Comparing two models of epistasis

The point of this document is to compare and contrast an alternative model of allelic interactions to the model that we present in our upcoming science submission(henceforth the Science Submission Model, or SSM). First I go through the model, and discuss the various methods by which its parameters can be determined and how its predictive power changes under different parameterizations. Then, I contrast this model with the **S**cience **S**ubmission **M**odel to show how despite their very different dynamics, they perform in fundamentally similar ways.

Finally I conclude by showing that although the models are equivalent, and examination of the available parameter space shows that if the simple linear model was true, we would be very unlikely to obtain as good a fit to the data as we do with our SSM, which implies that the SSM as it stands is the correct one.

The New Model

A second model of epistasis posits, henceforth the Simple Linear Model or **SLM**, posits that each allele has a set effect on the fitness of a strain, that is for the fitness of the single mutant (labeling the 1st, 2nd, 3rd and 4th mutants as A, B, C, D respectively) would be:

$$W_{1000} = 1 + A$$
$$W_{0100} = 1 + B$$
$$W_{1100} = 1 + A + B$$

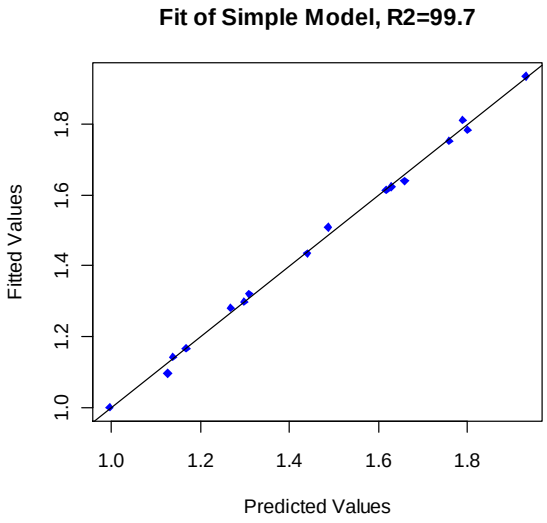
And in general that $W_i = 1 + I_A A + I_B B + I_C C + I_D D$ where I_Z is an indicator variable equal to 1 if the relevant mutation is present in genotype i and 0 otherwise.

Different Ways to Parameterize the Model

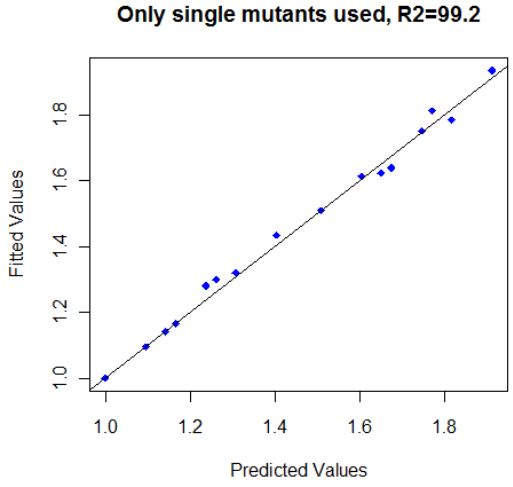
Our data from this study can be represented in the adjoining matrix, where the fitness is the value we are trying to predict and the columns A,B,C,D are representative of the information we will use to predict these fitnesses. That is, they are the value of the indicator variables in the above model for the presence of a beneficial mutation on a genotype.

In order to fit this model, we can use the entire data matrix (that is all fitness observations) simultaneously, to find the set of parameters which by the least squares criteria give the single best fit. This approach shows a very good fit of the model to the data, giving an R2 value of .997, which is a nearly perfect fit given the errors on our measurements.

Fitness	A	B	C	D
1.935	1	1	1	1
1.752	0	1	1	1
1.784	1	0	1	1
1.812	1	1	0	1
1.435	1	1	1	0
1.623	0	0	1	1
1.614	0	1	0	1
1.281	0	1	1	0
1.639	1	0	0	1
1.32	1	0	1	0
1.299	1	1	0	0
1.509	0	0	0	1
1.142	0	0	1	0
1.096	0	1	0	0
1.166	1	0	0	0
1	0	0	0	0

