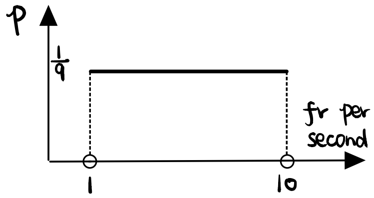
fr_bounds = [1, 10]



⭐ uniform distribution

↓

Set the firing rate of each neuron randomly

$fr' = (1.7209, 6.4609, \cdots, 4.5524)_{1\times 30}$

↓ for neuron j = 1:30

```
dspks = int64(geornd(1/(fs/fr(j)), ceil(2*fr(j)*t_record), 1));

% for neuron j,
% 1/(fs/fr(j)) = fr(j)/fs is the probility to have a spike at
%                          any time sample point.
% ceil(2*fr(j)*t_record) seems to be the num of spikes we are
%                         going to simulate for neuron j, but
%                         later, we will divide it by two.
% r = geornd(p,m,n) generates a multidimensional m-by-n array
%                   containing random numbers from the GEOMETRIC
%                   DISTRIBUTION with probability parameter p.
% geornd(1/(fs/fr(j)), ceil(2*fr(j)*t_record), 1) generates a
%          multidimensional ceil(2*fr(j)*t_record)-by-1 using
%          p=1/(fs/fr(j))

%  Here, GEOMETRIC DISTRIBUTION means to get the FIRST spike,
% how many time samples do we need to count from start sample
% point, given the prob to get a spike at a certain time point
% is p = 1/(fs/fr(j)).

% so, we can have a distribution like this:
%  time sample point T = t1     t2              ......  tn
%                    P = p      [(1-p)^(2-1)]*p         [(1-p)^(n-1)]*p

%  Based on the distribution, we randomly select time points
% in variable T => dspks
%  dspks is taken as the ISI of the adjacent spikes for neuron j.
%   for example, 0-1    1-2   2-3   ...
%                569   1120   954

% MY QUESTION: the assumption is the spikes are independent, the
% ISIs are independent, it seems not like the real situations?
```

⭐ geometric distribution

↓

$dspks' = (23365, 4922, 31, 118, \cdots, 87784)_{1\times 3442}$

↓

remove ISIs below the refractory period

$(2ms = fs \cdot \frac{2}{1000} = 50$ time sample points$)$

$dspks' = (23365, 4922, 118, \cdots, 87784)_{1\times 3430}$

↓

cumulative sum to get the spike times for neuron j

$res = (23365, 23365+4922, 23365+4922+118, \cdots)_{1\times 3430}$

↓

spk_time = $\boxed{\text{spk\_time} \quad\quad\quad}$

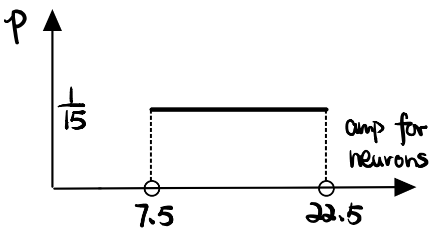
Sort all the spike times for all the neurons in acsend order, then keep spike times within 1000 second (or 1000 × 25000 samples)

$$\boxed{\begin{aligned} \text{spk\_times} &= [89, 131, 370, 490, \cdots, 24999999]_{172224\times1} \\ \text{clu} &= [23, 30, 4, 20, \cdots, 13]_{172224\times1} \end{aligned}}$$

at time sample 89, there is a spike from neuron 23.

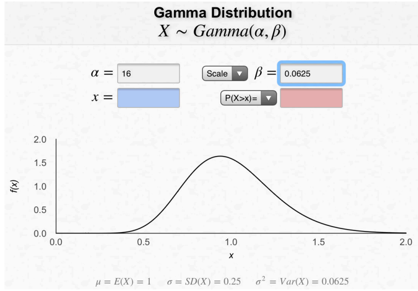
mu_mean = 15

get the mean amps of waveforms for 30 neurons



$$\boxed{\text{mu}' = (14.1355, 20.6638, \cdots, 21.0596)_{1\times30}}$$

```
% Simulate the amps for all the spikes in the spk_times

% this generates single spike amplitude with variability
amps = gamrnd(1/amp_std^2, amp_std^2, nspikes, 1);

% 1. r = gamrnd(a,b,sz1,sz2) generates an array of random
%     numbers from the gamma distribution with the shape
%     parameter a and the scale parameter b, where sz1,sz2
%     indicates the size of each dimension.
%    In this example,  shape parameter a=16, scale parameter
%    b=0.0625
```

