History

- Initial model by mikeg on 6/10/15.
- NSE model added by mikeg on 7/21/15.
- 08/16/28: Fixed error for Pr(Elongation at position j) in Equation (14) by adding $\lambda_j v_{g,i}$ term. Additionally, applied fix to Equation (17) by adding missing term to product term. Added notes about expecting $\alpha_c > 0$.
- 08/20/18
  - Replace Pr(Elongation at position j) with Pr(Ribosome sampled is at position j)
  - Added math analysis is partially documented in `rfp.math.nb` which was added to the repository.
  - Added second order approximation of σ_i
- 10/22/2019
  - Alex: Added continued fraction representation for approximating upper incomplete gamma function
- 06/30/2020
  - Added equation 25 as a simpler approximation for $v_{g,i}/(v_{g,i} + w_{g,i})$ for equation **??**.
  - Added note about how the integral function needs to use λ_c, we can't use λ'_c.
- Compiled on Tuesday 30$^{\text{th}}$ June, 2020 at 12:27

Goal

Create sensible model for interpreting RPF data with a special interest in working with data used in Pop et al. (2014).

Pausing Time Model Definition

Calculating the Likelihood of a sample

We are interested in calculating the probability of observing a ribosome footprint (RFP) experimentally. We assume there is a pool of RFP generated from the transcriptome, that the mRNAs in this pool are at close

to steady state in terms of ribosome initiation and completion of translating a transcript.

Beginning by considering a single mRNA molecule transcribed from gene g, the probability a ribosome is at position i of this mRNA molecule, $p_{g,i}$, is simply,

$$p_{g,i} \propto \kappa_g w_{g,i} \tag{1}$$

where κ_g is a rate constant scales the average rate at which ribsomes intercept and initiate translation of an mRNA molecule from gene g, mRNA$_g$, under the experimental conditions used. Formally, the initiation rate is determined by $\kappa_g \times r$, where r is the density of ribosomes in the cell. However, we assume all mRNAs are equally accessible to ribosomes so, as a result, r will cancel out in the following equation and, as a result, we ignore it throughout. Additionally, $w_{g,i}$ is the average waiting, pausing, or dwell time of a ribosome at position i of mRNA from gene g. Derivation of equation 1 is straight forward, but can also be found Gilchrist and Wagner (2006) equation (20) with the nonsense error rate = 0 and the ribosome recycling probability < 1. We can link it to Pop et al. (2014) work by noting the ribsome flux J_g on an individual mRNA$_g$ is $J_g = r\kappa_g$.

Biologically speaking, $w_{g,i}$ values for the same codon are not independent. The values of $w_{g,i}$ for the relevant codon likely vary within a gene as a function of mRNA structure and other factors. To capture this variation, we will assume that for when the codon at position i is of type c, $w_{g,i} \sim \text{Gamma}(\alpha_c, \lambda_c)$, where α_c is the shape parameter, λ_c is the rate parameter, and $E[w_{g,i}] = \alpha_c/\lambda_c$. Gene specific effects could also be incorporated since mRNA structures which interfere with efficient translation likely declines with expression level. As a result of this assumption, we can use a Negative-Binomial (NB) distribution model to analyze the RFP data. Regarding values of α_c, no variation in waiting times between instances of the same codon would correspond to an exponential model or $\alpha_c = 1$. Given that we expect heterogeneity between sites, we expect $\alpha_c > 1$.

Assuming independence in sampling, the total probability of a randomly selected footprint is from position i, $P_{g,i}$, is

$$P_{g,i} = p_{g,i} m_g / Z \tag{2}$$

where m_g is the density of mRNA$_g$ in the cell and Z is a partition function that ensures our sampling probabilities across the transcriptome sums to 1 and is defined as,

$$Z = \sum_g m_g \left(\sum_i p_{g,i} \right). \tag{3}$$

Equations (2) and (3) indicate that our choices of time and volume units for κ_g and m_g are irrelevant and we can only estimate their values relative to one another. This is because our choice of time units for κ_g and density for m_g will rescale both $p_{g,i}$ and, thus $Y_{g,i}$ and Z. One solution is to use the same approach in our other work and scale time such that the average protein production rate across genes $E_g\left(\kappa_g m_g \sigma_g(n_g)\right) = 1$ where $\sigma_g(n_g)$ is the probability a ribosome completes translation of the n_g codons in gene g. For our current model where we ignore nonsense errors, $\sigma_g(n_g) = 1$ and, thus, $\phi_g = \kappa_g m_g$. An alternative would be to constrain $E_g\left(\kappa_g m_g\right) = 1$, i.e. independent of $\sigma_g(n_g)$. [1]