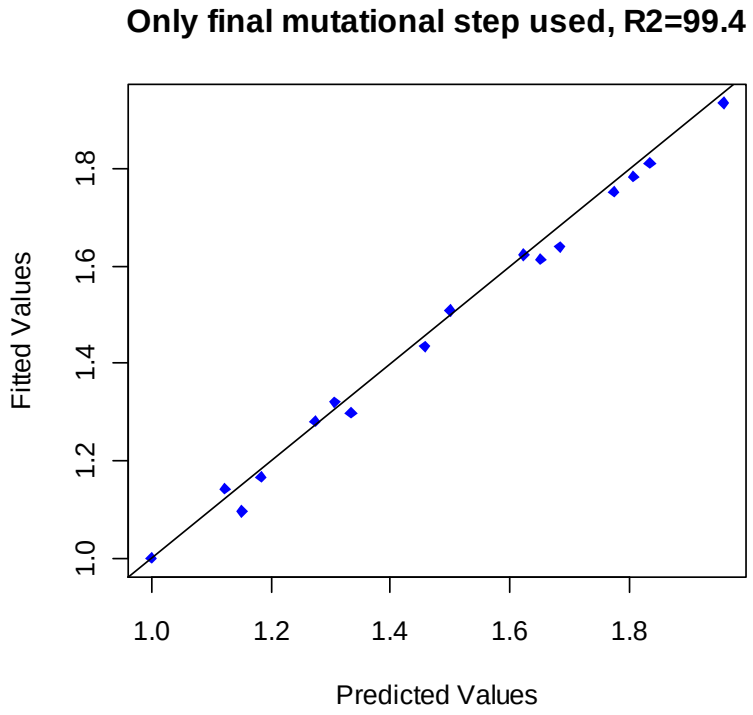


However, this does not provide an entirely fair comparison to the SSM because the parameters in the SSM were estimated with only data from the first mutational step. A simple estimate of the effect of a mutation on the fitness of a strain can be obtained for any genotypes connected by a single mutational step that adds this allele. For example, the effect of the A allele can be estimated for any combinational state, XXX, of the other three alleles as $\hat{A} = W_{1XXX} - W_{0XXX}$. Using this formulation and just the fitness effect of the alleles when introduced on the wildtype background, we can again see how well we can predict the fitness of all genotypes simply from the single mutant genotypes:



By using only these four data points, we can obtain an incredibly good prediction of the fitness values of all of the genotypes (the line is equal to y=x). Similarly, we can use just the information obtained in the last mutational step to estimate the effect of each beneficial mutation. That is, what if we only knew what happened when the mutation was added to a genotype with all other mutants already there, basically if we only removed the allele from the fittest genotype instead of adding it to the wildtype.

Again, we see that the prediction is incredibly good, with only four data points we can predict the fitness values of the remaining 11 incredibly well. If generally true, this means that ***there is no epistasis for this trait and we can perfectly predict a fitness landscape of 2^N genotypes with only N observations.*** This is pretty efficient and awesome.



Finally, we can ask what happens if instead of using either all the mutational effects in the first step, or all of the effects in the last step, we use the fitness differences involved in going from 2->3 mutations, or combinations of the first, second, third or final step to estimate the model. I haven't done this yet but am confident that the results for any combination would still be as good because our prediction as it stands is essentially perfect, meaning that any combination used to estimate must give excellent results.

