new-TF-modeling math

Albert Zhang

August 2024

1 Introduction

EcoMAC gives a log2(expression), in arbitrary units, e_i for each of 4198 mRNA genes i in a certain condition. Due to their quantile-normalization method, the "total expression" ($\sum_i 2^{e_i} = K$) is the same for all conditions. We will derive the expected fraction-of-total-mRNA-gene-expression by

$$m_i = \frac{2^{e_i}}{K} \tag{1}$$

for one-peak genes, where e_i is defined to be the median of the peak. For two-peak genes, we will similarly let e_i^{low}, e_i^{high} be the medians of the lower and higher peaks, respectively, and let

$$m_i^{low,high} = \frac{2^{e_i^{low,high}}}{K}. \tag{2}$$

In the WCM, we use three closely-related sets of values that describe gene expression. The first is *expression* (*exp*), which represents the fraction of counts of all transcribed TUs (mRNAs, rRNAs, tRNAs, etc.) that a particular TU takes up. Thus $\sum_{i forall TUs} exp_i = 1$. The m_i from EcoMAC is thus meant to correspond to $\frac{exp_i}{\sum_{i forall mRNA TUs} exp_i}$.

The second is *synthesisprobabilities* (*synthprob*), which represents the probability fraction of all TU transcription that a certain TU is transcribed at any given moment (mouthful), and so $\sum_{i forall TUs} synthprob_i = 1$ as well. During simulations, synthprobs are calculated at each timestep (described later), and RNAPs are distributed to TUs according to them. Assuming steady-state of mRNAs (mainly during the parca), *synthprob* and *exp* are thus related by:

$$synthprob_i = \frac{exp_i * (dilution + deg_i)}{\sum exp_i * (dilution + deg_i)} \tag{3}$$

$$\implies exp_i = \frac{synthprob_i/(dilution + deg_i)}{\sum synthprob_i/(dilution + deg_i)} \tag{4}$$

$$synthprob_i \propto exp_i * (dilution + deg_i) \tag{5}$$

where *dilution* and deg_i are the growth-rate-determined dilution rate, and degradation rate, of TU_i, respectively.

The third is $affinities$ (aff), which is the only type of parameter the simulation itself actually uses. This is meant to represent the "affinity" that a certain RNAP binds to a certain TU to transcribe it. They are not directly proportional to *synthprob*, because a given TU may be present in multiple copies. The relation between *synthprob* and aff is: at any given time-point in a simulation, or for any given modeled condition while fitting parameters,

$$synthprob_i = \frac{aff_i * copy_i}{\sum_i (aff_i * copy_i)} \tag{6}$$

$$\implies aff_i = \frac{synthprob_i * \sum_i (aff_i * copy_i)}{copy_i} \tag{7}$$

$$aff_i \propto \frac{synthprob_i}{copy_i} \tag{8}$$

where $copy_i$ is either the copy number of a given TU_i at a given time-point in a simulation, or the average copy-number of TU_i for a given condition (used mainly while fitting parameters).

Affinities are in fact not fully determined by these equations, since if you multiply every affinity by some constant factor, the equation still holds true (they are homogeneous linear equations, I think). So we must somehow normalize affinities in each calculation. The observation of one-peak genes (genes with approximately constant expression-fraction-of-mRNAs in different conditions) suggests that these genes may have constant affinity. This means we cannot simply normalize affinities by having them sum to 1 in each case, like we do for *exp* and *synthprob*: then, if a cell has 1 copy of each gene in a certain condition and 2 copies of each gene in another condition, with identical expression values for each gene, this gene would have about half the affinity in the latter condition. However, intuitively we would expect a per-gene-property like affinity to not change, and our assumption that constant affinity roughly is equivalent constant expression also would not hold in this case. That is to say, we expect affinities to roughly hold constant for many genes; when the total number of genes increases during different conditions, we expect the "total affinity" of all gene objects to increase, and the reason that constant affinity means roughly constant expression, is approximately that the copy number of a particular gene roughly rises in proportion to the "overall" rise in copy number of the genome.

To put this rigorously, if for a certain gene, constant affinity is to imply constant mRNA-fraction-of-expression between different conditions during the parca, we've that (note that in the denominator we are summing across all

mRNAs, instead of all TUs as previously):

$$exp - of - mRNA - fraction_i = \frac{exp_i}{\sum_{jmRNAs} exp_j} \tag{9}$$

$$= \frac{synthprob_i/(dilution + deg_i)}{\sum_{jmRNAs} synthprob_j/(dilution + deg_j)} \tag{10}$$

$$= \frac{aff_i \frac{copy_i}{\sum_{kallTUs}(aff_k copy_k)}/(dilution + deg_i)}{\sum_{jmRNAs} aff_j \frac{copy_j}{\sum_{kallTUs}(aff_k copy_k)}/(dilution + deg_j)} \tag{11}$$

$$\implies C = \frac{\frac{copy_i}{\sum_{kallTUs}(aff_k copy_k)}/(dilution + deg_k)}{\sum_{jmRNAs} aff_j \frac{copy_j}{\sum_{kallTUs}(aff_k copy_k)}/(dilution + deg_j)} \tag{12}$$

$$= \frac{copy_i/(dilution + deg_i)}{\sum_{jmRNAs} aff_j copy_j/(dilution + deg_j)} \tag{13}$$

