

Add a first intro "to better understand x and y, I sought to build a model x. However, ..."

## Chapter 1

# Modelling yeast biosynthesis strategies under constraints

~~challenging~~  
It is difficult to develop a fine-grained model for the aspects of the YMC, especially if the detailed molecular mechanisms are unclear and if the main read-outs from single-cell studies, ~~in~~ NAD(P)H and flavin autofluorescence — are aggregate measures derived from several biochemical phenomena, ~~and~~ It is perhaps more feasible to construct coarse-grained models to answer biological questions about the YMC.

Here, I use a genome-scale metabolic model and flux balance analysis (FBA) to address whether cellular events in the metabolic cycle reflect a strategy in resource allocation that optimises growth.

Specifically, I aim to evaluate these hypotheses:

~~scheduling?~~

1. A finite proteome pool gives rise to temporal partitioning of the synthesis of biomass components during growth: lipid, carbohydrate, amino acids, and nucleic acids. In other words, as an adaption, the cell synthesises biomass components in sequence rather than in parallel. This explains the timing of biosynthetic events in the phases of the yeast metabolic cycle.  
~~would explain~~
2. This resource allocation strategy remains advantageous in many nutrient conditions and ~~in~~ deletion strains. The ~~which would~~ retention of the advantage explains the robustness of the yeast metabolic cycle.

3. If there is a condition in which synthesising biomass components in parallel is advantageous, it is advantageous because the synthesis of biomass components share similar levels of enzymes from the same pathways.

*replaces  
for  
dairy*

## 1.1 Introduction to flux balance analysis

Metabolic network reconstructions are mathematical representations of a set of metabolic pathways in a living system. Usually, each metabolite is represented as a node, and metabolites are connected to each other through reactions that are represented as links (B. Ø. Palsson, 2015). This information can be represented in a two-dimensional stoichiometric matrix, in which the rows of the matrix represent the metabolites, the columns represent the reactions, and the values of each element in the matrix show the stoichiometry of the reactions in the system.

For example, if reaction  $R_1$  is defined by:



the elements in the stoichiometric matrix that correspond to the metabolite-reaction combinations  $(M_1, R_1)$ ,  $(M_2, R_1)$ , and  $(M_3, R_1)$  are -1, -2, and 3, respectively.

Metabolic network reconstructions may include reactions that do not correspond to true, singular reactions in the strict chemical sense, in which one or more chemical species react then create a different set of chemical species as products. Such reactions may include *exchange for nutrients, as well as* reactions that model substrates entering and leaving the biological system. Additionally, many models include a reaction that mod-

*bit unclear*

*flux ≠ metabolite flow*

els biomass formation from chemical species, termed as a *biomass reaction*. A genome-scale metabolic model is, in simple terms, a metabolic network reconstruction that aims to cover every biochemical reaction in a living system that is catalysed by a gene-encoded enzyme.

Flux balance analysis (FBA) is a mathematical method that finds the steady-state flow of metabolites through a metabolic network that is best for a given condition (Orth et al., 2010). Such flows are termed *metabolic fluxes*, which represents rates of chemical reactions. At its core, FBA is a method of solving a linear programming problem of finding the flux values that optimise the output value of an objective function, subject to *biological* constraints. The objective function is given as the task of minimising or maximising a mathematical expression that is based on a subset of fluxes. Most commonly, the objective function for an FBA problem is maximising the flux of a biomass reaction, thus optimising the growth rate of the cell. However, multiple objective functions are possible. The mathematical constraints for FBA are, in the most basic case, imposed by two things: the stoichiometric matrix and reaction flux bounds. The stoichiometric matrix balances reaction inputs and outputs, while flux bounds impose upper and lower limits on the fluxes of each reaction. These constraints restrict the solution space for the FBA problem.

FBA thus offers a computationally inexpensive way to simulate metabolism in a living system, as opposed to solving a set of differential equations that describe the kinetics of biochemical reactions ~~in such a system~~. However, FBA, in its most basic form, has several limitations. It only gives a steady-state picture of metabolism, and therefore cannot be used to describe changes in fluxes over time. Although, dynamic FBA has been developed to solve this problem (Mahadevan et al., 2009).

I suggest you add  
Max C.T.V  
s.t. S v = P  
 $v \leq v \leq V$   
and also the  
GECKO version later  
on.

for discussion  
about extensions to your work

et al., 2002). Additionally, it does not account for concentrations of metabolites. Nevertheless, there are many derivatives of FBA to overcome these limitations and extend the method to answer additional modelling questions. However, these derivatives are outside the scope of this thesis.

