Magic algorithm-documentation

I Motivation

1) **Introduction**
   Program was developed to automatically identify specific and unique increases of calcium concentrations in root hair cells after Nod factor addition
   . This algorithm is not build on differential equations, but on geometrical properties of a signal and Bayesian spectrum analysis. It is very important for researchers (recognition algorithm), because experimental noise, stochastic effect and underlying deterministic laws in non-trivial and initial data derived from biological processes are not often of sufficient quality to allow them to standard analyses.

2) **Assumptions:**
   Our program looks into geometrical properties of signal and tries to recognize if there appear flux or no. Experimentalists define flux as a smooth increase of calcium lever after NOD factor, but before spike regime (smooth hip on background function). Experimentalists measure calcium level in two places on tip and on cells. They call flux only when it appears simultaneous in both traces.

3) **Methods**
   In our analyze we use two numeric methods to recognize hips (*MinMax* and *Areas)* both modificated by function of local variance (during a flux there are no spikes, so our investigated hip should be smoother than rest of trace). Bayesian conditional probability was involved in adding *a priori* knowledge and Bayesian spectrum analyze was used to eliminate unneeded variables.

II Realization

1) **Program environment and data input**
   Program is written in R as a set of procedures. Input for this program are 3 vectors: time, signal on tip, signal at cells. Out for this program is plot of quasi-probability of appearing flux in time. There is possibility to establish level phase transition between positive and negative answer.

2) **Algorithms:**
   a) ***MinMax* –** We investigate Symmetric Moving Average (MA) of n/2 previous and n/2 ahead data point MA(n) and find its extremums. Distance between two local minimums and multiply by distance between maximum and average of that two minimums is base for calculating probability (earlier has to be normalize by both dimension of whole box which contain trace). We can illustrate it on graph (Fig.1) where arrows represent distances described above. We are calculating that distances for all intervals between two neighbourhood minimums, but we show on graph only on point of flux, because there are significant big then. Let compare trace and schedule of algorithm (Fig.1) with probability calculate by this method in time (Fig.3-right).

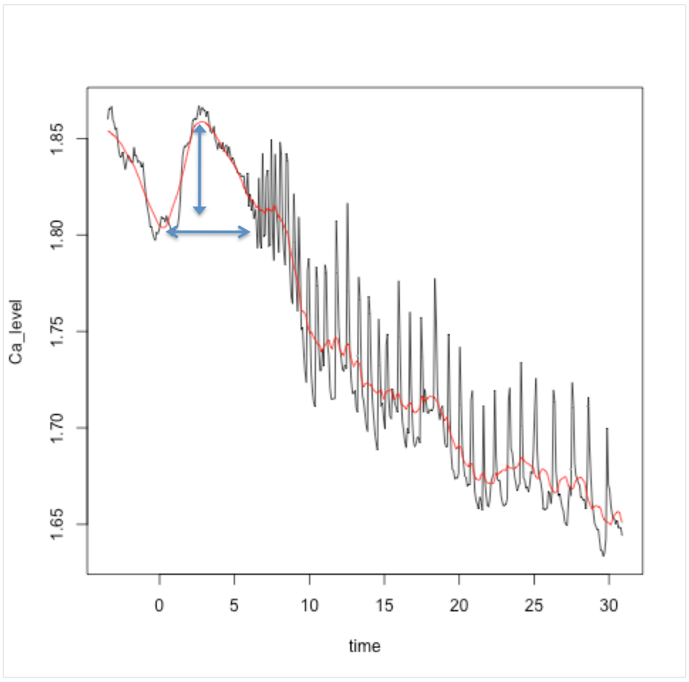

Fig. 1) Calcium trace with clear positive appearance of flux (black-signal, red-MA(22)). Arrows show distances used in *MinMax* algorithm

b) ***Areas***– We are fitting trace with continuous and smooth function. We have found that the best explanation of background (with consideration of complexity) gives polynomial of order 2 PF(2). The goal of this analyze is to find points of intersection between this functional fit and MA(n) and calculate area above fit and below MA(n). Monte Carlo method was used in that calculation. Let treat our areas as a 1-D lines (our data set is discrete, so its natural this 1-D area for all points of time individually). This calculated areas are also normalized by vertical dimension of all trace (distansce between minumal an maximal values) . Exemplification of how *Areas* algorithm catches flux is on Fig. 2.