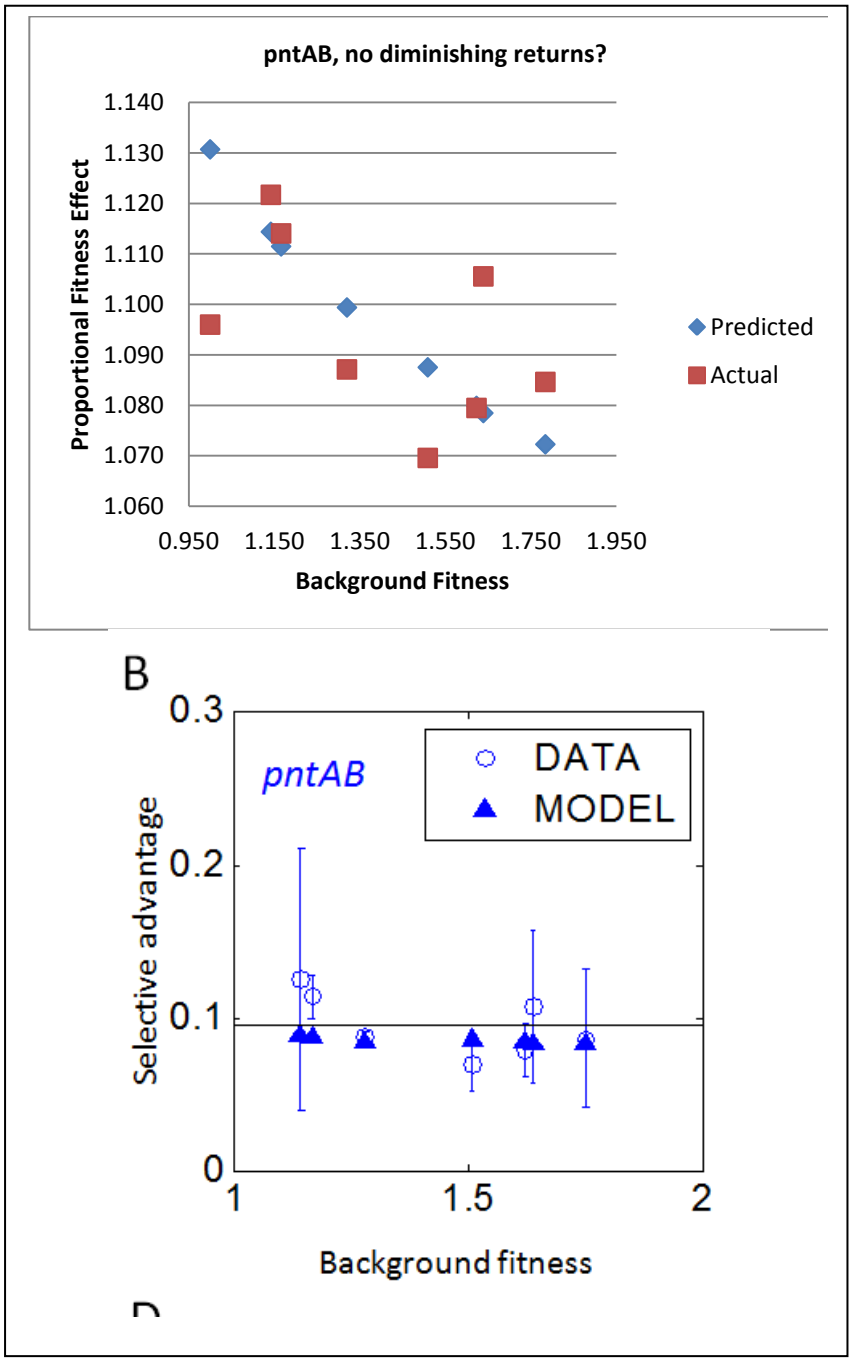
Model Comparison

The SLM and the SSM are very different, yet both provide similar predictions and behave in similar ways. This section explains the reasons that they agree and do not agree.

Diminishing returns is a feature of both models. In the SSM this is a consequence of the "cost" portion of the model, any mutation that diminishes the cost shows diminishing returns. In contrast, in the SLM, diminishing returns is simply a natural consequence of the model structure. For illustration, consider a fitness landscape in which every allele had the same constant effect (e) on the growth rate. The proportional effect of a beneficial mutation added to a background that already had n beneficial mutations would then be equal to:

$$E = \frac{W_+}{W_-} = \frac{1 + (N+1)e}{1 + Ne}$$

Clearly, as $N \to \infty, E \to 0$ which is clear diminishing returns behavior. In the SLM model, all alleles show diminishing returns epistasis. However, in the SSM only the mutations that have a cost reduction component shows diminishing returns. This allowed for pntAB to not show diminishing returns in the SSM, and if the SLM is correct, we must explain why we did not find evidence for diminishing returns with this allele. Two points are relevant here. First, although figure 3B in the science submission is reasonably flat, plowing up the y-axis and comparing it to the results one would obtain if the SLM were completely correct shows that the trend of diminishing returns, while not prominent, is hinted at by the data (hence the small cost reduction in the SSM for this allele).



Model Equivalency

Why is it that we would expect both models to give such similar results, despite their different formulation. In the SSM, the fitness is given as:

$$W_{mutant} = \prod_{i \in Alleles} \lambda_i b_0 - \prod_{i \in Alleles} \theta_i c_0$$

Below, I introduce some matrix notation and calculate the expected residuals of the linear model fit if the SSM is correct, I do this to show how the models are able to give equivalent results.

Now for matrix notation:

$$\begin{aligned}
&\boldsymbol{z} = Vector\ of\ indicator\ variables\ for\ the\ presence\ of\ an\ allele \\
&\boldsymbol{Z} = Matrix\ of \boldsymbol{z}s \\
&\boldsymbol{\lambda} = Vector\ of \log of\ benefit\ terms \\
&\boldsymbol{\theta} = Vector\ of \log of\ \ cost\ terms \\
&\boldsymbol{w} = Vector\ of fitnesses \\
&\hat{\boldsymbol{e}} = Vector\ of\ estimated\ linear\ effects \\
&\hat{\boldsymbol{w}} = Vector\ of\ estimated\ fitnesses \\
&\mathbf{1}_{\boldsymbol{N}} = a\ vector\ of\ N\ ones \\
&\boldsymbol{I_n} = Identity\ matrix\ of\ size\ N
\end{aligned}$$

I also introduce the following notation, for a vector $\boldsymbol{r}$ the operation $\boldsymbol{q} = e^{\boldsymbol{r}}$ means creating a new vector whose elements are equal to e raised to the old vectors elements, that is $\begin{matrix} q_1 & e^{r_1} \\ q_2 = & e^{r_2} \\ q_3 & e^{r_3} \end{matrix}$.

For our particular model, these vectors are represented as below.

$$\boldsymbol{Z} = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix} \quad \lambda = \begin{matrix} .0797 \\ .0592 \\ .0649 \\ .3097 \end{matrix} \quad \boldsymbol{\theta} = \begin{matrix} -.007 \\ -.7236 \\ -1.0161 \\ -1.1301 \end{matrix} \quad \hat{\boldsymbol{e}} = \begin{matrix} .096 \\ .142 \\ .166 \\ .509 \end{matrix}$$

Assuming that the SSM is correct, this implies that the fitness of each strain can be calculated as below:

$$\boldsymbol{b} = \boldsymbol{Z\lambda}$$
$$\boldsymbol{c} = \boldsymbol{Z\theta}$$

$$\boldsymbol{w} = \boldsymbol{e^b}(1 + c_0) - \boldsymbol{e^c}c_0 = \boldsymbol{e^{Z\lambda}} + c_0\left(\boldsymbol{e^{Z\lambda}} - \boldsymbol{e^{Z\theta}}\right)$$

