

---

# Modelling Competition for Nutrients between Microbial Cultures Growing on Solid Agar Surfaces

Author: Daniel Boocock; Supervisor: Dr Conor Lawless

August 4, 2016

---

## ABSTRACT

**Motivation:** The fitness of microbial strains can be estimated from the growth of cultures inoculated onto solid agar. In high-throughput procedures, an array of cultures is grown on the same agar plate and competition for nutrients between cultures is likely to affect growth. However, analysis assumes that cultures grow independently. We test a model of nutrient dependent growth and diffusion and try to correct for competition to provide more accurate and reliable fitness estimates.

**Results:** What should we say?

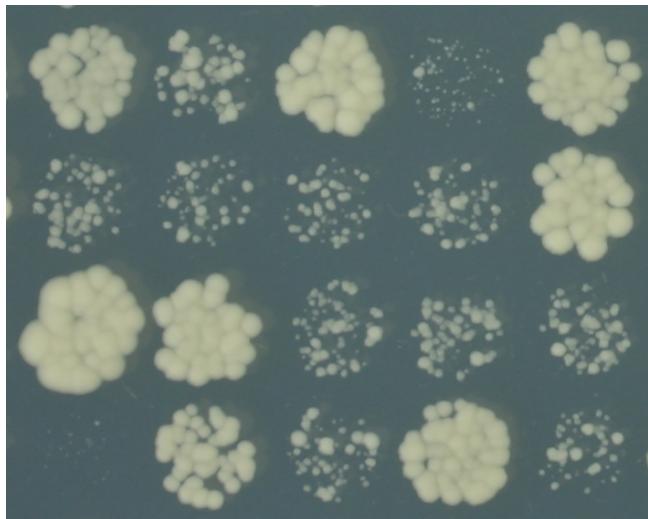
**Availability and Implementation:** CANS, a Python2 package developed for the analysis in this paper, is freely available from (github or PyPI).

**Contact:** daniel.boocock@protonmail.ch

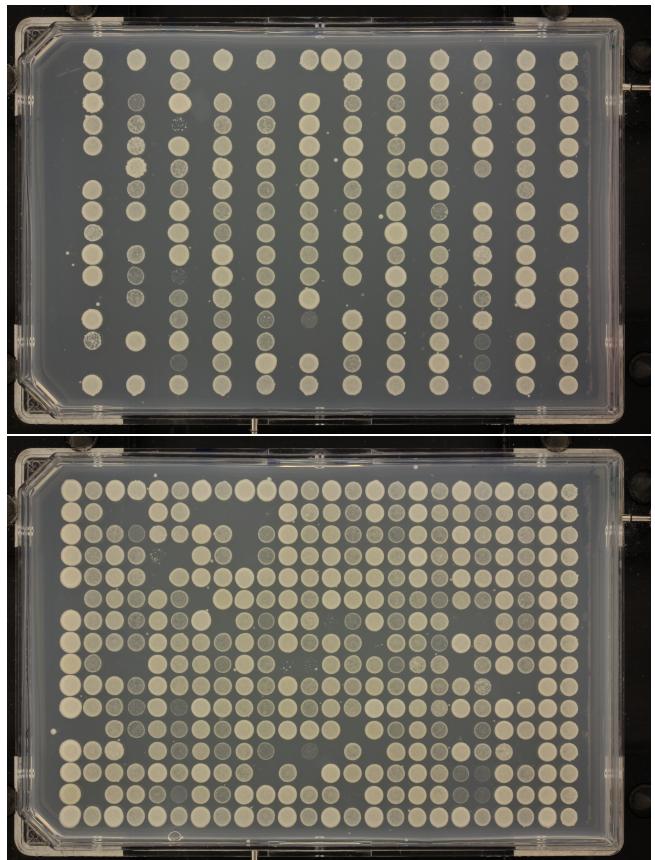
## 1 INTRODUCTION

Dummy Lawless *et al.* (2010) citations (Heydari *et al.*, 2016) (Addinall *et al.*, 2008).

### 1.1 Subsection



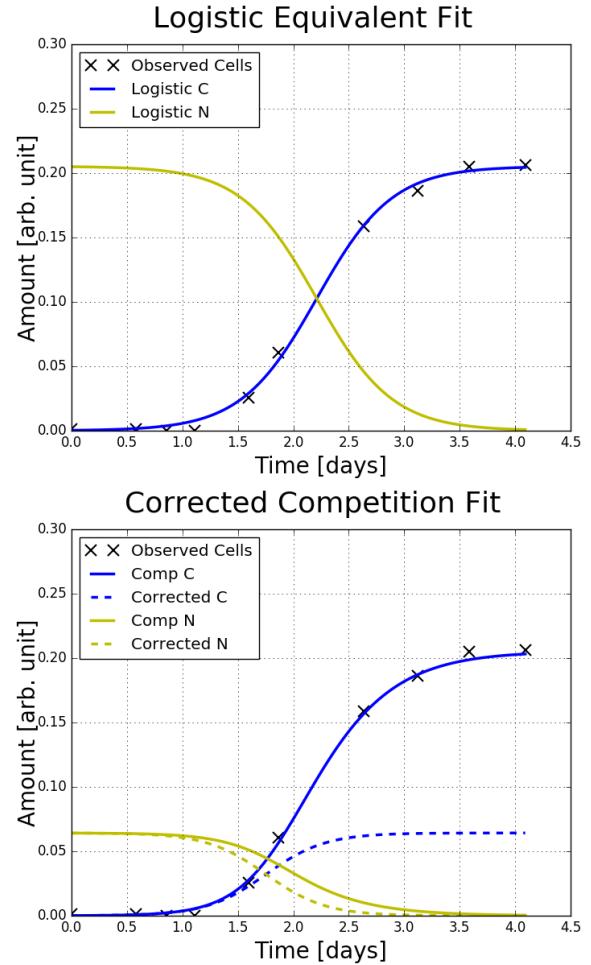
**Figure 1:** A section of a plate from a QFA experiment (Addinall *et al.*, 2011).