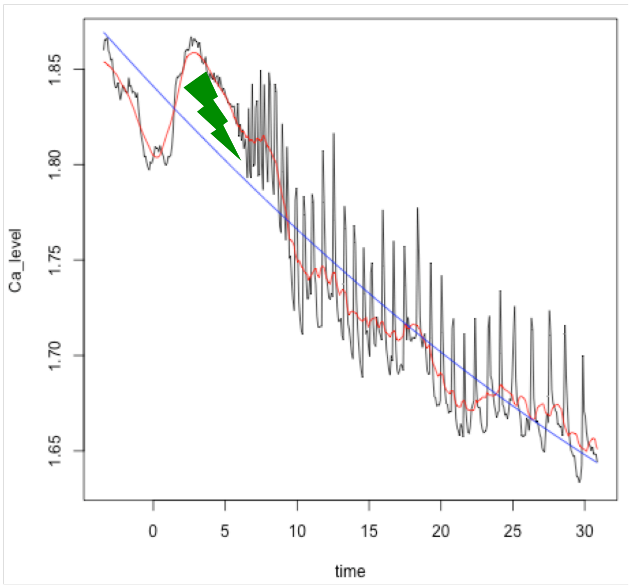

Fig. 2) Calcium trace with clear positive appearance of flux (black-signal, red-MA(22), blue-polynomial fit). Thunderbolt shows big area between MA(2) and polynomial fit:PF(2), which is indicator in *Areas* algorithm (compare with Fig.3-left)

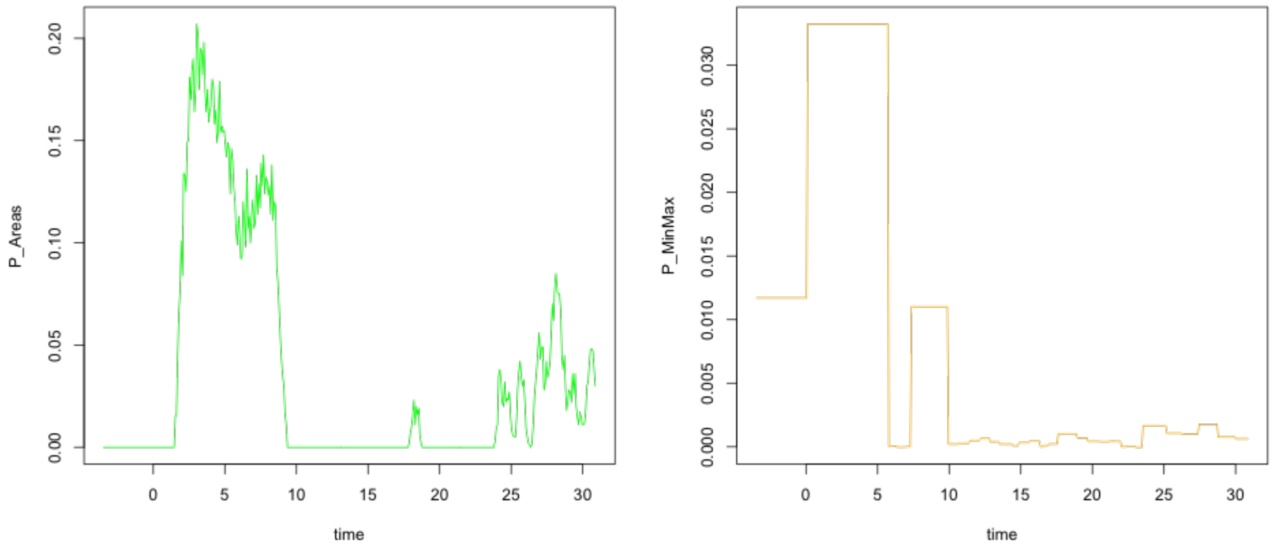

Fig. 3) Probability graphs for *Areas* method (left) and *MinMax* (right). As we can there are few difference between algorithms. First: *MinMax* gives non-zeros values for all time point, but *Areas* only in some time intervals. That happened, because Areas finds only that time intervals, then MA(n) is above square fit, so in other region gives zeros. On the other hand *MinMax* takes into account intervals between two minimums, what describe all time series (with special edges conditions).