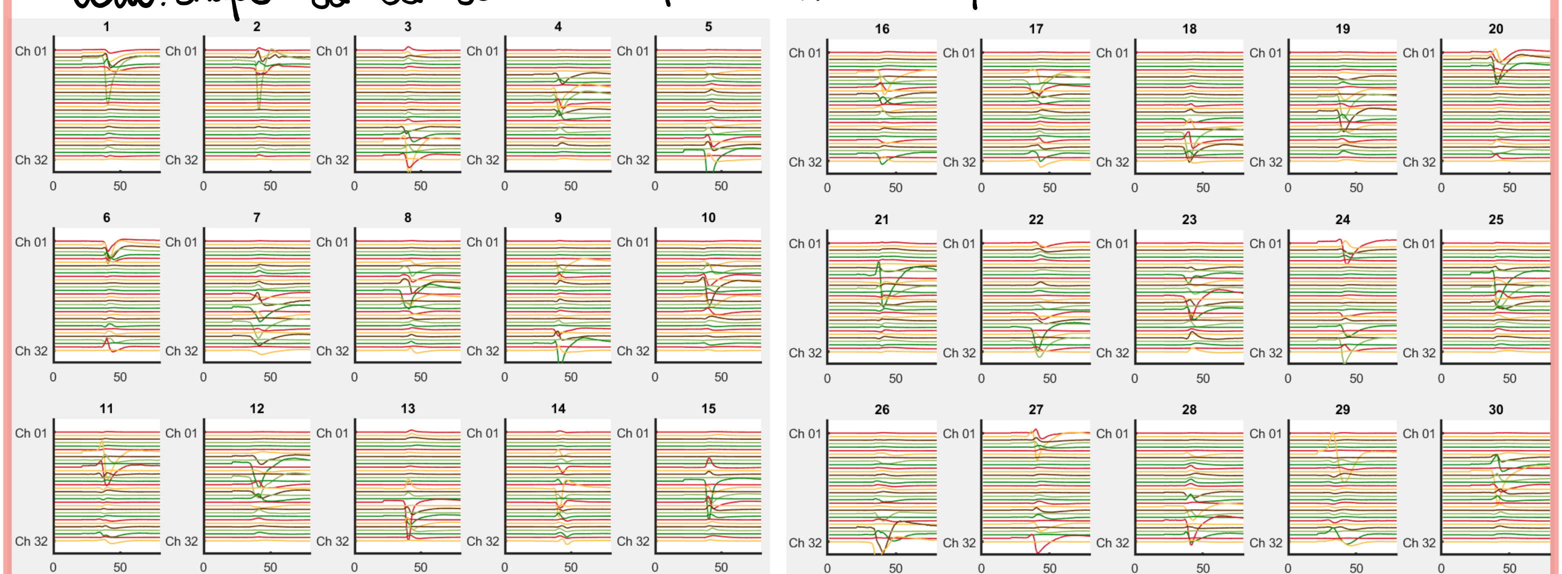



```
% 2. Why used gamma distribution not normal distribution?
%    property 1: non-negative
%    property 2: Skewness: The skewness of the gamma
%                distribution only depends on shape param a,
%                and it is equal to 2/square_root(a)

% 3. guessing: if we want a gamma distribution random
%             variable with a standard deviation=0.25, then
%             a=1/amp_std^2, b=amp_std^2,
%             also, a/b^2=amp_std^2
% 4. the amp for all the spikes is in the range(~0.5, ~1.5)
%    and no negative value will be sampled.
% 5. for example, shape=172224x1 int64
```

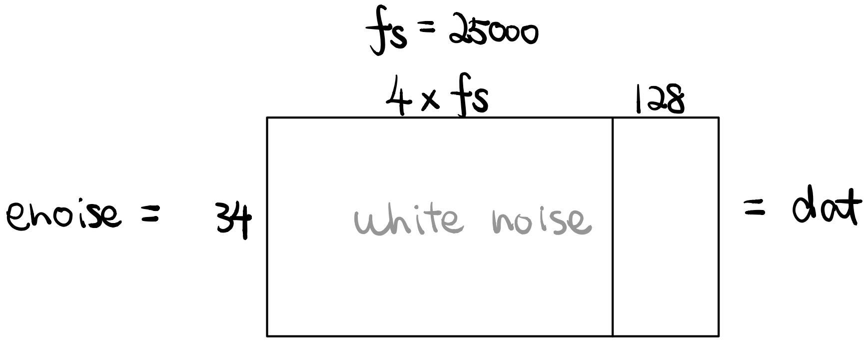
$$\boxed{\text{amp} = [0.5556, 0.7459, 1.1174, 1.0276, \cdots, 1.1444]_{172224\times1}}$$