We allow for imprecision due to $dilution$ or deg_i ($dilution$ is expect to change the same for all genes, and deg_i is expected to be approximately constant for a given gene across different conditions), because while including them would make constant affinity more accurately match constant expression, from the biochemical standpoint it makes more sense that aff can be directly calculated from $synthprob$ without regard to loss rates of RNAs (THINK). This means

$$C = \frac{copy_i}{\sum_{jmRNAs} aff_j copy_j} \tag{14}$$

is supposed to hold true for all conditions (that is, C is constant) for a given gene j.

So, we seek to incorporate this assumption while calculating affinities from exp or $synthprob$, for all genes. If C is known, we have the equation:

$$aff_i = \frac{synthprob_i}{C \sum_{jmRNAs} synthprob_j} \tag{15}$$

and the relation to exp likewise follows. C evidently cannot always be constant for all genes, since genes have different average copy numbers in a given condition and the denominator is constant. The best we can say, is that gene with constant affinity whose $copy_i$ roughly follows $\sum_{jmRNAs} aff_j copy_j$ are expected to have roughly constant expression. Thus, instead of presupposing that C is constant, we must instead somehow define our aff_i so that C for different conditions is expected to be roughly constant, despite widely changing $copy_i$'s. We can transform our assumption that constant affinity implies constant expression, which cannot hold for all genes, by adding in our expectation of for which genes it holds true: we expect that genes located roughly in the middle of the chromosome, and thus which increase in copy number roughly according to the "average" mRNA gene in the chromosome, satisfies this assumption. That is to say, $\frac{copy_i}{\sum_{jmRNAs} copy_j}$ constant $\implies$ $\frac{copy_i}{\sum_{jmRNAs} aff_j copy_j}$ constant. This means that $\frac{\sum_{jmRNAs} aff_j copy_j}{\sum_{jmRNAs} copy_j}$ is constant, or in other words the "average" affinity of

an mRNA gene is roughly constant in all conditions, which seems a reasonable assumption. Note that, the "average" affinity of an rRNA gene is expected to change due to this, since the rRNA fraction of total RNAs changes quite a bit in different conditions. We admit it is also equivalently possible (while holding our constant-affinity-implies-constant-expression assumption) that the average affinities of mRNAs as a whole to change significantly in different conditions if this change to be applied the same to all mRNAs (e.g. by ppGpp), but since it is more widely thought that ppGpps regulate rRNAs and not the majority of mRNAs, that many mRNAs don't have annotated regulation, and also since it makes for simpler modeling to have constant affinity for one-peak EcoMAC genes, we will take the former assumption.

Now, we finally seek to define aff based on *synthprob*. As noted before, we cannot define aff_i for a given condition by using $\frac{copy_i}{\sum_{jmRNAs} aff_j copy_j}$ since that would result in homogeneous linear equations (we could also previously define certain aff_j based on e.g. minimal media condition, but it seems nicer to have a self-contained definition for affinity in a certain condition), and we cannot simply define a C since this is not expected to hold constant for all genes. However, letting $M = \frac{\sum_{jmRNAs} aff_j copy_j}{\sum_{jmRNAs} copy_j}$ which we indeed will assume to hold constant across all conditions, we can now say that:

$$aff_i = M \frac{synthprob_i \sum_{jmRNAs} copy_j}{copy_i \sum_{jmRNAs} synthprob_j}. \tag{16}$$

Since the scale of aff_i is arbitrary anyways, we can let $M = 1$, and so our final definition of aff_i is:

$$aff_i = \frac{synthprob_i}{\sum_{jmRNAs} synthprob_j} \frac{\sum_{jmRNAs} copy_j}{copy_i} \tag{17}$$

$$= \frac{synthprob_i}{copy_i} \frac{\sum_{jmRNAs} copy_j}{\sum_{jmRNAs} synthprob_j}. \tag{18}$$

The second expression offers a simple interpretation of this definition of aff_i: it is simply the per-copy synthesis probability of a certain gene in a given condition, normalized (divided) by the average synthesis-probability-per-gene-copy of all mRNAs in a given condition. In this way, if for example between rich and minimal conditions, the genome size increases due to duplication and so the copy numbers of all mRNAs genes rises an average of 2x, and rRNAs also take up more of the genome so overall mRNA synthesis probabilities drop as well, a constant affinity for a given mRNA will result in a synthesis probability fraction of mRNA synthesis probability that is the same in minimal media condition (and the backward logic from constant synthesis probability fraction of mRNA to constant affinity holds too).

To calculate *synthprob* and *exp* from aff, we can simply normalize $aff \times$ *copy* to obtain *synthprob*, and normalize $synthprob \times (dilution + deg)$ to obtain *exp*.