## 1.2 Modelling temporal partitioning of biosynthesis

Previous studies have attempted to use FBA to model how each phase of the YMC has different metabolic requirements. Takhayev et al. (2023) showed that in different stages of the cell division cycle, the cell synthesises different components of its biomass at different levels. In the study, they blocked synthesis of each class of macromolecule and recorded the changes in single-cell NAD(P)H cycles, representing the YMC, to quantify the level of each class of macromolecule that the cell synthesises at each time point within a cell division cycle. Then, they used these activities as coefficients for a modified thermodynamic-stoichiometric metabolic model at each time point and used FBA to deduce biomass production rates. Additionally, Cesur et al. (2022) constructed a different FBA model for each YMC phase based on transcriptomic and epigenetic data. They did so using the GIMME algorithm (Becker and B. O. Palsson, 2008), which excludes reactions that correspond to genes that are not expressed at certain time points. Both studies model behaviours at each phase of the YMC, rather than predicting timescale?

✓✓  
Super clear  
Rephrased

An attempt in extending FBA to solve a time-dependent resource allocation problem was Reimers et al. (2017). This study extended a genome-scale model of the cyanobacterium *Synechococcus elongatus* PCC7942 to find the temporal order of intracellular synthesis reactions which optimises the growth rate of the

cell, under resource constraints. However, the model relies on an external oscillator — namely, the light-dark cycle. In contrast, the yeast metabolic cycle is autonomously generated. Therefore, it is difficult to extend the ideas in Reimers et al. (2017) to the study of the YMC.

Traditional genome-scale models assume that the uptake rate of carbon source~~s~~ limits production. However, levels of each enzyme also restrict reaction fluxes, leading to the development of enzyme-constrained models. An enzyme-constrained model fits the assumption that there is a fixed number of amino acids the cell has to distribute (Weiße et al., 2015). Models like Sánchez, Zhang et al. (2017) and Elsemman et al. (2022) constrain the total sum of fluxes based on a defined total amount of enzyme. Elsemman et al. (2022) additionally impose a ribosome capacity constraint and additionally imposes compartment constraints.

In this chapter, I use a genome-scale model of *Saccharomyces cerevisiae* and perform FBA.~~to X & Y.~~ Specifically, I use the enzyme-constrained Yeast8 (ecYeast8) model (Lu et al., 2019). I use this model because it is recent, offering a good coverage of reactions, and is continuously updated in a well-characterised and well-documented software repository. Here, I use the model ecYeast8.6.0, the latest version for which both original and enzyme-constrained variants are available. ecYeast8 uses the GECKO formalism (Sánchez, Zhang et al., 2017) — specifically, GECKO 2 (Domenzain et al., 2022), the latest published version of GECKO. GECKO applies an enzyme constraint by modifying the stoichiometric matrix of a genome-scale metabolic model. ~~ecYeast8.6.0 is the latest version created by GECKO 2.~~ In addition, the model has ‘pseudometabolites’ defined by ~~isa~~ ~~define~~ reactions (Heavner et al., 2012) that group specific chemical species in general classes. These formalisms make it ~~easy~~ <sup>allow</sup> to study each class of biomass component (e.g. lipid, protein, carbohydrate) individually.



Here you can summarize the approach to explain what's to come.  
1.3 model → 1.4 Abstractions → 1.5 timescales → 1.6 pool  
explain here in simple words

## 1.3 The Yeast8 and ecYeast8 models and their formalisms

In this section, I discuss the formalisms used in ecYeast8 because they differ from the usual formalisms used in genome-scale metabolic models.

The model

ecYeast8 contains four formalisms relevant to this chapter:

1. The biomass reaction is defined by having ‘pseudometabolites’ as reactants and a biomass species as a product. These pseudometabolites include lipids, proteins, carbohydrates, RNA, DNA, ions, and cofactors. This is in contrast to BiGG genome-scale models (Norsigian et al., 2020). In these models, biomass reactions have chemical species as reactants, and each has a stoichiometric coefficient that is equal to the species’ abundance in units of  $\text{mmol g}^{-1}_{\text{DW}}$ .
2. The pseudometabolites are defined by ‘*isa*’ reactions, which group specific chemical species into classes of metabolites. These reactions account for how some KEGG definitions of reactions require generic compounds and these reactions also allows flexibility of biomass definition in different growth conditions. The *isa* reactions have chemical species as reactants, each with a stoichiometric coefficient representing the species’ abundance in units of  $\text{mmol g}^{-1}_{\text{DW}}$ , and a pseudometabolite as a product. In effect, the species abundance information is shifted one reaction away from the biomass reaction.
3. The models implements SLIMER (Sánchez, Li et al., 2019), which Splits Lipids Into Measurable Entities, and adds constraints on lipid classes and acyl chain distribution. This formalism is needed because species of lipid backbones and acyl chain can combine to form lipids in more than a thousand ways, and the resulting lipid species are difficult to all be represented

in a genome-scale metabolic model. SLIMER thus introduces reactions that split lipids into their basic components and lipid pseudoreactions to preserve the distribution of acyl chains. As a result, the definitions of lipids are flexible.

4. GECKO was applied to Yeast8 to produce ecYeast8. GECKO modifies the stoichiometric matrix of an input genome-scale metabolic model to account for enzyme abundances and kinetics. Specifically, it adds to enzyme-catalysed reactions enzyme species with a stoichiometric coefficient derived from its  $k_{cat}$  value. The formalism also adds reactions to model drawing enzymes from a pool. GECKO simulates an upper limit of amino acids available for enzyme production.