Now, if this model is true and we instead fit a linear model on the dataset, how would the results look? There are several ways to fit the data to a linear model. Either all datapoints could be used, or just the first mutational steps, or any combination there off. If just the first mutational steps are used, then:

$$\hat{\boldsymbol{e}} = e^{\boldsymbol{\lambda}}(1 + c_0) - e^{\boldsymbol{\theta}}c_0 - \mathbf{1_4} = e^{\boldsymbol{\lambda}} + c_0\left(\boldsymbol{e^{\lambda}} - e^{\boldsymbol{\theta}}\right) - \mathbf{1_4}$$
$$\hat{\boldsymbol{w}} = \mathbf{1_{16}} + \boldsymbol{Z}\,\hat{\boldsymbol{e}}$$

Writing in this notation allows us to calculate the expected residuals between the linear fit model and the actual values as:

$$\boldsymbol{r} = vector\ of\ residuals$$
$$\boldsymbol{r} = \boldsymbol{w} - \hat{\boldsymbol{w}} = \boldsymbol{e^{Z\lambda}} + c_0\left(\boldsymbol{e^{Z\lambda}} - \boldsymbol{e^{Z\theta}}\right) - (\mathbf{1_{16}} + \boldsymbol{Z}\left(e^{\boldsymbol{\lambda}} + c_0\left(\boldsymbol{e^{\lambda}} - e^{\boldsymbol{\theta}}\right) - \mathbf{1_4}\right))$$

Any element of $\boldsymbol{r}$ is going to be equal then to:

$$r_i = e^{b_i} + c_0(e^{b_i} - e^{c_i}) - 1 - Z_{i,1:4}\left(e^{\boldsymbol{\lambda}} + c_0\left(\boldsymbol{e^{\lambda}} - e^{\boldsymbol{\theta}}\right) - \mathbf{1_4}\right)$$

$$r_i = e^{Z_{i,1:4}\boldsymbol{\lambda}} + c_0\left(e^{Z_{i,1:4}\boldsymbol{\lambda}} - e^{Z_{i,1:4}\boldsymbol{\theta}}\right) - 1 - \left(Z_{i,1:4}e^{\boldsymbol{\lambda}} + c_0\left(Z_{i,1:4}e^{\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\theta}}\right) - Z_{i,1:4}\mathbf{1_4}\right)$$
$$r_i = e^{Z_{i,1:4}\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\lambda}} + c_0\left(\left(e^{Z_{i,1:4}\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\lambda}}\right) - \left(e^{Z_{i,1:4}\boldsymbol{\theta}} - Z_{i,1:4}e^{\boldsymbol{\theta}}\right)\right) - 1 + Z_{i,1:4}\mathbf{1_4}$$

$$r_i = e^{Z_{i,1:4}\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\lambda}} + c_0\left(\left(e^{Z_{i,1:4}\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\lambda}}\right) - \left(e^{Z_{i,1:4}\boldsymbol{\theta}} - Z_{i,1:4}e^{\boldsymbol{\theta}}\right)\right) - 1 + Z_{i,1:4}\mathbf{1_4}$$

We note that the pattern shown below (where $\boldsymbol{x}$ is some vector) is found throughout the equation for the residuals. We explore this pattern to show how it influences the residuals.

$$e^{Z_{i,1:4}\boldsymbol{x}} - Z_{i,1:4}e^{\boldsymbol{x}}$$

Since $\boldsymbol{Z_i}$ is a vector of 1's and zeros the only terms that show up in this are the corresponding terms in $\boldsymbol{x}$ where the z_i value is positive, suppose indexes i, j, k in $\boldsymbol{Z_{i,:}}$ are positive, then:

$$e^{Z_{i,1:4}\boldsymbol{x}} - Z_{i,1:4}e^{\boldsymbol{x}} = e^{x_i + x_j + x_k} - e^{x_i} + e^{x_j} + e^{x_k}$$

Here we can give an alternate definition for $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}$ … allowing us to rewrite:

$$e^{x_i + x_j + x_k} \approx 1 + \left(x_i + x_j + x_k\right) + \frac{9O(x_a x_b\,)}{2} + \cdots$$

where $O(x_a x_b\,)$ are terms on the order of two values from the vector $\boldsymbol{x}$. Similarly:

$e^{x_i} + e^{x_j} + e^{x_k} \approx \left(1 + x_i + \frac{O(x^2)}{2}\right) + \left(1 + x_j + \frac{O(x^2)}{2}\right) + \left(1 + x_k + \frac{O(x^2)}{2}\right) \approx 3 + x_i + x_j + x_z + 3O(x_a x_b)$