c) ***Local Variance-*** Both of algorithms (*MinMax* and *Areas*) do not take into account smoothness of signal. To add this behaviour we are shifting probabilities by function of local variance. Shifting take place separately for both of algorithm. In case of *Areas* we are calculating variance in time intervals between two intersection points, inside there MA(n) is above square fit . In case of *MinMax* we are calculating local variance between two minimums. After that local variance in normalized by global variance and probability is dived by obtained value in every indicated time interval. Variance was calculated after detrending signal with MA(n). We choose shifting function as a midpoint between variance and standard deviation. Multiply factor for *MinMax:*

$S_t^{MM} = Var^{0.75}(T)/Var^{0.75}(T_i)$ **(eq. 1a)**

where $t \in T_i$ (i-th time interval between two minimums detetected by *MinMax*) and whole time series: $T = \cup T_i$.
Equavalent for *Areas*:

$S_t^{A} = Var^{0.75}(T)/Var^{0.75}(T_j)$ **(eq. 1b)**

where $t \in T_j$ (j-th time interval between intersection points where MA(n) is above PF(2) detetected by *Areas*) and $\cup T_j \in T$. If $t \notin \cup T_j$ then S_t^{A} does not exist.
To see how this shifting makes recognition better let look at Fig. 4.

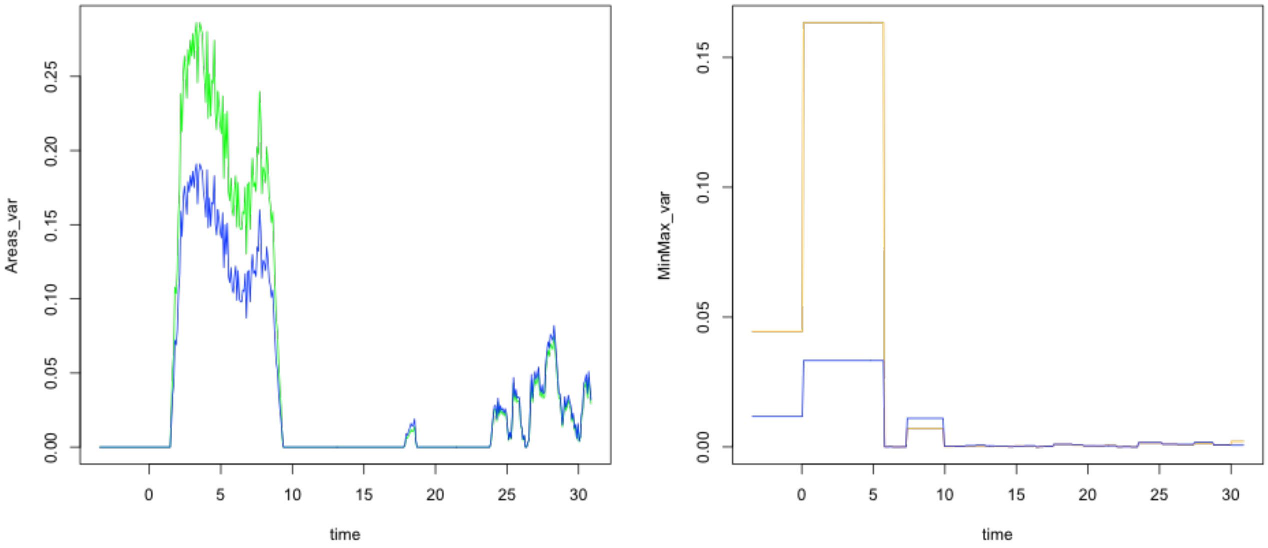

Fig. 4) Qasi-probability graphs for *Areas* method (left) and *MinMax* (right) with (green and orange) and without (blue) shifting by variance. We have to remember, that operation of shifting makes that result cannot be understand as a probability in full sense.