To simplfy calculations we can derive an expected value for Z as,

$$E(Z) = E(\sum_g m_g \sum_i p_{g,i}) = \sum_g \kappa_g m_g \sum_c n_{c,g}(\frac{\alpha_c}{\lambda_c}) \tag{4}$$

Letting $Y = \sum_{g,i} Y_{g,i}$ be the footprint sample size, where $Y \gg 1$ and $P_{g,i} \ll 1$, then we can approximate the probability of observing $Y_{g,i}$ samples of a footprint in mRNA_g at position i using a Poisson distribution with a sampling rate of $YP_{g,i}$. Note that deep sequencing may violate the sampling with replacement assumption of the Poisson distribution. Given our assumption about the distribution of $p_{g,i}$, $Y_{g,i} \sim \text{NB}(x = \alpha_c, p = \kappa_g m_g/(\lambda'_c + \kappa_g m_g)$ where $\lambda'_c = \lambda_c Z/Y$. Explicitly,

$$\begin{aligned}
\Pr\left(Y_{g,i}|\alpha_c, \lambda'_c, m_g, \kappa_g\right) &= \frac{\Gamma\left(\alpha_c + Y_{g,i}\right)}{\Gamma\left(\alpha_c\right) Y_{g,i}!} \left(\frac{m_g\kappa_g}{\lambda'_c + m_g\kappa_g}\right)^{Y_{g,i}} \left(\frac{\lambda'_c}{\lambda'_c + m_g\kappa_g}\right)^{\alpha_c} \\
&= \frac{\Gamma\left(\alpha_c + Y_{g,i}\right)}{\Gamma\left(\alpha_c\right) Y_{g,i}!} \left(\frac{m_g\kappa_g}{\lambda'_c + m_g\kappa_g}\right)^{Y_{g,i}} \left(1 - \frac{m_g\kappa_g}{\lambda'_c + m_g\kappa_g}\right)^{\alpha_c}
\end{aligned} \tag{5}$$

Note that m_g and κ_g are gene g and environment specific terms which can be equated to the equilibrium protein synthesis initiation rate ι_g for gene g under the experimental conditions, i.e. $\iota_g = \kappa_g m_g$. Further, because in this model nonsense errors are ignored, all ribosomes that initiate translation also complete translation. Thus, the initiation rate ι_g is also equal to the synthesis rate ϕ_g. The composite parameter λ'_c consists of the codon specific scale term λ_c and the ratio of Z to Y, two genome wide parameters. The term Y/Z can be interpreted as the sampling efficiency, i.e. what proportion of the RFP state space is sampled. If $Y/Z \ll 1$, then sampling is sparse.

Given the properties of the NB (Forbes et al., 2011, p. 141) and the fact that most codons appear within a gene's ORF multiple times, the likelihood of the parameters given the total number of RFP observed

[1] Prior to 06 Mar 2019 we we used ψ instead of ι (see below). However, this use of ψ was not wholly consistent with how ψ was defined in SelAC where ψ is the target protein functionality production rate so we've changed our notation.

derived from codons of type c in gene g, $Y_g^c = \sum_{i=\in c} Y_{g,i}$, is,

$$\mathrm{L}\left(\iota_g, \alpha_c, \lambda_c' | Y_g^c, n_g^c\right) = \frac{\Gamma\left(n_g^c \alpha_c + Y_g^c\right)}{\Gamma\left(n_g^c \alpha_c\right)} \left(\frac{\iota_g}{\lambda_c' + \iota_g}\right)^{Y_g^c} \left(1 - \frac{\iota_g}{\lambda_c' + \iota_g}\right)^{n_g^c \alpha_c} \tag{6}$$

$$= \frac{\Gamma\left(\alpha_{c,g}' + Y_g^c\right)}{\Gamma\left(\alpha_{c,g}'\right)} \left(\frac{\iota_g}{\lambda_c' + \iota_g}\right)^{Y_g^c} \left(1 - \frac{\iota_g}{\lambda_c' + \iota_g}\right)^{\alpha_{c,g}'} \tag{7}$$