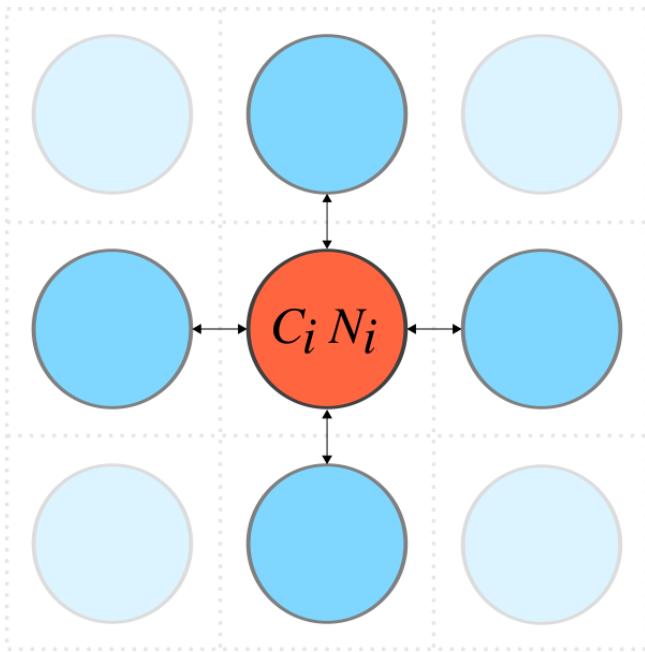
**Figure 2:** Images from an experiment designed to examine competition.

## 2 METHODS

### 2.1 Subsection



**Figure 4:** PUT KN VALUES (and r and K) ON THE PLOT. (Could even put obj fun values). Fits to culture (R10, C3) of P15 (Addinall et al., 2011) illustrating how the competition model can be seen as a correction to the logistic model. C - Cells; N - Nutrients. Top - Logistic Equivalent Fit; Bottom - Competition Fit (solid) and Corrected Logistic Equivalent Fit (dashed).



**Figure 3:** Schematic of the modelling approach. Each circle represents a culture, indexed  $i$ , on solid agar and arrows represent diffusion of nutrients.  $C_i$  - amount of cells;  $N_i$  - amount of nutrients; darker blue circles- neighbourhood  $\delta_i$ .

## 3 RESULTS

(Palková et al., 1997)

### 3.1 Guessing

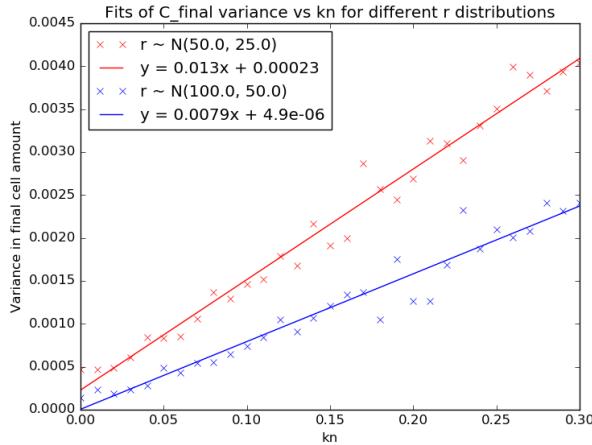


Figure 5: Guessing  $k_n$  from variation in final cell amounts.