3) Probability analyze

a) Adding and multiplying probability

Let make assumption: shifted values obtained from algorithms *MinMax* and *Areas* can be treated as probabilities. Let call:

A – There is a flux (advised by geometrical analyse)

A_{MM} - There is a flux (advised by shifted *MinMax*)

A_A - There is a flux (advised by shifted *Ares*)

$P(A_{MM})_t$ – Probability of having flux at point *t* (advised by shifted *MinMax*)

$P(A_A)_t$ – Probability of having flux at point *t* (advised by shifted *Areas*)

In terms of logic, A will be true if A_{MM} or A_A will be true ($A_{MM} \vee A_A = A$). We can write the equation of summing probabilities, there the interesting form for us will be probability of having flux at point *t* advised by shifted *MinMax* **or** by shifted *Ares:*

$$P(A_{MM} \vee A_A)_t = P(A_{MM})_t + P(A_A)_t - P(A_{MM} \wedge A_A)_t \qquad \textbf{(eq. 2)}$$

To simplified calculation let assumed, that both algorithms cannot find a flux at the same time point, probability of having flux at point *t* advised by shifted *MinMax* **and** by shifted *Ares* would be small $\{P(A_{MM} \wedge A_A)_t << 1\}$. With this assumption, which is likely to be realistic (usually algorithms finding a flux at different time) we can take $P(A_{MM} \wedge A_A)_t$ as zero and (eq.2) would take a form of:

$$P(A)_t = P(A_{MM})_t + P(A_A)_t \qquad \textbf{(eq. 2')}$$

That was implemented by us.

We have to remember: calcium level is measured in two places (cell and tip) and evidence of flux must be observed both to establish a flux. To distinguish different traces let use superscript notation, e.g. A^C - There is a flux at cell; A^T - There is a flux on tip .To satisfy that condition lets assume that A^C and A^T are independent, so it gives:

$P(A^C \wedge A^T)_t = P(A^T)_t \cdot P(A^C)_t$ **(eq. 3)**

To conclude that is a final equation without considering conditional probability {where $P(A)_t = P(A^C \wedge A^T)_t$}

b) Bayesian analyze of conditional probability

To have better understanding of process, let remained, that flux if appears it start few minutes after Nod factor. That gives us *aprioric* knowledge, which can be used to develop probability. Let remind symbols and add new:

A – There is a flux (advised by geometrical analyse)

B – There is a real flux (we assume that it is 1, because we are not analysing hidden fluxes)

Bayesian equation for conditional probability tells:

$$P(A \setminus B) = \frac{P(B \setminus A) \cdot P(A)}{P(B)} \qquad \textbf{(eq. 4)}$$

Conditional probabilities can be understood as:

P(B) – we assume that its 1, because we are not analysing hidden fluxes

P(A) – probability of detecting a flux (its can be estimated by *apriori* knowledge of time when a flux should appear)

P(A\B) – probability of flux if we have signs that it appears: *aposteriori* probability

Now we can rewrite (eq.4) for most important values *aposteriori* probability in function of time

P(B\A) – Probability of having flux conditioned by geometrical properties (what we calculated in section a))

$$P(A \setminus B)_t = P(B \setminus A)_t \cdot P(A)_t \qquad \textbf{(eq. 4')}$$

Our *apriori* probability $P(A)_t$ can be describe by shifted gamma density function. Parameters of that function are estimated to get maximum of that function in time point, where usually a flux appears (mean time value). Gamma density function had to be shifted, because it exist only for non-negative values (time in our case starts from negative values, because 0 means that Nod factor starts to affect). It had to be normalized (to get 1 on maximum point). After those manipulation *apriori* probability can be used (eq. 4'). Example of *apriori* probability is shown on Fig.5. *Apriori* probability is calculated for all traces (both for cell and tip) and it change individually probability which we are looking for.