Stating both of these more generally we see that

$$e^{Z_{i,:}\boldsymbol{x}} \approx 1 + \boldsymbol{z_{i,:}x} + \left(\frac{1}{2}\right)\mathbf{1}_n diagonal\left(\boldsymbol{z_{i,:}^T x^T}\right) diagonal\left(\boldsymbol{z_{i,:}^T x^T}\right)^{\boldsymbol{T}} \mathbf{1}_N^T$$

$$Z_{i,:}e^{\boldsymbol{x}} \approx \boldsymbol{z_{i,:}z_{i,:}^T} + \boldsymbol{z_{i,:}x} + \left(\frac{1}{2}\right) diagonal\left(\boldsymbol{z_{i,:}^T x^T}\right)^{\boldsymbol{T}} diagonal\left(\boldsymbol{z_{i,:}^T x^T}\right)$$

Taking the difference of these approximations we get:

$$e^{Z_{i,1:4}\boldsymbol{x}} - Z_{i,1:4}e^{\boldsymbol{x}} \approx 1 - \boldsymbol{z_{i,:}z_{i,:}^T} + \frac{1}{2}\sum_{i=1}^{N} z_i x_i \sum_{j \neq i} z_j x_j$$

The notation above has some important interpretations. The term $\boldsymbol{z_{i,:}z_{i,:}^T} = Z_{i,1:4}\mathbf{1_4}$ is simply the number of beneficial mutations present on the ith background. Similarly, for the ith background $\frac{1}{2}\sum_{i=1}^{N} z_i x_i \sum_{j \neq i} z_i x_j$ is equal to the summation of the effect of each allele present multiplied by every other allele present, excluding multiplications to itself, which cancelled out with the other term. For small x_i these terms are all small, and on a background with K beneficial alleles there are $K^2 - K$ of them, meaning this error term gets larger with more beneficial mutations present and is absent with one mutation present.

Since the $\boldsymbol{\lambda}$ values are small, we can use this approximation to obtain a reduced form of r_i:

$$\boldsymbol{r_i} = 1 - \boldsymbol{z_{i,:}z_{i,:}^T} - \frac{1}{2}\sum_{i=1}^{N} z_i\lambda_i \sum_{j \neq i} z_j\lambda_j + c_0\left(\left(e^{Z_{i,1:4}\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\lambda}}\right) - \left(e^{Z_{i,1:4}\boldsymbol{\theta}} - Z_{i,1:4}e^{\boldsymbol{\theta}}\right)\right) - 1 + Z_{i,1:4}\mathbf{1_4}$$

$$\boldsymbol{r_i} = \frac{1}{2}\sum_{i=1}^{N} z_i\lambda_i \sum_{j \neq i} z_j\lambda_j + c_0\left(\left(e^{Z_{i,1:4}\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\lambda}}\right) - \left(e^{Z_{i,1:4}\boldsymbol{\theta}} - Z_{i,1:4}e^{\boldsymbol{\theta}}\right)\right)$$

Here we see that the residual value for the ith alleles is a composed of sums of products of the λs, (the $\frac{1}{2}\sum_{i=1}^{N} z_i\lambda_i \sum_{j \neq i} z_j\lambda_j$) as well as another term that is scaled by the c_0 parameter (the $\left(\left(e^{Z_{i,1:4}\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\lambda}}\right) - \left(e^{Z_{i,1:4}\boldsymbol{\theta}} - Z_{i,1:4}e^{\boldsymbol{\theta}}\right)\right)$) . In order for the residual values to be small, this value must also be small. This term is the difference of values determined by the $\boldsymbol{\lambda}$ values, $\left(e^{Z_{i,1:4}\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\lambda}}\right)$ as well as a second term determined by the θ values, $\left(e^{Z_{i,1:4}\boldsymbol{\theta}} - Z_{i,1:4}e^{\boldsymbol{\theta}}\right)$. As shown before, because the λ values are generally near 0, this difference involving them can be approximated as:

$$\left(e^{Z_{i,1:4}\boldsymbol{\lambda}} - Z_{i,1:4}e^{\boldsymbol{\lambda}}\right) \approx -\#\ of\ alleles\ present + 1 + \Delta$$

where the Δ is representative of the other smaller terms and increases with more alleles present or with alleles of larger effect (which either increase the number of additional terms or increase the size of these terms). However, because in general the values in $\boldsymbol{\theta}$ do not appear to be near zero, we cannot use the same approximation for $\left(e^{Z_{i,1:4}\boldsymbol{\theta}} - Z_{i,1:4}e^{\boldsymbol{\theta}}\right)$. However, the behavior of this term can be adequately described by noting that as before the difference is of the form:

$$e^{z_1\theta_1+z_2\theta_2+z_3\theta_3} - (z_1e_1^{\theta} + z_2e^{\theta_2} + z_3e^{\theta_3})$$