where n_g^c is the number of times codon c is found in the ORF of gene g and $\alpha_{c,g}' = n_g^c \times \alpha_c$. Note we dropped the $Y_g^c!$ term because that will be the same for all parameter values. The total Likelihood of the data is

$$\mathrm{L}\left(\vec{\iota_g}, \vec{\alpha}_c, \vec{\lambda}_c' \middle| \vec{Y_g^c}, \vec{n_g^c}\right) = \prod_{g \in \mathbb{G}} \prod_{c \in \mathbb{C}} \mathrm{L}\left(\iota_g, \alpha_c, \lambda_c' | Y_g^c, n_g^c\right) \tag{8}$$

Note that using RFP data alone, $\kappa_g \times m_g = \iota_g$ and $\lambda_c' = \lambda_c Z/Y$ are only identifiable as joint parameters. (Although it seems like you should be able to calculate Z post-hoc from the state of the chain and estimates of m_g are available from other sources.) Most standard libraries require that the x parameter in a NB be discrete, which in this case it is not. Thus to simulate Y_g values based on equations (5) or (7), first pull ${X^c}_g$ from Gamma(n_g^c α_c, λ_c')[2], then pull Y from Poisson(${X^c}_g \iota_g$).

Pop et al. (2014) provide RNA-Seq based counts of mRNA abundances, M_g, in addition to RFP counts. If we assume that $M_g \sim$ Poisson($Y m_g$), we can easily combine both the RFP and the mRNA counts together and estimate κ_g and m_g separately. Alternatively, we could estimate the composite κ_g m_g parameter using the RPF data and then analyze the those results using the M_g data. An additional fact worth noting is that we are treating Z as a constant for much of our calculations when, in fact, it is a random variate. We should discuss this with Russ to ensure there are no issues.

Pausing Time with Nonsense Error Model Definition

The flux equation, Equation (1), no longer holds when nonsense errors (NSE) are possible. Instead, following Gilchrist and Wagner (2006), we have the conditional probability,

$$p_{g,i} | w_{g,i} \propto \kappa_g \sigma(i-1) \frac{w_{g,i} v_{g,i}}{w_{g,i} + v_{g,i}} \tag{9}$$

Where, as before, κ_g is the translation initiation rate constant per mRNA$_g$, $w_{g,i}$ is the waiting time to elongate codon i, and $w_{g,i}$ is 1 over the codon elongation rate ($1/c_i$ in rate based, rather than waiting time,

[2] recall that λ_c' is the 'rate' parameter.

terminology) and $v_{g,i}$ is the NSE 'wait' time, i.e. 1 over the NSE rate ($1/b_i$ in rate based terminology), and $\sigma(i-1)$ is the probability a ribosome that initiates translation will reach the ith codon. Note that because the waiting time to a NSE, $v_{g,i}$, is so much greater than the elongation waiting time $w_{g,i}$ we can ignore the actual variation in $v_{g,i}$ between codons of the same type (this is easier to understand if you consider $0 < b_i \ll 1$). Thus while we allow $v_{g,i}$ to be codon specific, we treat each of these values as fixed and use v_i instead of $v_{g,i}$. Further, again because $v_i \gg w_{g,i}$, we can approximate $\frac{w_{g,i}v_i}{w_{g,i}+v_i}$ as $w_{g,i}$ based on a Taylor series expansion around $1/v_i = 0$. Thus,

$$p_{g,i}|w_{g,i} \propto \kappa_g \sigma(i-1) w_{g,i} \tag{10}$$

which is equivalent to the simple pausing time calculation except $p_{g,i}$ is reduced by $\sigma(i-1)$.