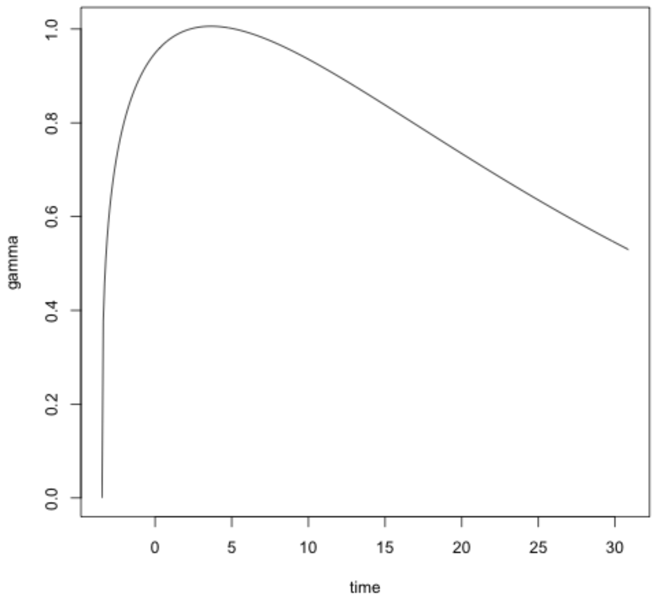

Fig.5) *Apriori* function of probability (taken from shifted gamma density function). Maximum of that probability function is set at one and at time point which should be a point of flux for all cases.

c) Bayesian parameter reduction

Prediction of our algorithms depends on parameters used inside. Suppose n is a "nuisance parameter" what we do not, at least at the moment, need to know. Extend our aposteriori probability with the parameter *n*, which is a parameter of Moving average and apply Bayes' theorem.

$$P(A,n \setminus B) = P(A,n)\frac{P(n \setminus A)}{P(n)} \qquad \textbf{(eq. 5)}$$

and integrate out *n*, obtaining the marginal *aposteriori* probability for A in time

$$P(A \setminus B)_t = \int dnP(A,n \setminus B)_t \qquad \textbf{(eq. 5')}$$

In our discrete case we do not to integrate, but only sum of all *n* with unitary distribution of *n* what as a result give as a mean of all probabilities calculated for different *n.* Experience of calcium spiking group at John Innes Centre gave most suitable *n*=22 and we average probability with values around that number.

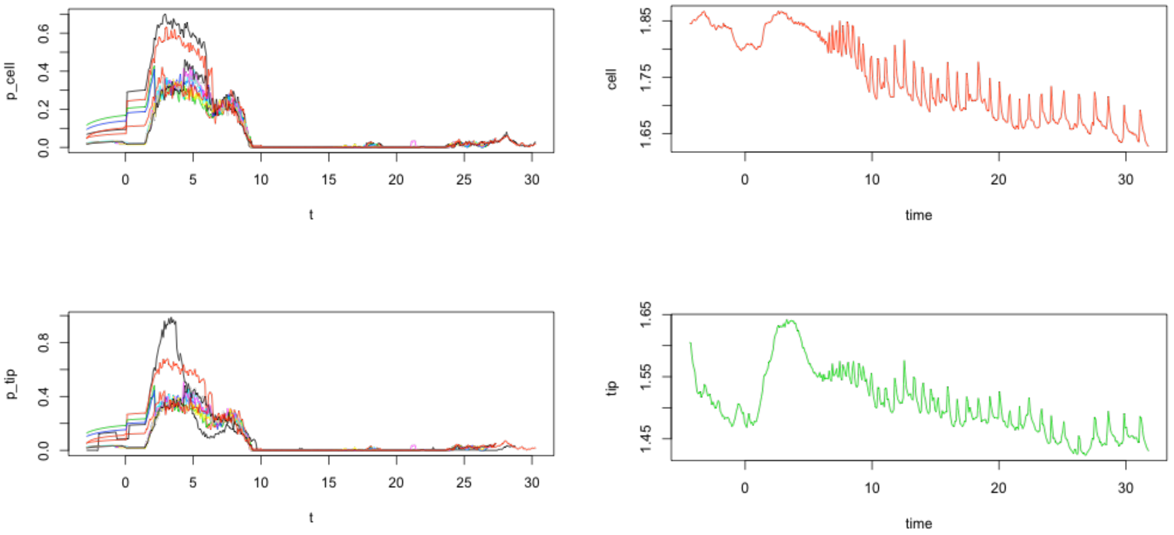

Fig.6) Graphs of probability for different *n* (left) and signal (right). Let focus, that for different *n* are visible quit significant changes of behaviour.