2 Parca overview

Now, for the model, our end goal is to derive two sets of parameters: 1. A function that calculates the affinities of each TU (mRNA or not) given the ppGpp level, $synth-aff-from-ppgpp(ppgpp)$. This function will linearly interpolate between two sets of parameters aff^{free} and aff^{ppGpp} given the fraction of RNAPs that are in ppGpp-bound-form f_ppgpp: $aff_i = aff_i^{free}(1 - f_{ppGpp}) + aff_i^{ppGpp} f_{ppGpp}$. We will touch on how f_{ppGpp} is calculated and parameterized later. These are mostly modifications from previous work with growth-rate control in the WCM. 2. A set of parameters that defines how TFs change expression when bound to their binding sites. For now, with only two-peak regulations included, this will be a single value for each TF-regulated gene pair: aff_i^{bound} which is the affinity of TU_i when the TF is bound to the corresponding binding site for this gene. Note that $aff_i^{unbound}$ is simply given by the $synth-aff-from-ppgpp$ function. For one-peak genes, $synth-aff-from-ppgpp$ also returns its affinity.

For now, we assume that one-peak genes and two-peak genes are not regulated by ppGpp. This means that $aff_i^{ppGpp} = aff_i^{free}$ for one-peak and two-peak genes. For the future, there are many choices for how to combine ppGpp-regulation and TF regulation, which could involve modifying either sets of parameters.

The raw data that we have to work with is: 1. RNAseq values for minimal conditions, which gives relative abundances (proportional to our *exp* value) for mRNAs, 2. Mass fraction of the cell that rRNA, tRNA, and mRNA occupy under minimal growth conditions, and some other growth-rate dependent parameters like rna-mass-fraction under different growth rates, which are used only to parameterize ppGpp parameters like f_{ppgpp}, 3. Expression-fraction-of-mRNA values for one-peak genes (a single value applying to all conditions), or two values (bound and unbound) for two-peak genes, with one of the two values applying to each condition, 4. ppGpp fold-change data (a fold-change for each gene between rich condition and rich condition with overproduced ppGpp, where we assume all RNAPs are ppGpp-bound) and ppGpp-regulated-genes as annotated by EcoCyc.

There is a fifth set of constraints-physiological constraints, which constrain the expression of RNAPs and r-proteins given the overall RNA expression. Briefly, for any supposed final expression of all RNAs in a certain condition, the doubling time can be used to obtain the expected RNA-mass fraction, which can be used to obtain the counts of each RNA in an average cell, which can then be used to determine how much active RNAP is needed on average given the elongation rates for mRNA or stable RNA and the active fraction of RNAP. A similar case for r-proteins: the expression of RNAs can be used to derive counts of proteins given the protein mass fraction for the condition's doubling time, which can be used to determine how much ribosomes are needed, and thus r-proteins. NOTE: this does not apply to rRNAs, however, so rRNAs are not necessarily being produced at the rate that would match these physiological considerations, which might be part of the reason for initialization effects?

The basic idea of our process to obtain the two sets of parameters is then:

1. We first fit key ppGpp parameters (including its K_M and the fold-change between aff^{free}/aff^{ppgpp} for stable RNAs) by using growth-rate dependent parameters. This will help set the ultimate aff^{free} and aff^{ppgpp} for stable RNAs, and the other set parameters will be needed to calculate other affinities. The rest of ppGpp fold-changes will be drawn from raw data.

2. We will fit minimal media-condition expression by these constraints: a) the expressions of rRNA, tRNA, and mRNA must result in the right rRNA, tRNA, and mRNA mass-fractions (given by raw data), b) for rRNAs, expression is evenly distributed by gene dosage, and according to ratios from raw-data for tRNAs, c) for mRNAs, i) the expression of one-peak and two-peak genes must be according to EcoMAC data (MAYBE), ii) the expression of r-proteins and RNAPs must satisfy the physiological constraints given overall expression, and iii) the other mRNAs will follow the ratios given by basal RNAseq.

3. This minimal-media expression will be converted to *synthprob* and aff as described above, and this will parameterize an initial version of aff^{free} and aff^{ppgpp} for all TUs. These are the final aff's other than for polymerizing genes (r-proteins, rRNAs, and RNAPs) which will use more conditions' physiological constraints to fit their expression, and one-peak/two-peak genes whose affinities will likewise be fit with more conditions, and adjusted for TF-binding at the end. NOTE: since one-peak/two-peak genes don't overlap with ppGpp-regulated genes at the moment, $aff^{free} = aff^{ppgpp}$ for them at this point.

4a. To parameterize one-peak and two-peak affinities, aff from aff^{free} and aff^{ppgpp} during each other modeled condition will be obtained, then converted to expression as noted above. Then, the same process as in (2) will occur to modify this *exp*: the initial mass fractions of stable RNAs must be satisfied, the one-peak and two-peak genes' expression must accord to EcoMAC data, and physiological constraints must be satisfied. This *exp* will be converted back to affinities as before, to obtain aff's for every condition. One-peak and two-peak affinities will then be fit and set with their respective aff's.