gamma distribution

wav.shape = 82 × 32 × 30 = time samples × channels × templates



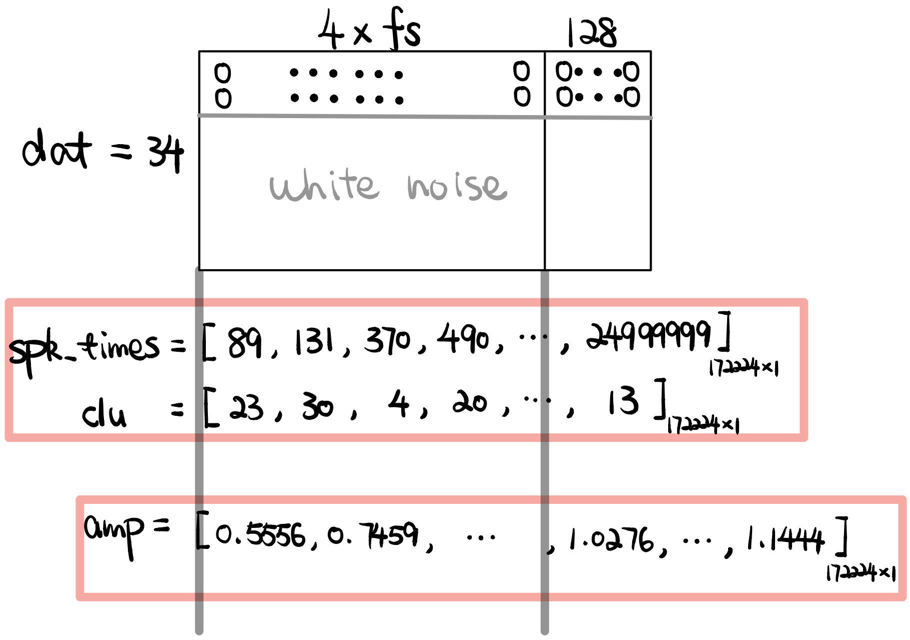
randomly selected based on seed rng(101)