The function $\sigma(i-1)$ depends on the probability of successful elongation at the $i-1$ upstream codons. When the waiting time for elongation each position j is known, then

$$\Pr(\text{Ribosome Sampled is at position } j) = \frac{v_j}{w_{g,j} + v_j} \tag{11}$$

and

$$\sigma\left(i-1\right) = \prod_{j=1}^{i-1} \frac{v_j}{w_{g,j} + v_j} \tag{12}$$

However, when fitting the model we don't wish to actually estimate individual $w_{g,j}$ values, but instead the parameters of the Gamma (α_j, λ_j) distribution the $w_{g,j}$ are drawn from. Thus we treat $w_{g,j}$ as a random variable whose uncertainty we integrate across to get an expected value. Letting $f(\alpha, \lambda)$ represent the PDF of the Gamma distribution, the expectation of the codon specific elongation probability is

$$E\left[\Pr(\text{Ribosome Sampled is at position } j)\right] = \int_0^\infty \frac{v_j}{w_{g,j} + v_j} f\left(w_{g,j}|\alpha_j, \lambda_j\right) dw_{g,j} \tag{13}$$

$$= \lambda_j v_j \exp\left[\lambda_j v_j\right] E_{p=\alpha_j}\left(\lambda_j v_j\right) \tag{14}$$

where $E_p(z)$ is the generalized exponential integral function (also referred to as Schlomilch functions (Oldham et al., 2009, p.380) and is represented as,

$$E_p(z) = \int_1^\infty \frac{e^{-zt}}{t^p} = z^{p-1} \int_z^\infty \frac{e^{-t}}{t^p} dt = z^{p-1}\Gamma(1-p, z) \tag{15}$$

where $\Gamma(1-p,z)$ is the upper incomplete gamma function. See http://dlmf.nist.gov/8.19 for more details.

Note that, although the complete Gamma function is discontinous at negative interger values, the upper incomplete is well defined even at negative integers so long as $z > 0$.

Using this formulation,

$$E\left[\Pr(\text{Ribosome Sampled is at position } j)\right] = \exp\left[\lambda_j v_j\right] \left(\lambda_j v_j\right)^{\alpha_j} \Gamma(1-\alpha_j, \lambda_j v_j) \tag{16}$$

Assuming independence in $w_{g,i}$ between positions means that the expected value of $\sigma_g(i)$ is,

$$E\left[\sigma(i)\right] = \exp\left[\sum_{j=1}^{i} \lambda_j v_j\right] \prod_{j=1}^{i} \left(\lambda_j v_j E_{\alpha_j}\left(\lambda_j v_j\right)\right) \tag{17}$$

Note that these λ_j terms are equivalent to λ_c (but not λ_c', which is used below). Because the calculations use nonlinear functions, we can't really use λ_c' as a substitute and, insteads, need to estimate Z.

To approximate $E\left[\sigma_g(i)\right]$, we set $v_i = 1/(c_i b)$ where c_i is a codon specific scaling factor which we assume $c_i > 0$ and define $\sum_i^n c_i = 1$ such that $b = f(\vec{v}) = 1/n \sum_i^n 1/v_i$ is the mean nonsense error rate. We could, alternatively, define $\prod_i^n c_j = 1$ which would make $b = f(\vec{v}) = \prod_i^n (1/v_i)^{1/n}$ the geometric mean of $1/vi$. Either way, taking a first order Taylor Series approximation around $b = f(\vec{v}) = 0$.

$$E\left[\sigma(i)\right] = 1 - \sum_{j=1}^{i} \frac{\alpha_j}{\lambda_j v_j} + O(b^2) \tag{18}$$

The second order approximation around $b = 0$ is,

$$E\left[\sigma(i)\right] = 1 + \sum_{j=1}^{i} \left[\frac{\alpha_j}{\lambda_j v_j}\left(-1 + \sum_{k=1}^{i} \frac{\alpha_k}{\lambda_k v_k}\right) + \frac{\alpha_j}{{\lambda_j}^2 v_j^2}\right] + O(b^3) \tag{19}$$

which can be written in terms of the moments of w_i

$$= 1 + \sum_{j=1}^{i} \left[\frac{E\left[w_j\right]}{v_j}\left(-1 + \sum_{k=1}^{i} \frac{E\left[w_k\right]}{v_k}\right) + \frac{\text{Var}\left(w_j\right)}{v_j^2}\right] + O(b^3) \tag{20}$$

Note also that there should be a way to represent and, hopefully, efficiently calculatethe above equation using matrix notation.