4b. With other TFs still modeled with the old scheme, old-TF fold-change effects are applied to each of the other modeled condition (before any of the iterative fitting, so the fold-changes may be perturbed a bit). Then, these affinities for each condition will be fit according to the iterative fitting procedure of the old-TF modeling to obtain basal-aff and delta-aff values (corresponding to alpha's and r's of before). This basal-aff is then modified to produce the correct one-peak and two-peak affinities.

5. aff^{free} and aff^{ppgpp} for one-peak and two-peak genes (which are equal at this point) will be reset to be equal to the one-peak aff, or two-peak $aff^{unbound}$. With old-TF modeling, aff^{free} and aff^{ppgpp} are also scaled identically to produce the basal-aff (which automatically includes matching one-peak and two-peak aff's since basal-aff has already been adjusted to match those). NOTE: more calculations should be done here when TF-regulated genes overlap with ppGpp-regulated genes.

6. The aff's obtained for polymerizing genes in each condition will be used to fit the aff^{free} and aff^{ppgpp} for these genes with least-squares, given the

same f_{ppgpp} as before in each condition.

3 Parca details

1. The work here is a modification of the original work considering exp_{ppGpp} and exp_{free}, but made to account for copy numbers and affinities. The basic assumption is that, if a certain fraction f_{ppGpp} of RNAPs are bound to ppGpp, and ppGpp-bound RNAPs are active at a rate a_{ppGpp} and free RNAPs are active at a rate a_{free}, then a certain gene has affinity given by

$$aff_i = \frac{aff_i^{free} f_{ppGpp} a_{ppGpp} + aff_i^{ppGpp}(1 - f_{ppGpp}) a_{free}}{f_{ppGpp} a_{ppGpp} + (1 - f_{ppGpp}) a_{free}}. \tag{19}$$

"Ideally", we would distribute the ppGpp-bound fraction of RNAPs according to $aff_i^{ppGpp} / \sum_{jallTUs} aff_j^{ppGpp}$ and likewise for the free fraction, which gives somewhat different results than the current method since $\sum_{jallTUs} aff_j^{ppGpp} \neq \sum_{jallTUs} aff_j^{ppGpp}$. However, this would make the definition of a single overall affinity complicated, and it seems difficult to keep the affinity of a particular gene constant across all different ppGpp concentrations. So we use the former linear affinity-extrapolating assumption.

The first goal is to fit five parameters that will dictate the amount of stable RNA made during different growth-rate conditions, which we assume completely accounts for the difference in RNA mass-fraction of the cell. For reasons described later, we also make the simplifying assumption that tRNAs constitute a constant fraction of this across all growth rates, and subtract this from the RNA mass-fraction.

The idea is that the total cellular RNAP can be split between ppGpp-bound and ppGpp-free pools, the fraction determined by $f_ppgpp = \frac{ppGpp^2}{ppGpp^2 + K_M^2}$ where K_M is the key fit parameter. These two pools have characteristic fractions of RNAP that are actively transcribing: a_{ppgpp} and a_{free}. K_M, a_{ppgpp}, and a_{free} is constrained by the raw data for overall active fraction of RNAP:

$$rnap - active - frac = f_{ppgpp} a_{ppgpp} + (1 - f_{ppgpp} a_{free}) \tag{20}$$

known for each considered growth-rate condition.

These two pools of active are allocated to rRNAs or mRNAs at a rate that is proportional to their relative affinities and copy numbers. We will assume an "average" affinity of RNAPs towards rRNAs $aff_{ppGpp,free}^{rRNA}$ and towards mRNAs, $aff_{ppGpp,free}^{mRNA}$. We make the simplifying assumption that $aff_{ppGpp}^{mRNA} = aff_{free}^{mRNA} = aff^{mRNA}$ since we have previously presumed the "average" mRNA affinity to be constant in different growth conditions. Then for a given condition, the fraction of synthesis rate (excluding tRNAs) that corresponds to rRNAs is

given by:

$$aff^{rRNA} = \frac{aff_p^{rRNA} f_p a_p + aff_{free}^{rRNA}(1 - f_{ppGpp}) a_{free}}{f_p a_p + (1 - f_p) a_{free}} \tag{21}$$

$$synthprob^{rRNA} = \frac{aff^{rRNA} copy^{rRNA}}{aff^{rRNA} copy^{rRNA} + aff^{mRNA} copy^{mRNA}} \tag{22}$$

where $copy^{rRNA}$ is the average summed copy number of rRNAs in this condition and $copy^{mRNA}$ is the average summed copy number of mRNAs in this condition. In order to avoid having to define separate aff^{rRNA} and aff^{tRNA} since $copy^{rRNA}$ and $copy^{tRNA}$ may not rise completely in tandem (and so we cannot simply define an average aff^{stable} that is constant in all conditions), we instead make the assumption (as before) that tRNAs constitute a constant relatively small (0.14) fraction of total RNA and exclude it.