temporal & spatial smooth

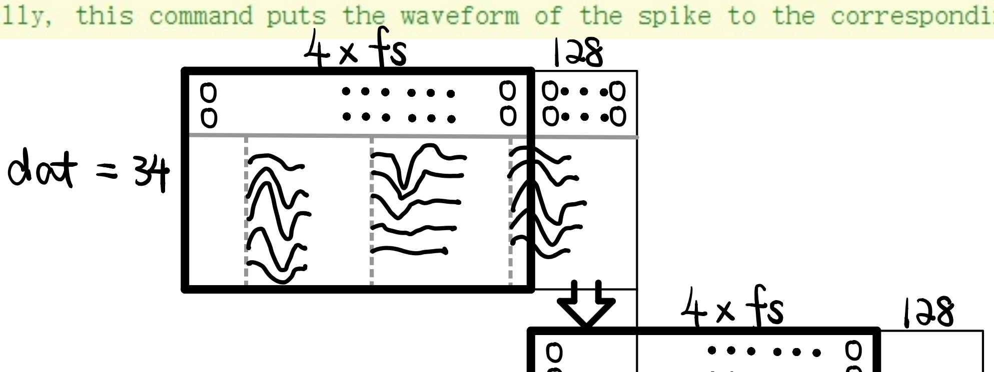
zscore normalization of each channel

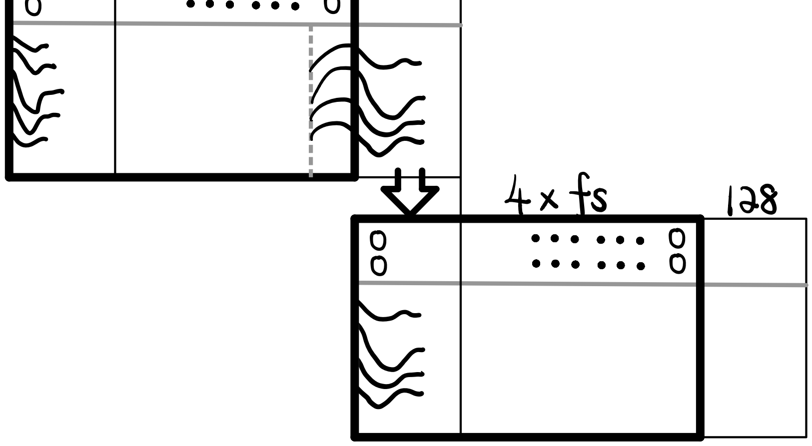
set 1st and 2nd channel (dead) to zero



$$mu = (14.1355, 20.6638, \cdots, 21.0596)_{1\times 30}$$

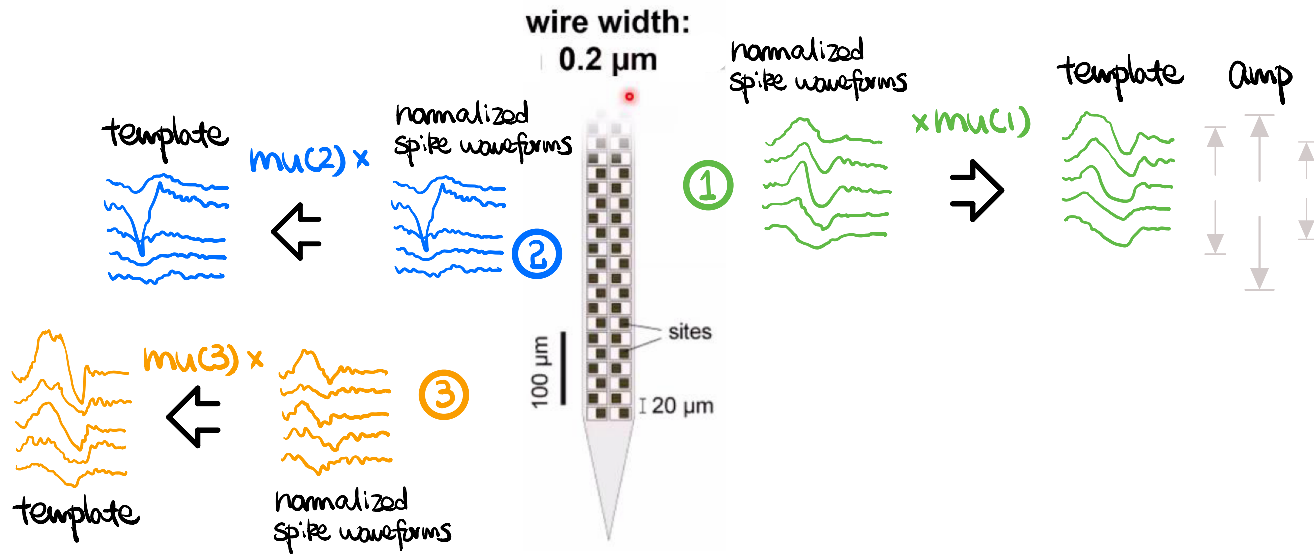
```
dat(time_range, channel_range) = dat(time_range, channel_range) + am(i) * mu(ids(i)) * wav(:,:,ids(i));
% ids(i): is the belonged cluster of the spike ts(i)
% therefore, wav(:,:,ids(i)) is the normalized waveform of the spike
%           mu(ids(i)) is the averaged amplitude for the spike in the cluster ids(i)
%           am(i) is the amplitude variation of this SPECIFIC spike with spike time ts(i)
% Generally, this command puts the waveform of the spike to the corresponding spike time position.
```



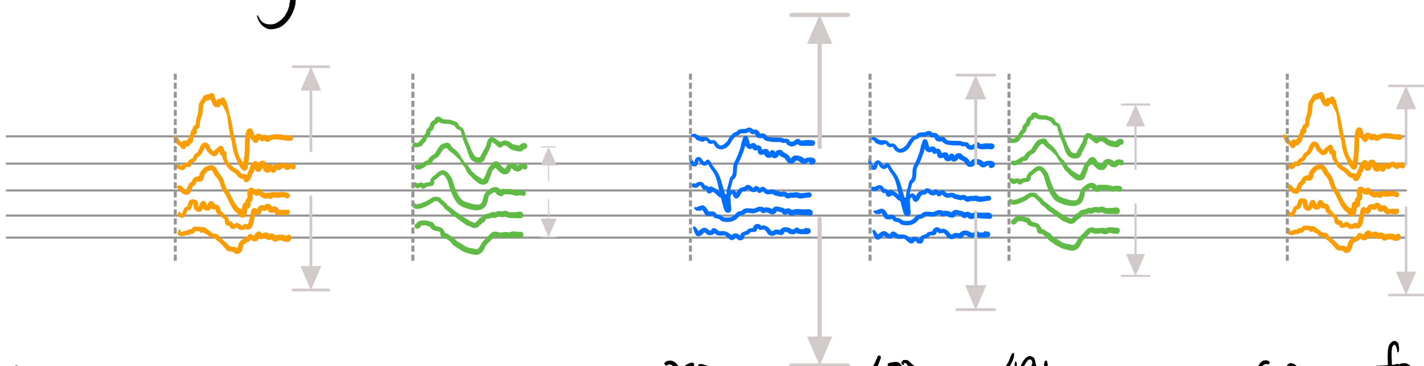



```
if t_all>0
    enoise(1:buff, :) = enoise_old(NT-buff + [1:buff], :);
    %                1 ... 4*fs      4*fs+1  ...    4*fs+buff     4*fs+buff+ 1  ... 4*fs+buff+ 4*fs
    % enoise_old     ||               <-------   same   ------>||                              (previous batch)
    %     enoise                    ||<-------   same   ------>                          ||  (current batch)
end
```