An alternative is to approximate $\ln(\sigma_i)$ around $b = 0$ and then exponentiate this approximation. The

first order approximation is,

$$\ln(E\left[\sigma(i)\right]) = -\sum_{j=1}^{i} \frac{\alpha_j}{\lambda_j v_j} + O(b^2) \tag{21}$$

The second order approximation is,

$$\ln(E\left[\sigma(i)\right]) = \sum_{j=1}^{i} \left[-\frac{\alpha_j}{\lambda_j v_j} + \frac{\alpha_j}{{\lambda_j}^2 v_j^2} + \frac{1}{2}\left(\frac{\alpha_j}{\lambda_j v_j}\right)^2 \right] + O(b^3) \tag{22}$$

which can be written in terms of the moments of w_i

$$\ln(E\left[\sigma(i)\right]) = \sum_{j=1}^{i} \left[-\frac{E\left(w_j\right)}{v_j} + \frac{\mathrm{Var}\left(w_j\right)}{v_j^2} + \frac{1}{2}\left(\frac{E\left(w_j\right)}{v_j}\right)^2 \right] + O(b^3) \tag{23}$$

which seems simpler to compute, but I am unsure if and when it would perform better.

Another alternative implementation is to use a continued fractions approximation for the upper incomplete gamma. This is currently implemented in PANSEModel.cpp as PANSEModel::UpperIncompleteGammaHelper. Following notation from functions.wolfram.com,

$$\Gamma(a,z) = \cfrac{z^a e^{-z}}{z + \cfrac{1-a}{1 + \cfrac{1}{z + \cfrac{2-a}{1 + \cfrac{2}{z + \cfrac{3-a}{1 + \cfrac{3}{z+\cdots}}}}}}} \tag{24}$$

One could ignore the fact that w is a RV and, instead use it's expected value, i.e. replace $w_{g,j}$ with α/λ and equation **??** becomes

$$\sigma\left(i-1\right) \approx \prod_{j=1}^{i-1} \frac{v_j}{\alpha/\lambda + v_j} \tag{25}$$

In addition, there are likely simple, rule of thumb extensions we could do that would increase the precision

(or help us bound the precision). This approximation may be useful at the start of a run as well or to find good ICs.

Noting that the reasoning which led to equations 2 and 3 for the pausing time model should still apply if we use Equation (9) for $p_{g,i}$, rather than Equation (1). As a result,

$$\Pr\left(Y_{g,i}|\alpha_c, \lambda'_c, \iota_g, E\left[\sigma_g(i-1)\right]\right) = \frac{\Gamma\left(\alpha_c + Y_{g,i}\right)}{\Gamma\left(\alpha_c\right) Y_{g,i}!}\left(\frac{\iota_g E\left[\sigma_g(i-1)\right]}{\lambda'_c + \iota_g E\left[\sigma_g(i-1)\right]}\right)^{Y_{g,i}}\left(1 - \frac{\iota_g E\left[\sigma_g(i-1)\right]}{\lambda'_c + \iota_g E\left[\sigma_g(i-1)\right]}\right)^{\alpha_c} \tag{26}$$

or, equivalently

$$= \frac{\Gamma\left(\alpha_c + Y_{g,i}\right)}{\Gamma\left(\alpha_c\right) Y_{g,i}!}\left(\frac{\iota_g E\left[\sigma_g(i-1)\right]}{\lambda'_c + \iota_g E\left[\sigma_g(i-1)\right]}\right)^{Y_{g,i}}\left(\frac{\lambda'_c}{\lambda'_c + \iota_g E\left[\sigma_g(i-1)\right]}\right)^{\alpha_c} \tag{27}$$

where n_g is the number of amino acids encoded in gene g, i.e. $n_g \sum_c n_g^c$. As before, $\kappa_g \times m_g = \iota_g$ and $\lambda'_c = \lambda_c Z/Y$. Note also that $Y_{g,i}!$ can be dropped from any Likelihood calculations and that the $p_{g,i}$ terms used to calculate Z include $\sigma(i-1)$ terms. Further, note that Z implicitly has the average value of ι_g in it and, so our choice of scale for ι_g will, in turn, scale, Z. Thus, in this model we have both λ_c and λ'_c and, as a result, we have an additional, genome wide parameter to estimate $U = Z/Y$.