We define $y_{ppGpp} = aff_{ppGpp}^{rRNA}/aff^{mRNA}$ and $y_{free} = aff_{free}^{rRNA}/aff^{mRNA}$ so the expressions simplify to only having these two unknown constants:

$$k^{rRNA} = \frac{y_p f_p a_p + y_{free}(1 - f_p) a_{free}}{f_p a_p + (1 - f_p) a_{free}} \tag{23}$$

$$synthprob^{rRNA} = \frac{k^{rRNA} copy^{rRNA}}{k^{rRNA} copy^{rRNA} + copy^{mRNA}} \tag{24}$$

We can do this simplification because, ultimately, we care only about the ratio $aff_{ppGpp}^{rRNA}/aff_{free}^{rRNA} = y_{ppGpp}/y_{free}$.

Now, we can produce the second constraint by converting the different rRNA mass-fractions in different conditions to concentration C_{rRNA}, and obtain the total concentration of RNAPs in different conditions by C_{RNAP}. The loss rate of rRNAs is naturally given by $C_{rRNA} gr$ where gr is the growth rate (dilution rate). The production rate is equal to

$$C_{RNAP}(f_p a_p + (1 - f_p) a_{free}) synthprob^{rRNA} \frac{e}{l} \tag{25}$$

where f, a, and k are defined previously, e is the (constant across growth conditions) RNAP transcript elongation rate on rRNAs, and l is the average rRNA transcript length. Setting this equal to the production rate gives:

$$C_{rRNA} gr = C_{RNAP}(f_p a_p + (1 - f_p) a_{free}) \frac{e}{l} \frac{k^{rRNA} copy^{rRNA}}{k^{rRNA} copy^{rRNA} + copy^{mRNA}} \tag{26}$$

Here, $C_{rRNA}, gr, C_{RNAP}, ppGpp, copy^{rRNA}$ and $copy^{mRNA}$ are all known growth-rate dependent parameters obtained or derived from raw data, and the other variables $K_M, a_{ppGpp}, a_{free}, y_{ppGpp}$ and y_{free} are all unknown growth rate-independent constants that will be fit.

A gradient descent least-squares fitting method will fit these five unknowns for all (five) considered growth rate-conditions, according to this equation and

the previous one about overall active fraction of RNAPs, with a hyperparameter λ the relative weights of the two constraints. The end result is: f_{ppGpp} that can be calculated for any *ppGpp* concentration with K_M, active fraction of RNAPs that can be calculated for any *ppGpp* concentration with K_M and $a_{ppgpp,free}$, and the fold-change in affinity for rRNAs (and assumed to be the same for tRNAs) between ppGpp-bound and unbound conditions, $y_{ppGpp}/y_{free} = aff_{ppGpp}/aff_{free} = fc^{stableRNA}$ for any stable RNA.

Will validate against overall fraction of RNAP synthesizing stable RNA, from growth-rate-dependent parameters.

2. All three constraints will be put in an iterative solver. Previously, the iterative solver only considered mass fractions and physiological constraints, whereas TF-related constraints were applied previously to the solver. *exp* will be modified to fit each constraint one by one: first scaling mRNA, tRNA, and rRNA expressions to match the mass fractions, then setting *exp* for one-peak and two-peak genes to be the desired expressions (which is actually a fraction of total mRNA expression, so the exact value will change with each iteration) and normalizing afterward, and finally setting r-protein and RNAPs to the desired expressions and normalizing afterward. After each iteration, they may exactly match none of the three, but when the solver converges, the solution will be a good match for all.

Since EcoMAC gives expressions for genes (cistrons), the expressions must be converted to those for TUs first. For now, we will make the simplifying assumption that if a cistron is a one-peak gene, then the entire operon follows the one-peak assumption; and if it's a two-peak gene, all TUs in the operon are also two-peak regulated by the TF (TODO: change this, this probably is not true generally). As of now, for one-peak genes, we will first get a list of all cistrons contained within the operon for this one-peak gene. We'll then fit TU mRNA-fraction-expressions with NNLS to cistron mRNA-fraction-expressions calculated by: the one-peak gene(s) in the operon will have mRNA-fraction-expressions from EcoMAC, and the other gene(s) in the operon will have it from the RNAseq raw data. Fitting NNLS gives values for TUs in the operon. This value will be normalized by the total raw sum of fitted-values for TUs for all mRNA operons (which will be saved in sim-data.process.transcription) to obtain TU-level expression-fraction-of-mRNAs. These values will be stored as the *exp* constraint above. For two-peak genes, a similar process occurs for all genes in an operon: for the peak corresponding to minimal media, the same process will occur as for one-peak genes to obtain TU-level expression fraction-of-mRNAs for the TUs within the operons containing the genes. NOTE: right now, for simplicity, this could theoretically fix the expression of TUs that don't contain the gene/are not regulated by the TF (for a two-peak gene). NOTE/TODO: probably wan't to change this.

For the other peak, for each operon containing the genes, we'll solve the NNLS problem for these TUs to obtain the cistron-fraction-mRNA-expression that EcoMAC gives, while setting no constraints on the expression of genes other than the two-peak genes that might be contained within the operon. These TU-level values are converted into TU-level expression-fraction-of-mRNAs as before,

and stored. For other TUs in the operon that don't contain the two-peak genes, we'll simply store the same expression as they had in the minimal media peak.

NOTE/TODO: this will be wrong if there are other TUs containing the gene that the TF can't regulate. So fix it for these cases.