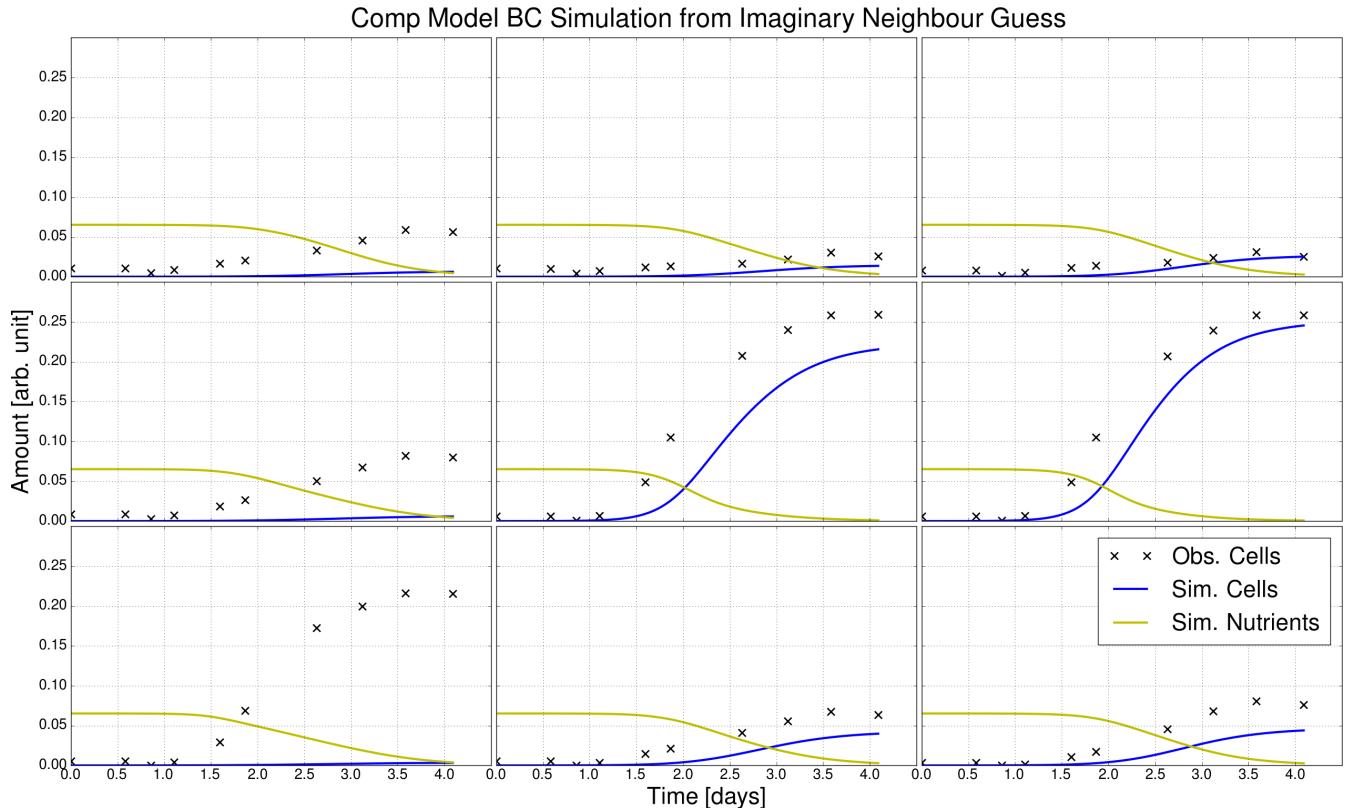


Figure 6: Comp Model BC simulation using parameters from Imaginary Neighbour guessing of P15 for cdc13-1 P15 at 27C (R5, C18). NOT TO GO HERE This method of guessing requires a b-guess to be supplied to fix the faster growing neighbour. I iterated through cell ratios. I iterated through a range of b guesses supplied at the plate level; running a different script with a  $C_0$  guess,  $b_{guess}$ . It would probably have been better iterate through a list of b-guess values for each culture and choosing the estimated b value from the best fit of each culture. Guessing time is currently about four minutes which is fast compared to fitting which takes approximately three hours. However, this is unlikely to stop us from encountering local minima when we fit the Competition Model.

Scripts were run with combinations of the following values. `cellratios = np.logspace(-3, -5, num=5)` `fittype = ["imagineigh", "logeq"]` `zerokn = [True, False]`