Because every $e^{\theta_i} < 1$, the more of these terms that are multiplied (more alleles present), the smaller this value becomes, while at the same time the more of these that are added, the larger the summation of ($z_1e_1^{\theta} + z_2e^{\theta_2}$ …) becomes. This means that this difference is zero if only one mutation is present, but becomes increasingly negative with more mutations present. For the particular values in our model, this increasing magnitude on backgrounds with more mutations allows the term based on the $\boldsymbol{\theta}$ values to roughly keep pace with the term based on the $\boldsymbol{\lambda}$ values. However, for larger values the magnitude of the $\boldsymbol{\lambda}$ term begins to dominate, meaning that the final value scaled by c_0 and added to the residual becomes increasingly negative. However, because the error introduced by the first half of the equation is becoming increasingly positive at the same time that this term scaled by c_0 is becoming increasingly negative, this actually allows for an improved model fit. This explains a trade off in the c_0 parameter, with values to small the fit is made poorer because there is no compensating effect, resulting in increasingly positive residuals. In contrast, at high values the term multiplied by c_0 begins to dominate the error and increasingly negative residuals are seen. Such a fortunate counter balance in the model comes at our present parameter settings, and in fact is true only for a very narrow range of c_0 values. However, discordance between the two models will increase for an increasingly larger number of mutations added to the different backgrounds.

To draw this point out a bit more, I will also compare how an either "pure cost" or "pure benefit" model would behave compared to a linear model. It shows that the errors in either model increase with additional mutations, but is of a different sign.

What if just a multiplicative model were estimated?

The canonical epistasis model can be represented as a $w_i = \prod_{i=1}^{N} z_i\phi_i$ where N are the number of alleles considered and $\phi_i > 1$ is the effect of the ith allele on fitness. If this model were true, the estimated effects from the first step mutations in the linear model would be equal to $\widehat{e_i} = \phi_i - 1$, as before we could then calculate the residuals as:

$$\widehat{\boldsymbol{w}} = 1 + \boldsymbol{Z}\hat{\boldsymbol{e}}$$
$$\boldsymbol{r} = \boldsymbol{w} - \widehat{\boldsymbol{w}} = \boldsymbol{e^{Zlog(\phi)}} - \boldsymbol{1_{16}} - \boldsymbol{Z}\hat{\boldsymbol{e}} = \boldsymbol{e^{Zlog(\phi)}} - \boldsymbol{1_{16}} - \boldsymbol{Z}(\phi - \boldsymbol{1}_4)$$

This can be rewritten defining $\boldsymbol{\Phi} = \log(\boldsymbol{\phi})$ lets us rewrite the value and make the same approximation as before (as most of these values will be near zero):

$$\boldsymbol{r} = \boldsymbol{e^{Z\Phi}} - \boldsymbol{1_{16}} - \boldsymbol{Ze^{\Phi}} + \boldsymbol{Z1}_4$$
$$r_i = \boldsymbol{e^{Z_{i,:}\Phi}} - \boldsymbol{Z_{i,:}e^{\Phi}} - \boldsymbol{1} + \boldsymbol{Z_{i,:}1}_4$$
$$r_i = \frac{1}{2}\sum_{i=1}^{N} z_i\Phi_i \sum_{j\neq i} z_j\Phi_j$$

That is, when a linear model is fit to a biological reality that mirrors the classic epistasis model, the result is the residuals become increasingly positive as more beneficial mutations are added or the mutations are of larger and larger effects.

What if just a cost model were estimated?

Alternatively, we can regard all evolution as reducing a cost imposed by some trait, and thus consider all adaptive walks as a climb up a summit with a peak at a known height. In this parameterization we could consider fitness to be equal to $w_i = W_{Max} - c_0 \prod_{i=1}^{N} z_i \gamma_i$, where $0 < \gamma_i \leq 1$ is the proportion of the cost that is reduced by a given beneficial mutation, and making the parameterization of $\boldsymbol{\Gamma} = \log(\boldsymbol{\gamma})$ allows us to write the fitness values and residuals as:

$$\boldsymbol{w} = W_{Max}\mathbf{1_{16}} - c_0 e^{\boldsymbol{Z\Gamma}}$$
$$\hat{\boldsymbol{e}} = c_0\left(\mathbf{1}_4 - e^{\boldsymbol{\Gamma}}\right)$$
$$\hat{\boldsymbol{w}} = W_{Max}\mathbf{1_{16}} - c_0\mathbf{1_{16}} + \boldsymbol{Z}\hat{\boldsymbol{e}}$$
$$\boldsymbol{r} = \boldsymbol{w} - \hat{\boldsymbol{w}} = W_{Max}\mathbf{1_{16}} - c_0 e^{\boldsymbol{Z\Gamma}} - \left(W_{Max}\mathbf{1_{16}} - c_0\mathbf{1_{16}} + \boldsymbol{Z}c_0\left(\mathbf{1}_4 - e^{\boldsymbol{\Gamma}}\right)\right)$$
$$\boldsymbol{r} = -c_0 e^{\boldsymbol{Z\Gamma}} + c_0\mathbf{1_{16}} - \boldsymbol{Z}c_0\mathbf{1}_4 + \boldsymbol{Z}c_0 e^{\boldsymbol{\Gamma}}$$
$$\boldsymbol{r} = c_0(\mathbf{1_{16}} - \boldsymbol{Z}\mathbf{1}_4 + \boldsymbol{Z}e^{\boldsymbol{\Gamma}} - e^{\boldsymbol{Z\Gamma}})$$

As before, for multiple mutations present on a background this residual becomes increasingly negative and becomes equivalent to: $r_i = c_0(1 - \#\ of\ mutations + \Delta)$, where delta is a positive term that is composed of the sum of mutational effects minus the products of these effects. Although small c_0 values would reduce the size of all residuals, it would also imply that not much of an increase in fitness was possible and so this adaptive walk would not be biologically very interesting.

