions each represent 11.6% of the timesteps, -2 and 2 represent 1.7%, and so on. A jump of 3, 4, and 5 number

	Number of open channels				
	0	**1**	**2**	**3**	**4**
Count	66626	133244	183147	207870	157010
Mean	18.6	7.4	3.0	3.2	2.6
Std	381.7	43.7	4.6	7.5	3.2
Min	1	1	1	1	1
25%	1	1	1	1	1
50%	2	2	2	2	2
75%	5	4	3	3	3
Max	24024	2596	168	685	101

	Number of open channels					
	5	**6**	**7**	**8**	**9**	**10**
Count	120620	113144	150272	133275	72645	18800
Mean	2.3	1.7	1.8	1.8	1.9	1.9
Std	4.1	1.1	1.3	1.4	1.7	3.2
Min	1	1	1	1	1	1
25%	1	1	1	1	1	1
50%	1	1	1	1	1	1
75%	2	2	2	2	2	2
Max	350	19	32	67	45	263

Table 1: Statistics about the interval lengths per number of open channels

of open channels (whether positive or negative) occur in 0.2, 0.02, 0.002%, respectively.

The public test set is presented Figure 13. The subbatches are coded in different colors. There is also drift, and signal from all 5 groups described above, distributed as follows: 1s, 3, 5, 1s, 1f, 10, 5, 10, 1s, 3, 1s, 1s. In terms of timesteps for each group, the test set is composed of 65% of 1s data, 5% of 1f data, and 10% of data from each of the other groups. This imbalanced distribution will not impact the score however, as the competition is scored with macro F1 score, the unweighted average of the F1 scores of each class.

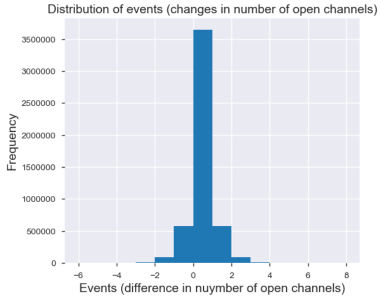

Figure 12: Changes in configuration are called events. They range from -6 (6 fewer open channels compared to the previous timestep) to 8 (8 extra open channels compared to the previous timestep).

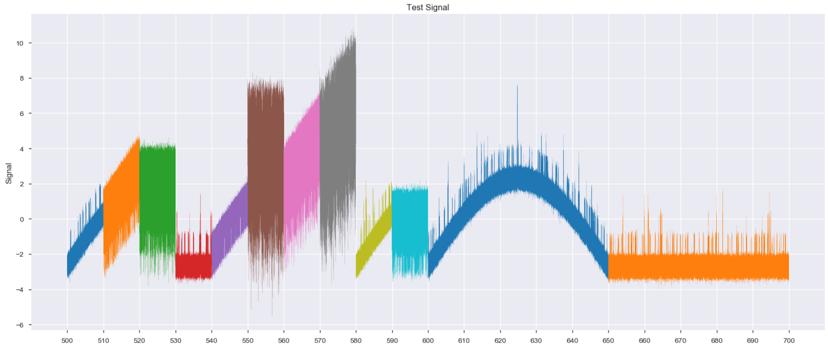

Figure 13: The test set consists of 4 batches, but the amplitude and drift are not uniform within each batch. The signal has been manually divided into subbatches color-coded on the figure.

3 *Signal Preprocessing*

The first step to clean the signal was the removal of the drift. This was done beautifully by Chris Deotte and is explained in details in this discussion. The drift had been added and uses a sine function, sometimes truncated (for example batch 2 of the training signal). The train and test signal with and without drift are presented Figure 15 and 16, respectively.

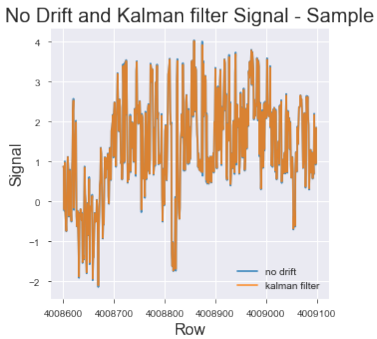

Figure 14: Sample of signal with and without Klaman filter (filter applied by michaln available here). The two signals are hard to distinguish, but the blue trace has a broader amplitude.

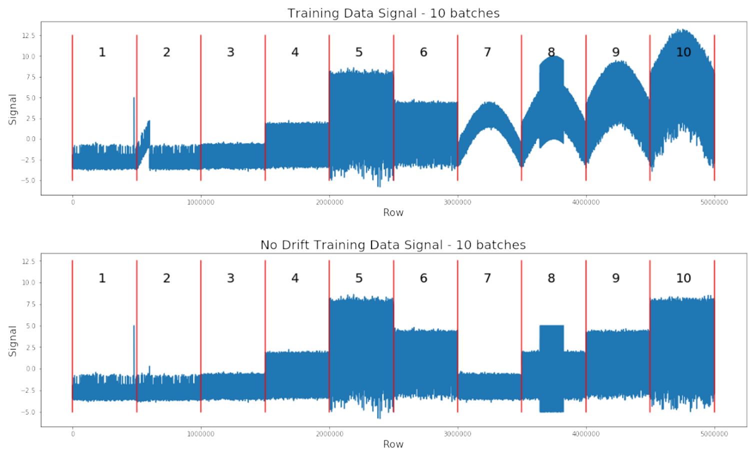

Figure 15: Plots of the train set signal with and without drift.