For completeness, we define ϕ_g as the target production rate of the functionality produced by a complete, error free protein, i.e. $\phi_g = \iota_g \sigma_g(n_g)$. (For completeness, we note that in SelON we measure functionality production rate which is $\phi_g \times B$ where B is the functionality of a complete error free protein as encoded relative to the optimal sequence. Since we don't try to infer the optimal sequence here, ψ is irrelevant.) Thus, in the absence of translation errors, ϕ_g is equal to the total synthesis rate protein g, ι_g. In contrast, in the presence of translation errors, $\iota_g > \phi_g$ since the translation initiation rate must be elevated to compensate for the reduction in protein functionality due to translation errors. Furthermore, while the NSE model requires the likelihood for each position be calculated separately, the underlying terms for $E\left[\sigma_g(i-1)\right]$ need only be calculated once per parameter evaluation since it is only how these terms are combined that varies between codons at different positions.

Simulation

Unlike with the Pausing Time Model, we cannot aggregate codon specific RPF counts within a gene. Instead we have to simulate each position and as before we simulate in two steps. First, we generate $X_{g,i}$ from

Gamma(α_c, λ'_c) and then we generate $Y_{g,i}$ from Poisson $(\iota_g \, \sigma_g(i-1) \, X_{g,i})$. Note we are using $\sigma_g(i)$ as defined in Eq. (12) and not $E\left[\sigma_g(i)\right]$, because when simulating data, $w_{g,i}$ is known and, as a result, integration is unnecessary. Note that ι_g values are scaled by the footprinting sequencing effort. Thus, we may find that you need scale all of your ι_g values up or down by a constant term such that the expected number of footprints is on par with the number observed for a given set of genes.

Parameter Definitions

	Definition	Units
$w_{g,i}$	Ribosome waiting/pausing/dwell time at codon position i in gene g	1/t
α_c, λ_c	Shape and rate parameter for distribution of waiting times for codon c . The rate parameter is inversely related to average wait time, i.e. for codon c $E[w_{g,i}] = \alpha_c/\lambda_c$	-
m_g	Density of mRNA transcripts for gene g in cytosol.	1/Vol
M_g	Observed mRNA counts from RNA-Seq data.	1/Vol
κ_g	Rate constant determining ribosome initiation rate per mRNA. Function of diffusion of ribosomes, mRNA, and other factors.	$\frac{1}{\text{rib. mRNA Vol}t}$
ϕ_g	Target protein functionality production rate under a given set of conditions. Equal to $m_g\kappa_g\sigma_g(n_g)$ which is initiation rate discounted by the expected functionality of a protein.	1/t
ι_g	Rate of initiation of protein synthesis under a given set of conditions. Equal to $m_g\kappa_g = \phi_g/\sigma_g(n_g)$ and, thus, is always greater than or equal to ϕ_g.	1/t
$\sigma_g(i)$	Probability ribosome reaches and successfully translate codon i.	
$E\left[\sigma_g(i)\right]$	Expectation of $\sigma_g(i)$ used when fitting model to data.	
$p_{g,i}$	Probability a ribosome is found at position i on an mRNA transcript from gene g when translation initiation and completion of mRNA is at steady state.	
$P_{g,i}$	Probability of observing a footprint for position i of mRNA from gene g.	
P_g^c	Probability of observing a footprint for codon c of mRNA from gene g.	
Z	Partition function which scales the codon footprint sampling probabilities $P_{g,i}$ summed across i and g equals 1.	
n_g^c	Number of codons type c in mRNA of gene g.	
$Y_{g,i}$	Number of RFP observed for position i in gene g	
Y_g^c	Number of RFP observed for codon c in gene g	
Y	Totalnumber of RFP in dataset.	
λ'_c	Composite parameter equal to $\lambda_c Z/Y$	
π_j	Prior probability for parameter j.	

Table 1: Table of model parameters

References

C. Forbes, M. Evans, N. Hastings, and B. Peacock. Statistical Distributions. Wiley, Hoboken, NJ, 4th edition, 2011.

M. A. Gilchrist and A. Wagner. A model of protein translation including codon bias, nonsense errors, and ribosome recycling. Journal of Theoretical Biology, 239:417–434, 2006.

K. Oldham, J. Myland, and J. Spanier. An Atlas of Functions. Springer, New York, 2nd edition, 2009.

C. Pop, S. Rouskin, N. T. Ingolia, L. Han, E. M. Phizicky, J. S. Weissman, and D. Koller. Causal signals between codon bias, mrna structure, and the efficiency of translation and elongation. Molecular Systems Biology, 10(12):770, 2014. ISSN 1744-4292. doi: 10.15252/msb.20145524. URL http://dx.doi.org/10.15252/msb.20145524. 770.