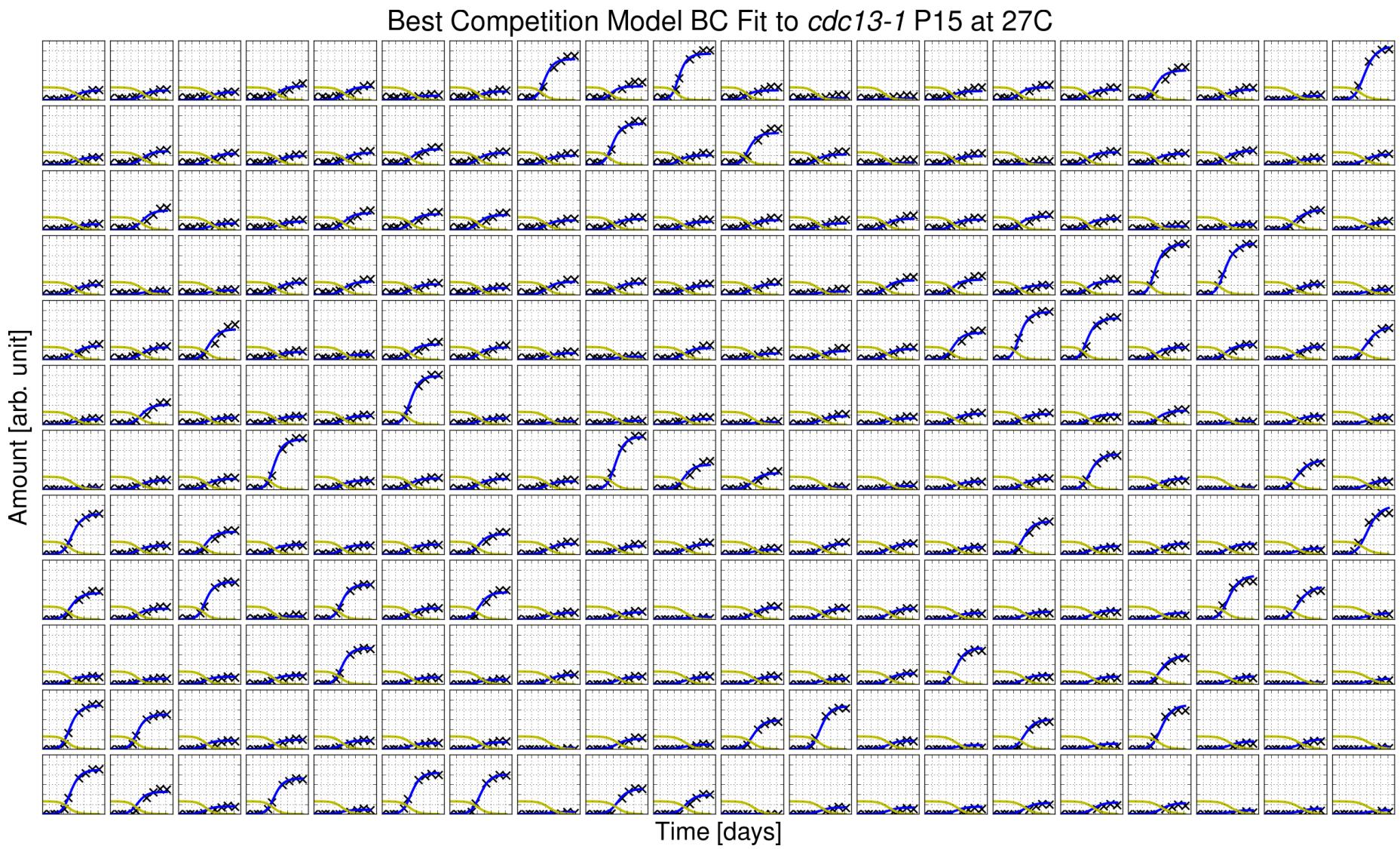
Each script looped through the following array of b values which were supplied to the initial guesser and used at the plate level.

for b-guess in [35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 95, 100, 150]:

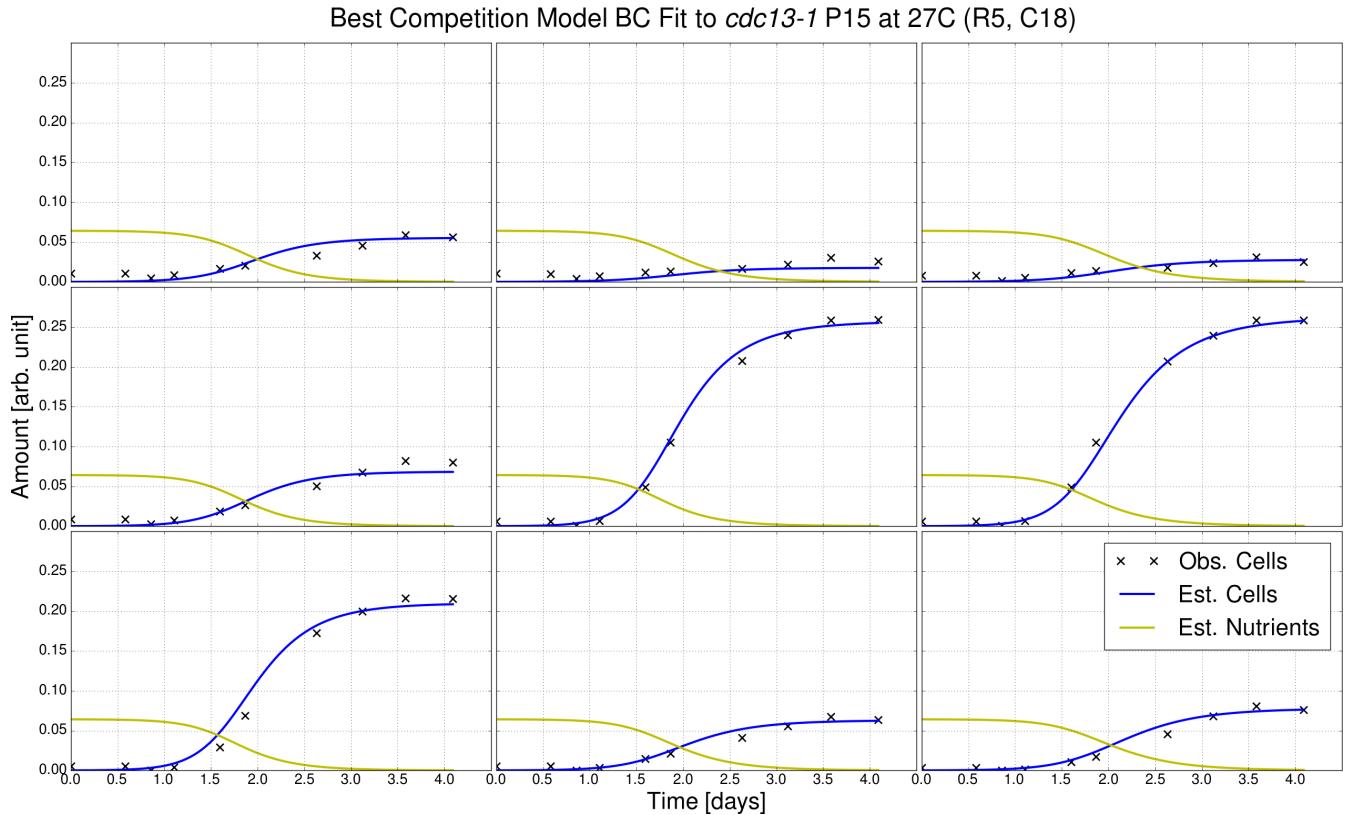
Each b-guess value is used to guess a complete set of b parameters for every culture in the plate. Each of these parameter sets is then used as an initial guess to Competition Model fitting. For the 13 b-guesses we must run 13 Competition Model fits. It would be better and more efficient to loop through the b-guesses at the culture level. Each culture still undergoes imaginary neighbour guessing with each of the 13 b-guess values, but now, for each culture, we choose just the b estimate from the best of the 13 fits. This will produce one set of b guesses which should be superior to any of the guesses attained when iterating through b-guess at the plate level. Then we only need to fit the Competition Model to 1 guess rather than 13. This will reduce