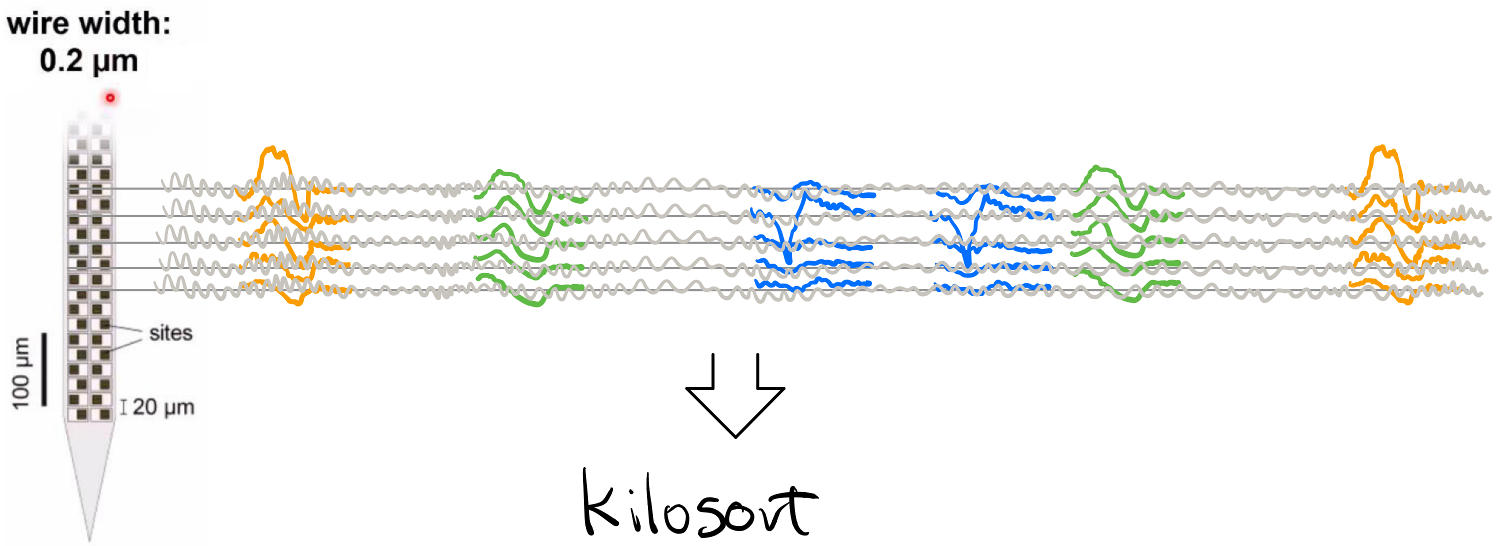
```
if t_all>0
    % dat(1:buff/2, :) = dat_old(NT-buff/2 + [1:buff/2], :);  % dat_old.shape = 100128x34 single

    dat(1:buff, :) = dat_old(NT-buff + [1:buff], :);
    % This is my modification. I believe their code wont't generate
    % proper spike wave near the 'edge' of the segments!
end
```



data modeling:



spike time	131	213	367	433	491	602	fs=25000
neuronID	3	1	2	2	1	3	
amp	1.3	0.5	4.3	2.1	0.9	1.8	



Kilosort

neuron ?

normalized spike waveform ?

mu for each neuron ?

spike time ?

neuron ID ?

amp ?