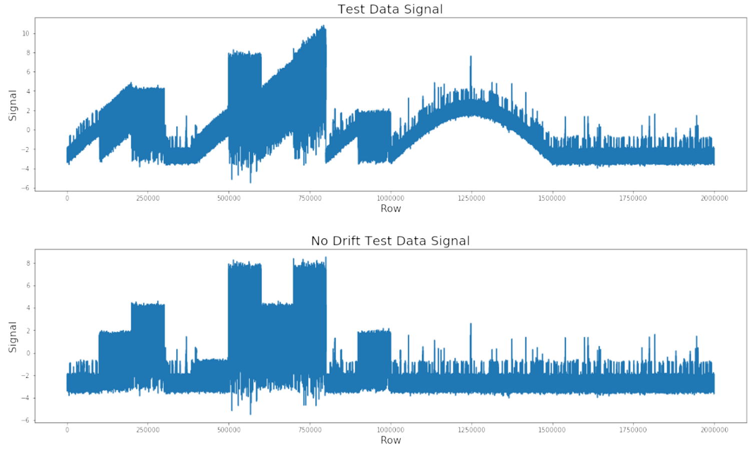

Figure 16: Plots of the test set signal with and without drift.

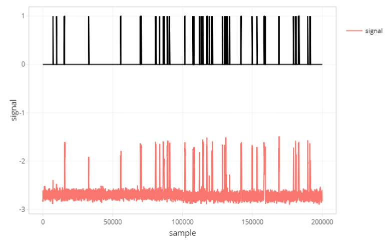

Figure 17: Sample of signal filtered by Klaman filter with parameters H =0.1 and Q = 0.001.

In addition, the signal is noisy. Applying a Kalman filter helped getting a cleaner signal. Kaggle user michaln shared a filtered signal here. Improvement due to the Kalman filter is not noticeable at full scale, see Figure 14 for a sample of the signal.

We tried to improve the signal by applying Kalman filters with different parameters in RStudio (functions SSModel and KFS from package KFAS, parameters: Q and H as arguments of SSModel).

Examples of filtered signals are presented Figure 17 and 18. We tried various combinations of H and Q without investigating the theory. The level of noise is different, but it is hard to tell if the filtered signal will be a better feature than the raw signal, especially for higher numbers of open channels (signal not pictured). We will try multiple options during the modelling part of the project.

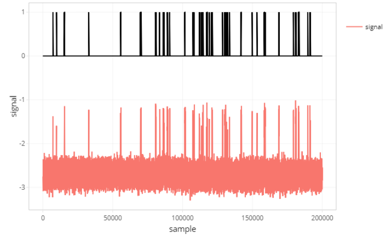

Figure 18: Sample of signal filtered by Klaman filter with parameters H =0.1 and Q = 0.1.

We looked at the Fast Fourier transform (FFT) of the signal to detect patterns and tune filter(s). There is a peak at 50 Hz, which corresponds to the line noise. To remove it, we created a notch filter, which takes the FFT of the signal, zeroes out the Fourier coefficients at/around 50 Hz, then takes the inverse FFT. FFT before and after notch filter are presented Figure 19. The small size of the peak at 50 Hz made us decide agaisnt applying the transformation after: we don't want to lose information around 50 Hz for this small source of noise. This portion of the work was also perform with R.

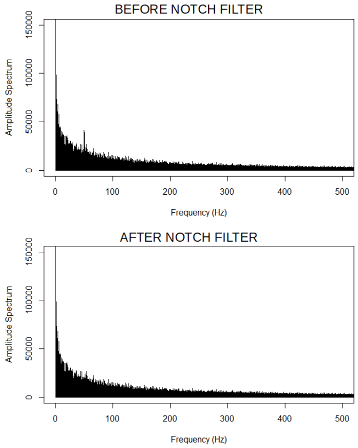

Figure 19: Notch filter applied on the line noise (50 Hz) of the signal: FFT of signal before (top) and after (bottom) transformation.

4 *Conclusion*

The labels of this datare integers between 0 and 10. We can treat it as a regression problem, especially since the events are mostly jumps of one number of open channels, or a classification. Some classes are underrepresented (6, 8-10 open channels), which might pose a problem because the score for the competition is macro F1 score: the average of the F1 score for each class, unweighted. Wrong predictions for the less likely intervals are going to cost as much as other wrong predictions, even though less observations are available for training.

The signal without drift and processed with Kalman filter is the preferred candidate for modelling, but we have other options if the results are not good.