the number of scripts that need to be run in parallel, or the use of a finer grid over  $C_0$ , and should make convergence faster. However, if using a gradient method we are still likely to encounter local minima from these guesses. Instead, this improvement could be considered when developing a genetic algorithm (if initial guesses are required) or if fitting using a brute force method with a fine grid of fixed plate level parameters. We will see later that with true plate level parameters fixed we can recover good estimates for b using a gradient method. It may be possible to evolve candidates of plate level parameters, fix these, and minimise using the current gradient method.

### 3.2 Competition Model Fitting to P15

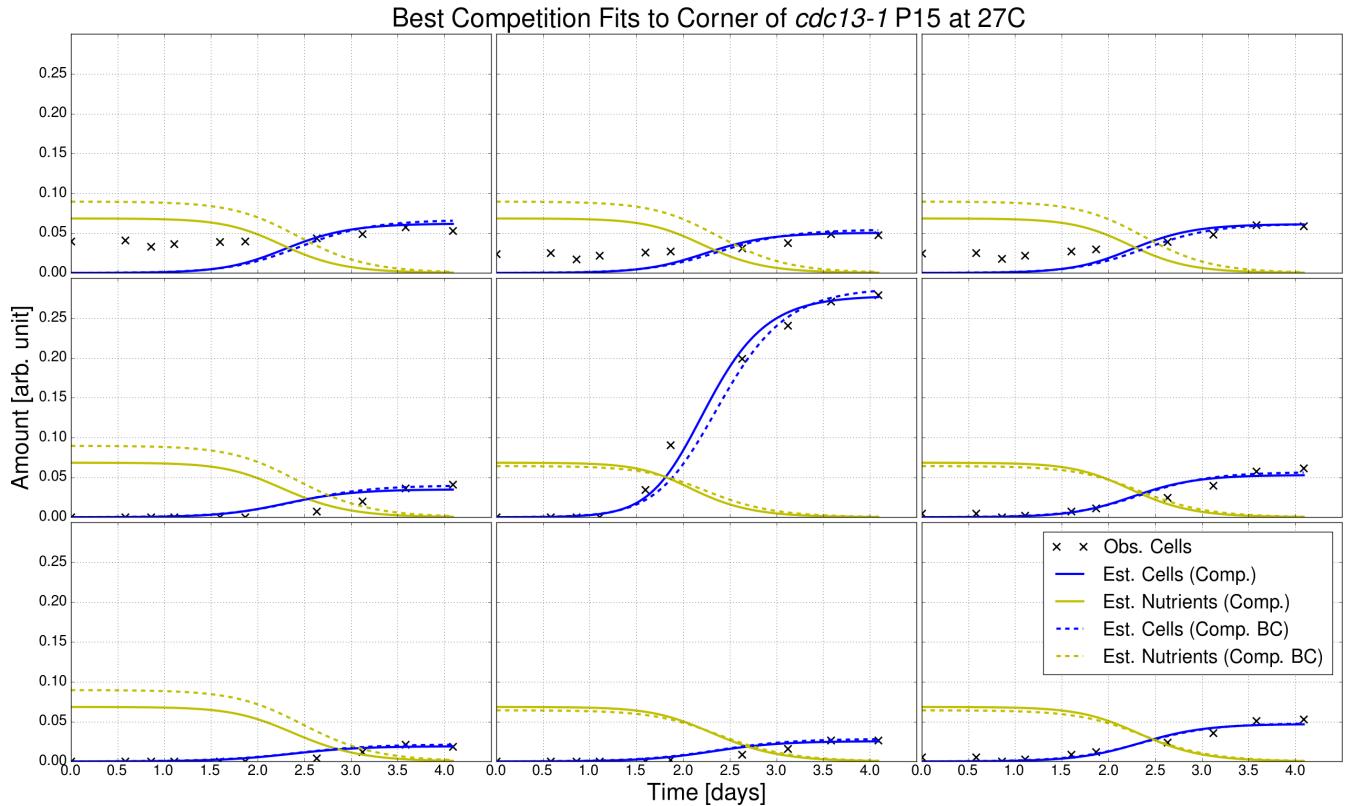


*Figure 7: (R5, C18) P15 requires legend*



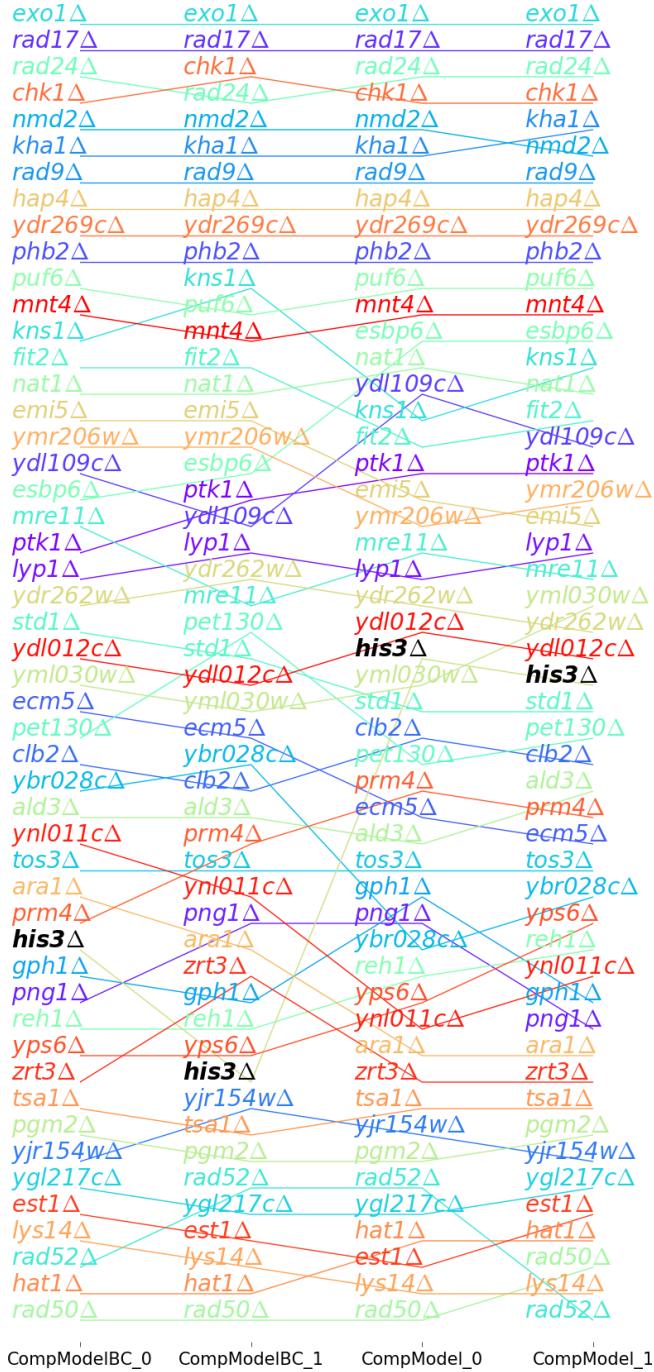
**Figure 8:** (R5, C18) P15

### **3.3 Evaluating the treatment of boundaries**



**Figure 9:** Top left corner of the Competition Model and Competition Model BC estimates for P15. NEED TO FIND OBJ FUN FOR DEPTH ONE AND TWO AND COMPARE TO AVG OBJ FUN FOR WHOLE PLATE (JUST DIVIDE BY 384). ALSO FIND COV FOR THE HIS3 EDGES.

### 3.4 Agreement of b rankings



**Figure 10:** Comparison of *b* ranking for best five Competition Model fits P15. Includes HIS3 from edge cultures (could separate or exclude these).