NOTE: might want to come back to this if complicated operons start getting involved. Also, would have problems if two-peak and one-peak genes are in the same operon, so need to consider that carefully. TODO: consider these cases?

NOTE/TODO: this part probably needs changing when more complex operons are included!

3. For all genes, $exp^{minimal}$ is converted to *synthprob* and then aff by:

$$synthprob_i^{minimal} = \frac{exp_i * (dilution + deg_i)}{\sum exp_i * (dilution + deg_i)} \tag{27}$$

$$aff_i^{minimal} = \frac{synthprob_i}{\sum_{jmRNAs} synthprob_j} \frac{\sum_{jmRNAs} copy_j}{copy_i} \tag{28}$$

where *copy* is determined by minimal media growth rate. The goal now is to set an initial aff_{ppgpp} and aff_{free} that matches minimal media conditions given *ppgpp* during this condition, and also matches the fold-changes in *exp* from raw data. One constraint is that for genes unregulated by ppGpp, $aff_{ppgpp} = aff_{free}$, that is their affinities will not change with difference *ppGpp* levels.

Since the raw data fold changes are all in cistron-level, we will first convert $aff^{minimal}$ to cistron-level by saying: $aff^{minimalcistron} = aff^{minimal} cistron - tu - matrix$ (the rationale for this provided near the end of this section, the same as the explaining why we perform NNLS at the end). We'll then calculate cistron-level $aff_{cistron}^{ppgpp,free}$, and convert back to TU-level by performing NNLS on these two sets of affinities to obtain $aff^{ppgpp,free}$. This process ensures that $aff_{ppgpp} = aff_{free}$ for ppgpp-unregulated genes, that raw fold changes are applied at the cistron-level, and that the overall scale of mRNA affinities doesn't change very much in accordance with the affinity definition (since we are simply multiplying by the cistron-tu-matrix, applying fold-changes to a fraction of their cistrons, and "undoing" the cistron-tu-matrix by applying NNLS) (it will change slightly, but that is to be expected with the $aff_{ppgpp} = aff_{free}$ assumption for ppgpp-unregulated genes). We will need a final step to rescale $aff_{ppgpp,free}$ by the same amount so that we correct for any inaccuracies introduced during NNLS, and re-obtain $aff^{minimal}$ for minimal condition-level of *ppgpp*. TODO: write out where to do this final step

We will first calculate a fold-change $fc_i = aff_i^{ppGpp} / aff_i^{free}$ for all genes ($fc_i = fc^{stableRNA}$ for stable RNAs). Given this fold-change, we can then calculate aff^{ppGpp} and aff^{free} because we have that:

$$aff_i^{minimal} = \frac{aff_i^{ppgpp} f_{ppGpp} a_{ppGpp} + aff_i^{free}(1 - f_{ppGpp} a_{free}}{f_{ppGpp} a_{ppGpp} + (1 - f_{ppGpp}) a_{free}} \tag{29}$$

$$= aff^{free} \frac{fc_i f_{ppGpp} a_{ppGpp} + (1 - f_{ppGpp}) a_{free}}{f_{ppGpp} a_{ppGpp} + (1 - f_{ppGpp}) a_{free}} \tag{30}$$

and so aff^{free} can be directly solved from $aff^{minimal}$.

To calculate fc for mRNAs, we have raw data that for most mRNAs, gives exp^{ppGpp}/exp^{rich} where exp^{ppGpp} is from rapidly (10min) after cells growing on rich media have ppGpp overproduced within them, and exp^{rich} is just from the rich media. We'll assume these the copy numbers of mRNAs have not changed between the two conditions, as well as dilution and mRNA degradation rates. So,

$$\frac{exp_i^{ppGpp}}{exp_i^{rich}} = \frac{\frac{synthprob_i^{ppGpp}/(dilution+deg_i)}{\sum_{jallcistrons} synthprob_j^{ppGpp}/(dilution+deg_j)}}{\frac{synthprob_i^{rich}/(dilution+deg_i)}{\sum_{jallTUs} synthprob_j^{rich}/(dilution+deg_j)}} \tag{31}$$

$$= \frac{\frac{synthprob_i^{ppGpp}}{\sum_{jallTUs} synthprob_j^{ppGpp}/(dilution+deg_j)}}{\frac{synthprob_i^{rich}}{\sum_{jallcistrons} synthprob_j^{rich}/(dilution+deg_j)}} \tag{32}$$

