Scoring the Parameter Estimation Challenge

This challenge will be scored based only on the following two criteria: 1) The "distance" between simulated and predicted values, and 2) The distance between estimated and known parameters.

1. Distance between simulated and predicted values

The challenge requests for model 1 predictions of three proteins at times $t_i = 0.5\ i$, for $i = 1, 2, \ldots, 40$. Let's denote by $p_k^{pred}(t_i)$ and $p_k^{sim}(t_i)$ the predicted and simulated levels of protein *k* at times t_i. The proteins required to be predicted are the following:

Model	Time-courses to predict
1	p3, p5 and p8

Because the initial conditions are given, the real challenging predictions take place after some time has elapsed from *t*=0. We will take that time to be 5. Therefore the squared distance between predicted and measured protein abundances for model *j* can be taken to be:

$$D_j^{prot} = \frac{1}{90} \sum_{i=11}^{40} \frac{\left[p_k^{pred}(t_i) - p_k^{sim}(t_i)\right]^2}{0.01 + 0.04\left[p_k^{sim}(t_i)\right]^2}$$

Note that the squared difference terms are normalized with the variance, and the variance follows the noise model that was implemented in the data provided. The noise model is simply $\sigma_{baseline}^2 + \sigma_{signal}^2$, were $\sigma_{baseline} = 0.1$ and $\sigma_{signal} = 0.2\ p_k^{sim}$. $\sigma_{baseline}$ represents a baseline, signal independent, measurement noise, and σ_{signal} represents a signal dependent measurement noise with a constant coefficient of variation of 0.1.

Also note that we are dividing by 120 (3*40) the number of terms being added. So a noisy but excellent prediction will give a value close to 1, as we should take into account the fact that the models are based on fits to noisy data. We could think could have used protein concentrations in logarithmic scale. However, it is often the case that the protein abundances go to vales close to 0, and the logarithm will give a divergence.

A null model will be created from this distance, based on the predictions of all the participants. For example, if there are M participants, we will choose at random one of the M predictions for $p_k^{pred}(t_{11})$, then at random one of the M predictions for $p_k^{pred}(t_{12})$, …, and finally one of the M predictions for $p_k^{pred}(t_{40})$. We will therefore get a value of D_j^{prot} which would correspond to one possible random choice of predictions amongst all the participants. If we do the same process a large number of times, we will generate a distribution of distance squares, for which a p-value can be estimated for D_j^{prot}. That p-value will be denoted as p_j^{prot}.

2. Distance between estimated and known parameters

The challenge requests predictions of all the parameter for the two models. As protein degradation rates are equal, they are in practice only 1 parameter for each of the 2 models. Hence, Models 1 and 2 have N_p = 45 and 61 parameters considered for scoring. Let's denote as v_i^{pred} and v_i^{real} the predicted and actual parameter values used in the simulations where *i* runs between 1 and N_p. The mismatch between estimated and true parameters will be assessed on the log-scale. In this way, a mismatch by a factor of x has the same penalty independently on the nominal value of a parameter. Furthermore, the ratio becomes independent on changes of physical units. Therefore the "distance" between predicted and real parameters can be taken to be:

$$D_j^{param} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left[\log\left(\frac{v_i^{pred}}{v_i^{real}}\right)\right]^2$$

Similarly to the case of the distance between simulated and predicted protein abundance, a null model will be created from the distance between estimated and known parameters, based on the predictions of all the participants. If there are M participants, we will choose at random one of the M predictions for v_1^{pred}, then at random one of the M predictions for v_2^{pred}, …, and finally one of the M predictions for $v_{N_p}^{pred}$. We will therefore get a value of D_j^{param} which would correspond to one possible random choice of predictions amongst all the participants. If we do the same process a large number of times, we will generate a distribution of distance between known and estimated parameters, for which a p-value can be estimated for D_j^{param}. That p-value will be denoted as p_j^{param}.

3. Distance between estimated and true network

The challenge requests predictions for 3 missing links, knowing that a gene can regulate up to two genes when they are in the same operon, 6 gene interactions have to be indicated by the participants (3 links*2 genes) and whether these interactions are activating (+) or repressing (-).

We define the score: $S_2^{netw} = \sum_{i,j=1}^{3} L_j + \delta_{ij} \qquad 0 \leq S_2^{netw} \leq 1$

where L_j is 6 if both genes defining the link and the nature of the regulation are correct (i.e +/-). In case a link is **NOT** correct, δ_{ij} adds 1 for each correct regulated gene, 2 if the regulated gene and the nature of the regulation (i.e +/-) are correct and 1 if the regulator gene is correct.

For each participant, the p-value associated with the score S_2^{netw}, will be calculated by generating a distribution of scores from a large number of surrogate gene networks obtained by randomly adding 3 links that follow the connection rules to the initial gene network.

4. There will be no scoring based on used credits

There is no standard or obvious way of scoring in which it is optimal to spend some but not all the credits. We will ignore for this challenge the amount of credits spent. Therefore participants are encouraged to spend the whole budget.

5. Overall score over the two gene networks

Model 1 and Model 2 will be scored separately.

Model 1will have the p-value for the time trace predictions and a p-value for the parameter predictions. We will take the overall score to be a function of the product of all the p-values.

Model 2 will have the p-value for the network prediction. More specifically:

$$Score\ Model\ 1 = -\log\left(p_1^{prot}.p_j^{param}\right)$$

$$Score\ Model\ 2 = -\log\left(p_2^{netw}\right)$$