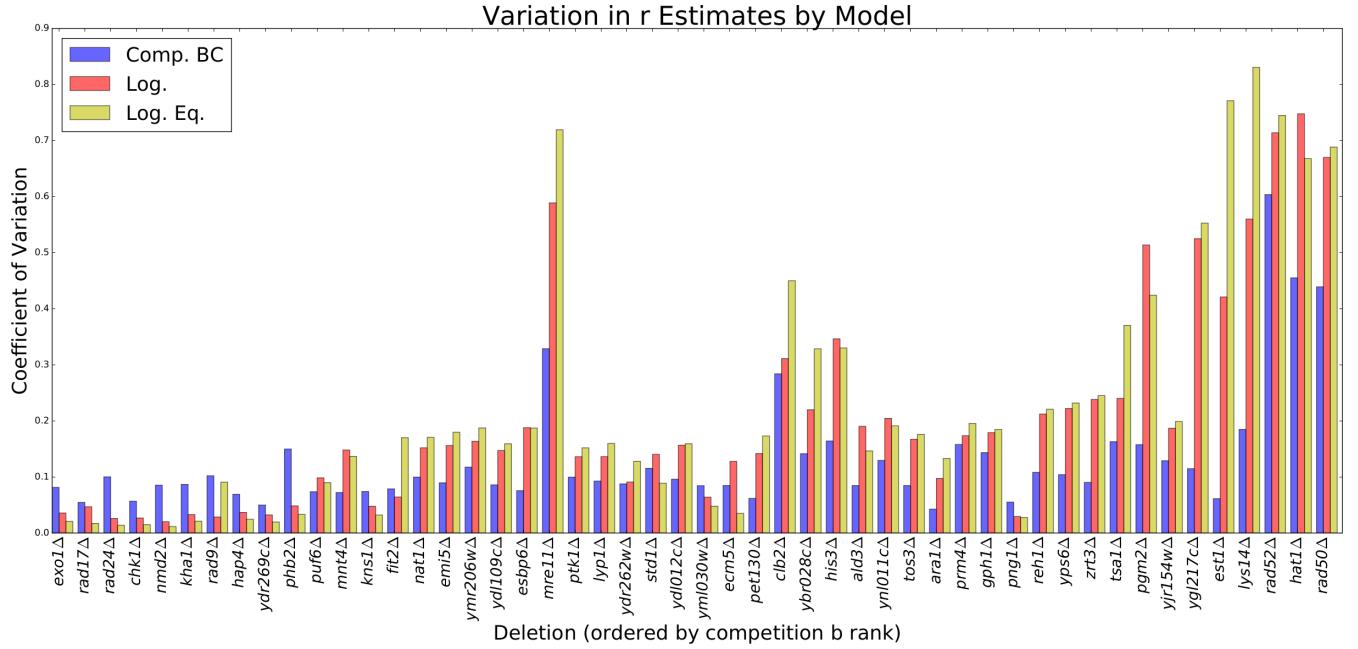
### 3.5 Comparison of fitness ranking



**Figure 11:** *r* correlations for Comp Model BC, QFA R logistic, and logistic equivalent P15. INSTEAD GO WITH MDR AND USE GENERALIZED LOGISTIC MODEL AS WELL? ADD P-VALUES.

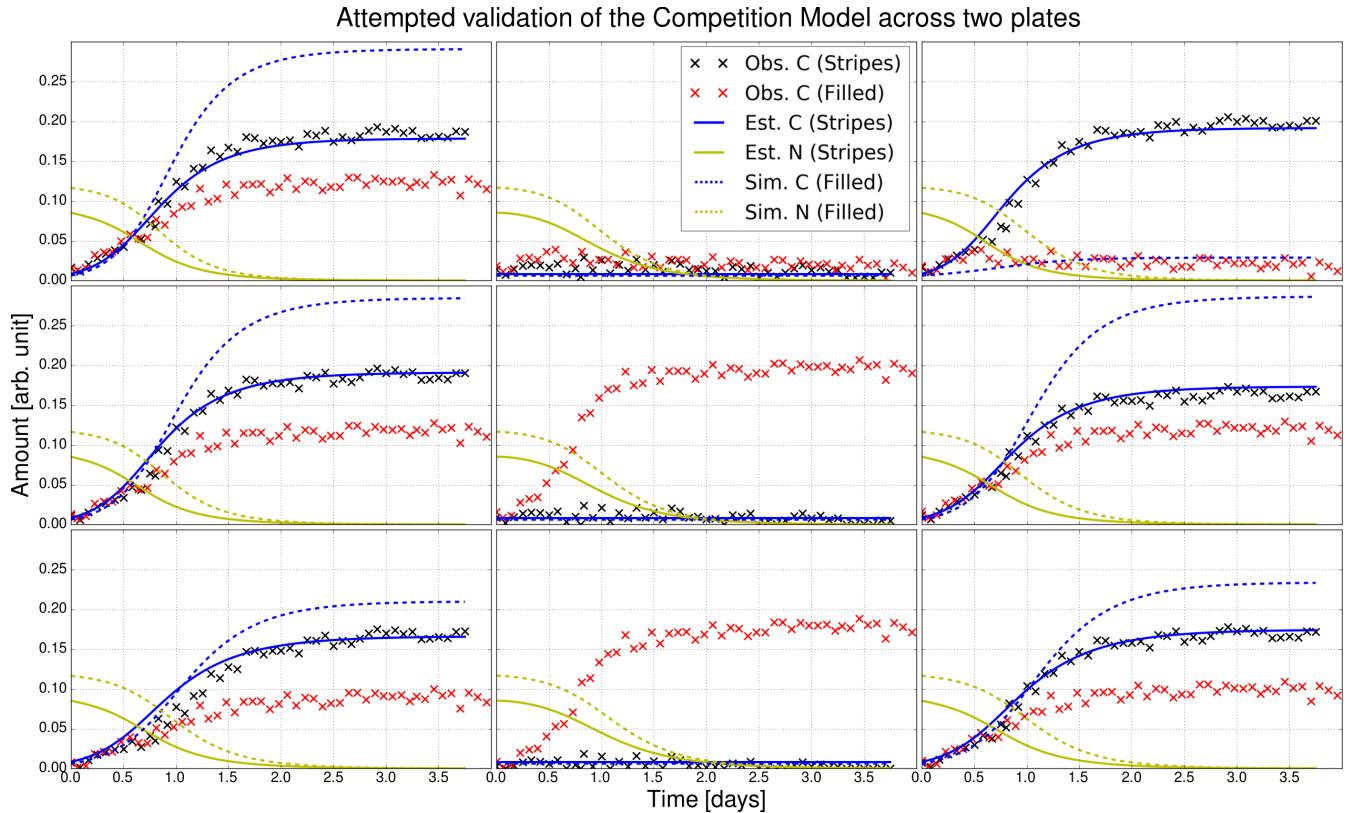
### 3.6 Comparison of Variation in Fitness Estimates

Use repeats on plate 15 (6 per deletion) to calculate coefficient of variation (COV) of estimated *r* or MDR.