For simplicity, we'll assume the sums in numerator and denominator don't change either. We'll also define $b_{ppGpp} = f_{ppGpp}a_{ppGpp}$ and $b_{free} = f_{free}a_{free}$ to ease notation:

$$\frac{exp_i^{ppGpp}}{exp_i^{rich}} = \frac{synthprob_i^{ppGpp}}{synthprob_i^{rich}} \tag{33}$$

$$= \frac{x_i^{ppGpp}}{(b_{ppGpp}x_i^{ppGpp} + b_{free}x_i^{free})/(b_{ppGpp} + b_{free})} \tag{34}$$

$$= \frac{b_{ppGpp} + b_{free}}{b_{ppGpp} + b_{free}x_i^{free}/x_i^{ppGpp}} \tag{35}$$

$$x^{ppGpp} = \frac{aff_i^{ppGpp}copy_i^{rich}}{\sum_{jallcistrons} aff_j^{ppGpp}copy_j^{rich}} \tag{36}$$

$$x^{free} = \frac{aff_i^{free}copy_i^{rich}}{\sum_{jallcistrons} aff_j^{free}copy_j^{rich}} \tag{37}$$

$$\implies \frac{x_i^{ppGpp}}{x_i^{free}} = \frac{aff_i^{ppGpp}}{aff_i^{free}} \frac{\sum_{jallcistrons} aff_j^{free}copy_j^{rich}}{\sum_{jallcistrons} aff_j^{ppGpp}copy_j^{rich}} \tag{38}$$

$$\approx fc_i \frac{aff_{mRNA}copy_{mRNAs}^{rich} + aff_{stable}^{free}copy_{stableRNAs}^{rich}}{aff_{mRNA}copy_{mRNAs}^{rich} + aff_{stable}^{ppGpp}copy_{stableRNAs}^{rich}} \tag{39}$$

We will assume that the "average" mRNA affinity aff_{mRNA} doesn't change between minimal media and rich conditions, so

$$aff_{mRNA}^{minimal} = \frac{\sum_{jmRNAs} aff_j^{minimal}copy_j^{minimal}}{\sum_{jmRNAs} copy_j^{minimal}}. \tag{40}$$

Furthermore, since we already have fc^{stable} as well as $aff_i^{minimal}$ for stable RNAs, we can calculate $aff_i^{free,ppGpp}$ for stable RNAs beforehand. This leaves

us with:

$$\frac{x_i^{ppGpp}}{x_i^{free}} \approx fc_i \frac{aff_{mRNA}^{minimal} copy_{mRNAs}^{rich} + \sum_{jstableRNAs}(aff_j^{free} copy_j^{rich}}{aff_{mRNA}^{minimal} copy_{mRNAs}^{rich} + \sum_{jstableRNAs}(aff_j^{ppGpp} copy_j^{rich})} \quad (41)$$

so in fact we can calculate fc_i for each mRNA with an expression fold-change. For mRNAs without an expression fold-change, we will take mostly the same process as before (we'll set positive fc's to the average positive fc, and negative fc's to the average negative fc, and for negative aff_{free} we'll set it equal to 0 TODO: check if there are any of these negative cases?).

So far, this will calculate the cistron-level aff_{ppgpp} and aff_{free}. To convert to TU level, note that:

$$synthprob_{icistron} = \frac{\sum_{jTUsw/cistroni} synthprob_j}{\sum_{kcistrons} \sum_{jTUsw/cistronk} synthprob_j} \quad (42)$$

$$= \frac{\sum_{jTUsw/cistroni} aff_{jTU} copy_{jTU}}{\sum_{kcistrons} \sum_{jTUsw/cistronk} aff_{jTU} copy_{jTU}} \quad (43)$$

$$= \frac{copy_{icistron} \sum_{jTUsw/cistroni} aff_{jTU}}{\sum_{kcistrons} copy_{kcistron} \sum_{jTUsw/cistronk}} \quad (44)$$

where the last step we assume $copy_{jTUw/cistroni} = copy_{icistron}$. Thus, $aff_{cistroni} = \sum_{jTUsw/cistroni} aff_{jTU}$, and so we can solve NNLS on the cistron affinities aff_{ppgpp} and aff_{free} to obtain the TU-level aff_{ppgpp} and aff_{free}.

Finally, we scale aff^{ppgpp} and aff^{free} proportionally so that at minimal condition *ppgpp*, the calculated synthesis affinities matches the minimal affinity for each gene, while maintaining the fold changes. This accounts for any inaccuracies introduced from NNLS, etc. TODO: check does any gene change drastically here?

4. The goal here is to fit ideal expression values for other conditions given the various constraints, and to use this to fit affinities for TF-regulated genes and ppgpp-affinities for polymerizing genes.

For any condition *condition*, it has an $f_{ppGpp}^{condition}$ and we can obtain $aff^{conditioninit}$ by the equation listed in 1. These are then converted in $exp^{conditioninit}$ by the definitions above (reverse of process at beginning of 3.). This will play the role of the RNAseq expression measurements during fitting for 1: while TF-regulated genes and polymerizing genes will be fit separately, the rest of the TUs will have expression according to this ratio. We will try to maintain the mass ratio of rRNAs/tRNAs/mRNAs, but since no direct data is available for conditions other than minimal, we'll simply set the mass ratios as the one calculated from $aff^{conditioninit}$, in order to not have expression stray from it.

By the same process as 1., expression will be fit to maintain mass ratios, keep one-peak and two-peak expression at their designated mRNA expression fractions, and match physiological constraints, while the other genes' expressions are scaled down proportionally. This produces the final $exp^{condition}$, which will be converted back to $aff^{condition}$ (NOTE: there is a little bit of inaccuracy

here, since here most unregulated genes' affinity probably decreases slightly, but during simulations they won't change, and so the overall expression of TF-regulated genes and polymerizing genes will be slightly lower than expected. However, this bit of inaccuracy is accepted in order to keep the aff definition simple and self-contained, without having to pick out particular sets of genes as "affinity anchors").