Which Model is Right?

With such concordance between the two models, it is natural to ask which model we might prefer? The SSM and the SLM are able to agree with each other because of a happy scaling between the cost and benefit terms. Although the discordance between the two models is expected as additional mutations are added, so in theory we could use the other mutations to test this. Alternatively, we could ask, if the simple linear model were true, how likely would we be to obtain such a good fit with the SSM? That is, is there real significance to our model? Since the SSM parameterizes the benefit as the difference between the observed fitness and the observed cost, it is natural to ask if random values for the plasmid cost (c_0) and for the reduction in this cost provided by each mutation ($the\ \theta_i's$) would allow for an equivalently good fit. Simulations are shown below for these values.

The results show that if the SLM were true, we are extremely unlikely to have obtained as good a fit to the data if we used values that were unrelated to the biological reality. This gives strong support to the SSM being the correct one.

If the SLM were true, would we expect to be able to always fit a model that showed good agreement with the data?

Table 1: No Constraints on Parameters

Test	C_0	θs	Fitness Values of 1st Step mutations	Model Fit	P-Value for As good r better result	Conclusion
Can c_0 vary?	.261*Beta(1,10)	Fixed	Fixed	Cost Benefit	.14	The particular c_0 parameter is reasonably important to the fit
Can cell folding vary?	Fixed	Beta(1,1)	Fixed	Cost Benefit	.017	The particular θ parameters were very important
Can both vary?	Beta(1,10)	Beta(1,1)	Fixed	Cost Benefit	.007	The particular combination of parameters is very important.

Table 2: Constrains enforced $\lambda_i \geq 1$

Test	C_0	θs	Fitness Values of 1st Step mutations	Model Fit	P-Value for As good r better result	Conclusion
Can c_0 vary?	Beta(1,10) Truncated at .261	Fixed	Fixed	Cost Benefit	.145	The particular c_0 parameter is reasonably important to the fit
Can cell folding vary?	Fixed	Uniform over range	Fixed	Cost Benefit	.0252	The particular θ parameters were very important
Can both vary?	Beta(1,10) Truncated at .261	Uniform over allowable range	Fixed	Cost Benefit	.008	The particular combination of parameters is very important.