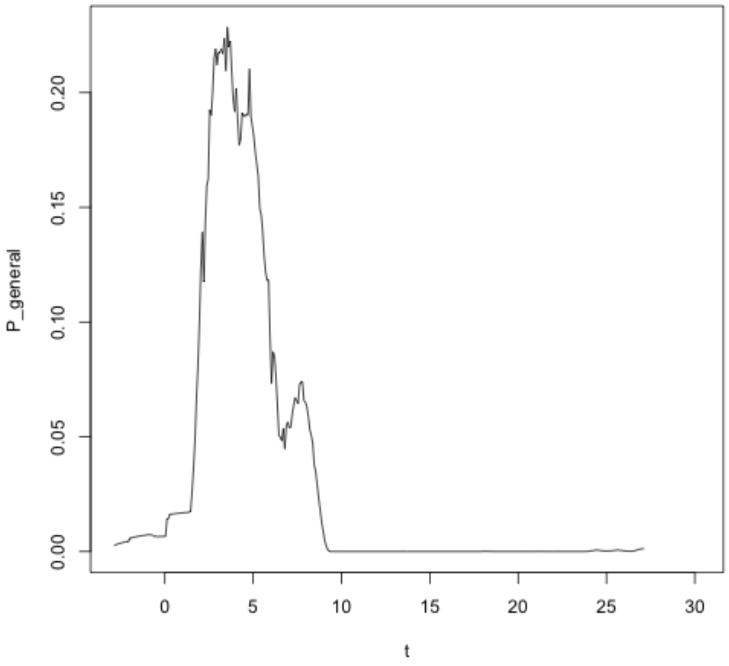

Fig.7) Graphs of probability. This is the final product of our program. Level of probability (that for sure there was no flux) we have set on 0.1. On this picture maximal point above 0.2, so we can say that there is a flux.

III Discussion and interpretation

As we could imagine program give only hints and results has to be interpreted (like X-ray photography in bone injuries). First thing we have to establish threshold which differentiate flux or no. Our idea is following:

Let describe background function with spikes and noise. Then look into anomalies of that model (flux would be one of them). Our program tries to find those anomalies both methods *Areas* and *MinMax* base on how far away signal is from model at certain point.

Experience of analyzing calcium data helps us to establish threshold level. If we treat it as a inverse problem, we can collect all cases, which we know if there are positive or negative by 'eye' we can find level of threshold, which will give the same numbers. This process gives at the end level P=0.1, which was chosen arbitrary and is only a suggestion for experimentalists. Other condition was formulated for further investigation. For some very noisy datasets we had to change rules (P=0.1) a little. And now:

- if there is a second peak at least half size of first and above 0.075 (3/4 of threshold) so now threshold is 0.2
- if there are a third or further peaks we decided to say, that data is to noisy to determine if it is positive or negative
- if peak is narrower of 3-4 minutes we do not accept it

With those rules we can avoid some nasty evidences, which sometimes happened in dataset. Rule of wideness of peak could be avoid by adding more MA parameters to our integration over that, but it would change time consumption very much, so we decided to stay with that rule. We have already developed algorithm very much, because at the beginning those narrow peaks appear quit often (if there is in data small time interval with small variance and it would be catch by the chance by Areas or MinMax, then quit big peak could appear, which have nothing to do with flux). Those additional assumption are involve only in few percent of traces and apply only to very noisy and problematical to analyze, but there are very important supplement in terms of statistical analyze of huge numbers of data (we are avoiding bias of method this way)