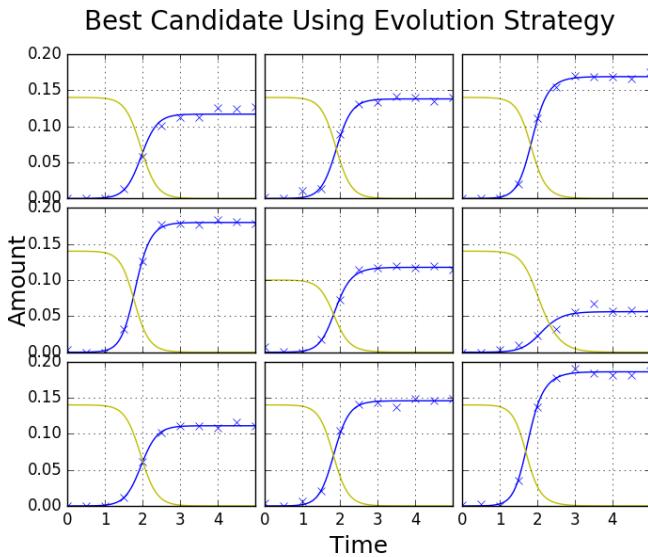
**Figure 12:** Coefficient of variation of  $r$  estimates from fits of the Competition Model BC, the QFA R Logistic Model, and the Logistic Equivalent Model to P15.

### 3.7 Cross-plate validation

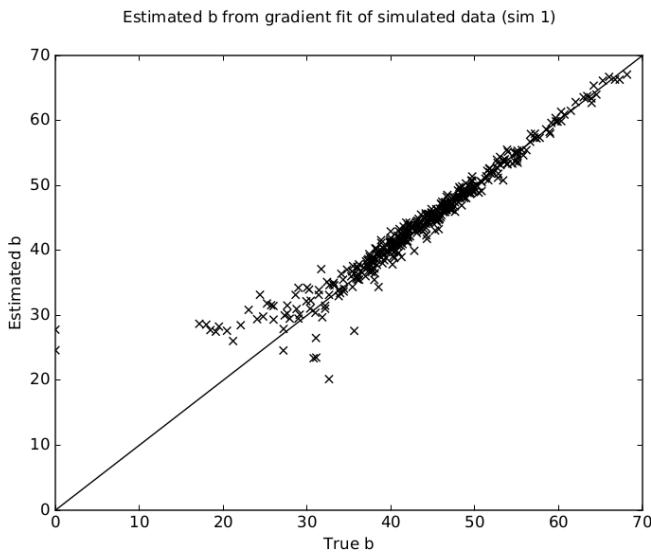


**Figure 13:** (R9, C10) top-right probable pinning mistake. Bottom-left not close to any such mistake.

### 3.8 Towards a genetic algorithm



**Figure 14:** Genetic algorithm fit to a 3x3 simulation. MIGHT TAKE A LITTLE BIT OF WORK TO REPRODUCE AND COULD USE PARAMETERS FROM THE BEST P15 FIT RATHER THAN JUST PICKING/RANDOMIZING. NEED TO CHECK THAT PLATE LEVEL PARAMETERS WERE ALSO EVOLVED.



**Figure 15:** Recovery of true  $b$  values from a gradient method with fixed plate level parameters. I simulated timecourses from the best five (which model? all BC?) fits to p15, fixed the true plate level parameters, and used a gradient method to recover  $b$ . This plot shows the worst case from the five sets of values.

## 4 DISCUSSION

### 4.1 Subsection

### REFERENCES

Addinall, S.G. *et al.* (2008) A genomewide suppressor and enhancer analysis of cdc13-1 reveals varied cellular processes influencing telomere capping in *saccharomyces cerevisiae*. *Genetics*, **180**, 4, 2251–2266.

Addinall, S.G. *et al.* (2011) Quantitative fitness analysis shows that nmd proteins and many other protein complexes suppress or enhance distinct telomere cap defects. *PLoS Genet*, **7**, 4, 1–16.

Heydari, J. *et al.* (2016) Bayesian hierarchical modelling for inferring genetic interactions in yeast. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **65**, 3, 367–393.

Lawless, C. *et al.* (2010) Colonyzer: automated quantification of micro-organism growth characteristics on solid agar. *BMC Bioinformatics*, **11**, 1, 1–12.

Palková, Z. *et al.* (1997) Ammonia mediates communication between yeast colonies. *Nature*, **390**, 6659, 532–536.