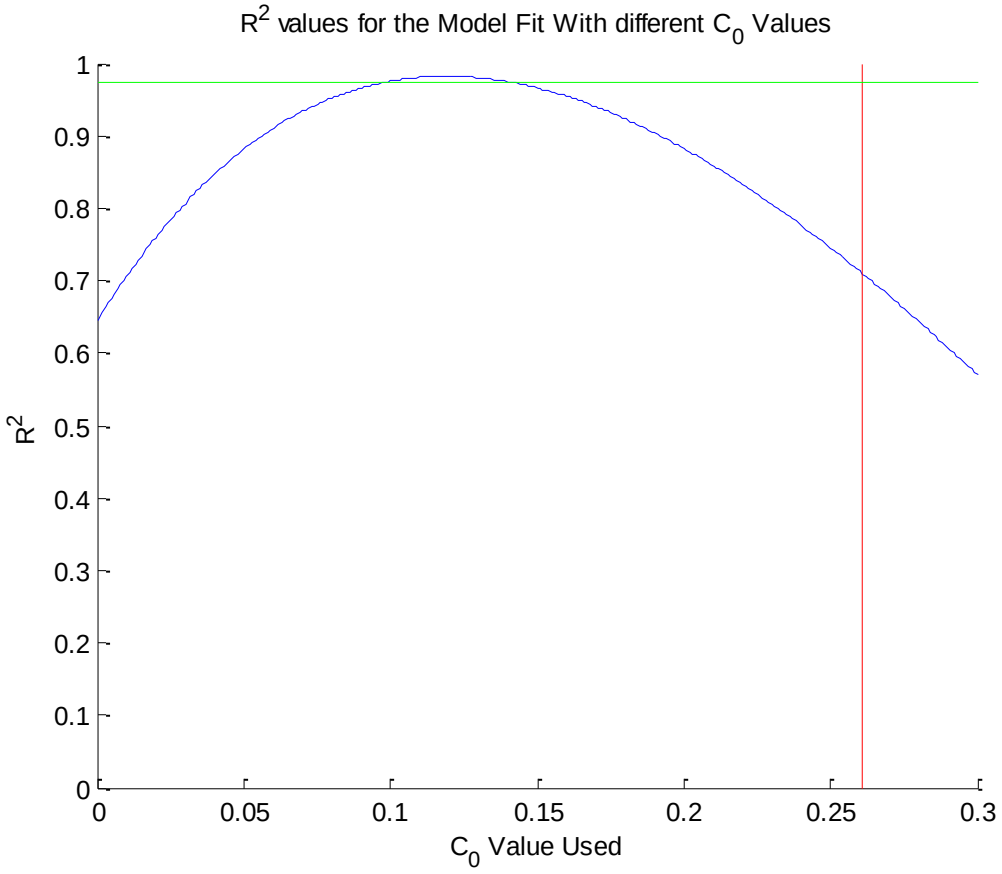


Figure C0: Demonstration of how the model fit changes for different C_0 values. The green line represents the R^2 value at the empirically determined model settings, and only values in the range $.098 \leq c_0 \leq .141$ provide an equivalent or better fit. The $\beta(1,10)$ distribution has significant density in this region giving the p-value shown in the table, but this value is entirely dependent on the assumption that the distribution of costs will have significant density in this narrow range. The red line indicates the highest c_0 value possible with this dataset that maintains the constraint that $\lambda_i \geq 1$ for all i. Note that the $\beta(1,10)$ distribution spans the domain 0 to 1, and so does not always allow for this constraint to be met.

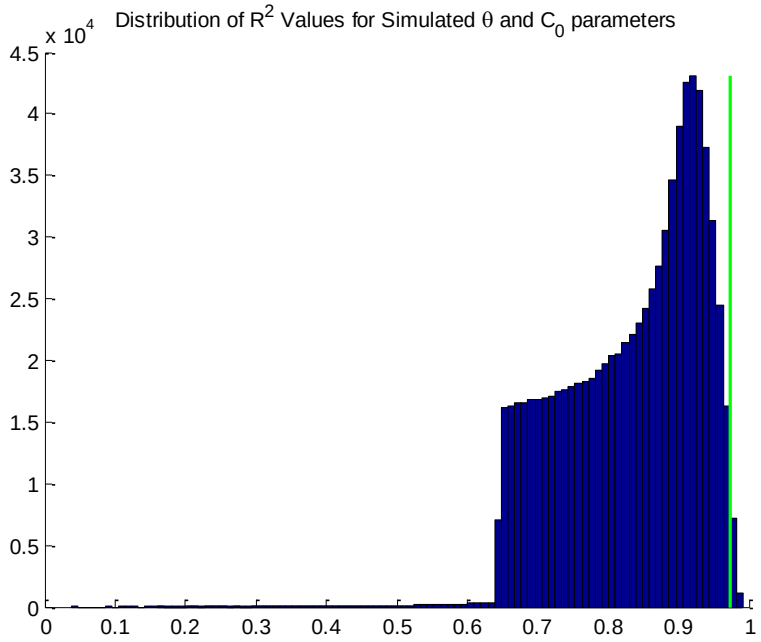


Figure $\theta and\ c_0$: Distribution of R^2 values from >825,000 simulations with c_0 parameters drawn from a truncated $\beta(1,10)$ distribution and the θ vector drawn uniformly from the 4 dimensional unit hypercube and constrained so the $\lambda \geq 1$. Although only a minute fraction (.8%)of these values give a fit as good or better than that from the empirically determined values, in general a large area of this parameter space provides a good fit, with ~65% of values giving an $R^2 \geq .8$.

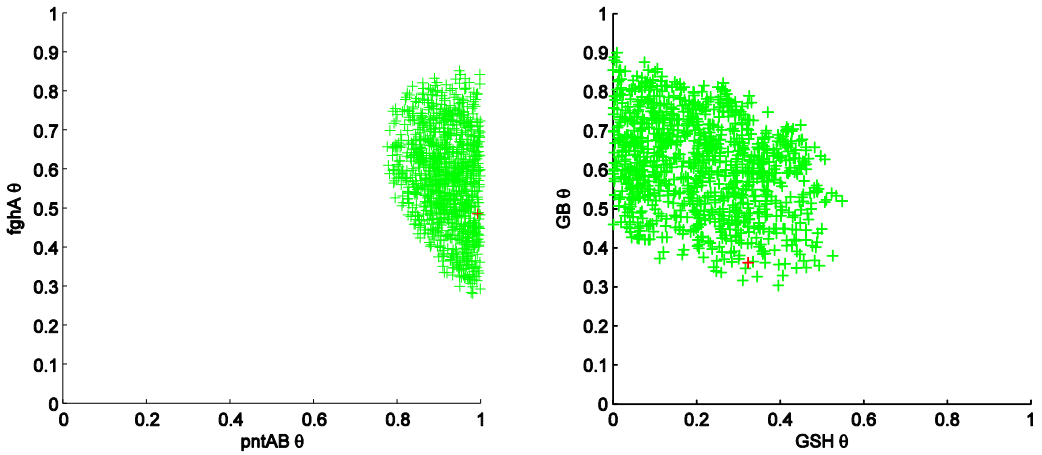


Figure θ: Green points represent simulated θ values whose SSE was in the lowest 1% of all simulated values (n=100,000). Each theta value was randomly drawn from a uniform(0,1) distribution, and the empirically estimated values are shown in red.