5. We then fit one-peak and two-peak affinities. For one-peak genes, $aff_i = avg(aff_i^{allcondition})$. For two-peak genes, the raw data will include lists of which conditions correspond to an "unbound" case, and which are a "bound" case, and so $aff_i^{unbound} = avg(aff_i^{unboundconditions})$, $aff_i^{bound} = avg(aff_i^{boundconditions})$. These are stored within sim-data.

Since affinities drawn from ppGpp represent affinities for all genes without TFs bound, and aff is proportional to aff^{free} and aff^{ppGpp} if they're scaled proportionally, we'll keep the same calculations as in the model previously, and scale $aff^{free,ppGpp}$ by a factor of $aff_i^{one-peak}/aff_i^{minimal}$ for one-peak genes and $aff_i^{unbound}/aff_i^{minimal}$ for two-peak genes. For now, since ppGpp-regulated genes don't overlap with TF-regulated genes, this simply means setting $aff^{free} = aff^{ppGpp} = aff^{unbound,one-peak}$. NOTE: more thought should be put into this when ppGpp-regulated genes do overlap with TF-regulated genes; currently, this assumes that the same ppGpp-fc is applied to the unbound affinity.

We'll also keep the same calculations as before for relating ppGpp and TF-regulated genes during simulations: if a TF is bound to a two-peak genes, then we'll change the new affinity to $aff_i^{unbound} + (aff_i^{bound} - aff_i^{unbound}) \frac{aff_i^{current}}{aff_i^{unbound}}$ where $aff_i^{current}$ is the current affinity calculated from just ppGpp. This calculation will be moved to all occurring during the simulations though, to simplify the code. For the moment, this shouldn't produce any different affinities. NOTE: more thought should be put into this when ppGpp-regulated genes do overlap with TF-regulated genes; it SHOULD change for two-peak genes since currently, this means that if a gene has e.g. higher expression due to ppGpp during rich media, the effect of the TF-binding will also be higher, but this probably isn't right since the overall expression should ultimately more-or-less match the two-peak data, since the effect of TF-binding is calculated directly from rich- and minimal-media expressions (which should've already "taken into account" other effects including ppGpp). NOTE: what to do about polymerizing genes if they're affected by TFs at some point, there's that TODO in the code rn?

6. For polymerizing genes (rRNAs, r-proteins, and RNAPs), we've calculated $aff_i^{condition}$ during each condition by preserving mass-ratios from $aff^{conditioninit}$ for rRNAs and physiological constraints for r-proteins and RNAPs (NOTE: so rRNAs and r-proteins might not match, which might be an issue?). Since these affinities may no longer match the original ppGpp-derived affinities, but we should roughly obtain them from ppGpp-derived affinities (barring interactions with TFs, but we haven't considered those yet), we'll refine $aff^{ppGpp,free}$ for these TUs to best match these new affinities (TODO: check that it shouldn't

change much for rRNAs, but does change for r-proteins and RNAPs since the initial ppGpp-derived fc's for them from data probably don't match physiological constraints that well). The same least-squares problem as previously in the model will be solved:

$$\begin{bmatrix} \frac{(1-f_{rich})a_{free}}{(1-f_{rich})a_{free}+f_{rich}a_{ppGpp}} & \frac{f_{rich}a_{ppGpp}}{(1-f_{rich})a_{free}+f_{rich}a_{ppGpp}} \\ \frac{(1-f_{minimal})a_{free}}{(1-f_{minimal})a_{free}+f_{minimal}a_{ppGpp}} & \frac{f_{minimal}a_{ppGpp}}{(1-f_{minimal})a_{free}+f_{minimal}a_{ppGpp}} \\ \frac{(1-f_{anaer})a_{free}}{(1-f_{anaer})a_{free}+f_{anaer}a_{ppGpp}} & \frac{f_{anaer}a_{ppGpp}}{(1-f_{anaer})a_{free}+f_{anaer}a_{ppGpp}} \end{bmatrix} \cdot \quad (45)$$

$$\begin{bmatrix} aff_i^{free} \\ aff_i^{ppGpp} \end{bmatrix} = \begin{bmatrix} aff_i^{rich} \\ aff_i^{minimal} \\ aff_i^{anaer} \end{bmatrix} \quad (46)$$

Since we don't expect rRNAs, r-proteins, or RNAPs to be one-peak or two-peak genes, and other TF-regulation hasn't been included, the right array doesn't account for TF-effects as of now. NOTE: they should eventually though, maybe something like what is currently in the model. TODO: make them account for TF-effects when their TF-regulation is included.

This will make new $aff_i^{ppGpp,free}$ for each polymerizing gene, and we will set the two sets of ppGpp-related affinities to use these instead, so during simulations the affinities should more-or-less match the physiologically-constrained affinity values allowing for the residuals calculated during least-squares.