Here, I detail how I apply these formalisms for this chapter. In particular, I detail GECKO (item 4) and computing molecular weights of pseudometabolites (based on items 1 and 2).

### 1.3.1 GECKO

In a conventional genome-scaled model, metabolic fluxes through reactions are constrained between ~~a lower bound~~ and ~~an upper bound~~. This constraint narrows down the solution space when the objective function is optimised. GECKO imposes an additional constraint on the metabolic fluxes based on the concentration of the enzyme that catalyses the reaction.

As a simple example, for a reaction  $R_j$  catalysed by enzyme  $E_i$  ~~and  $E_i$  only~~,



the formalism imposes:

$$v_j \leq k_{\text{cat}}^{ij} \cdot [E_i] \quad (1.3)$$

where  $v_j$  is the flux of the reaction  $R_j$ ,  $k_{\text{cat}}^{ij}$  is the catalytic constant of the enzyme  $E_i$ , and  $[E_i]$  is the concentration of the enzyme  $E_i$ . In other words, the flux through the reaction must not exceed  $v_{\max}$ . (Can you use  $R_i$  instead of  $R_j$ ?)

To apply this constraint, GECKO modifies reactions in the genome-scaled model it is applied to. For example, using equation 1.2, GECKO adds a term to the equation, modifying the stoichiometric matrix, to make it:



with the stoichiometric coefficient  $n_{ij} = \frac{1}{k_{\text{cat}}^{ij}}$ .

This transformation, adding the enzyme as a pseudo-reactant, is based on the interpretation that the system uses some amount of enzyme at a specific time to catalyse the flux going through the reaction.

Slightly different formalisms are applied to reversible reactions, isozymes, promiscuous enzymes, and enzyme complexes. Namely:

- Reversible reactions are modelled as the forward and reverse reactions separately.
- For isozymes, the original reaction is copied multiple times corresponding to the number of reactions that the isozyme catalyses. Each has an isozyme catalysing the reaction. In addition, there is an ‘arm’ reaction to act as an intermediate between the substrate and the products.
- No actions are needed for promiscuous enzymes.
- Enzyme complexes are modelled as one reaction that uses all subunit proteins that all share the same  $k_{\text{cat}}$  value.

GECKO adds an enzyme as a pseudo-reactant for all enzyme-catalysed reactions in the genome-scale model to create an enzyme-constrained model.

To constrain enzyme levels in the model, GECKO defines a pseudo-reaction



with a flux

$$e_{\text{pool}} \leq (P_{\text{total}} - P_{\text{measured}}) \cdot f \cdot \sigma \quad (1.6)$$

in units of  $\text{g g}_{\text{DW}}^{-1}$ , where

*"P<sub>total</sub> and P<sub>measured</sub> over ..."*

- $P$  represents protein fraction with respect to the dry weight of the cell,
- $f$  represents the fraction of proteins that are enzymes, and
- $\sigma$  is a parameter that represents the average saturation of enzymes.

ecYeast8.6.0 assumes parameter values of:  $f = 0.5$ ,  $P_{\text{total}} = 0.5$ , and  $\sigma = 0.5$ .

Defining such parameters is a judgement call, especially when the protein fraction varies across growth rates (Elsemman et al., 2022), but  $f = 0.5$  is close to the protein mass fraction of ecYeast8.6.0 (to be discussed in section 1.3.2). Subsequently, GECKO changes the carbohydrate composition based on the assumption that a change in the amino acid composition is offset by the reverse change in the carbohydrate composition; experimental data justifies this assumption. (ref)

Then, for each enzyme  $E_i$ , GECKO defines enzyme usage pseudoreactions of the form



with  $MW_i$  being the molecular weight of the enzyme in units of g mmol<sup>-1</sup>. The flux of enzyme usage pseudoreactions are defined in units of mmol g<sub>DW</sub><sup>-1</sup>.

Taken together, the modelled cell thus has an enzyme pool in terms of a mass fraction of the cell's dry weight, and the modelled cell allocates certain fractions of this mass to the synthesis of each enzyme at steady-state. The mass of each enzyme in the cell determines the amount (in moles) of each enzyme and therefore its catalytic activity.

GECKO takes  $k_{cat}$  values from BRENDA (Chang et al., 2021) and enzyme data from SWISSPROT (The UniProt Consortium, 2023) and KEGG (Kanehisa et al., 2023), including molecular weight of protein<sub>S</sub> and associated pathways.

### 1.3.2 Computing molecular weights of pseudometabolites

In the yeast cell, biomass components (lipids, carbohydrates, proteins) are present in different fractions. Knowing these fractions<sub>IS</sub> are useful in computing the time it takes to synthesise each biomass component. These fractions are represented in the biomass reaction. However, the mass fraction of each component varies according to strain and growth rate (Nilsson and Nielsen, 2016; Elsemman et al., 2022). It is not straightforward to implement these changes as stoichiometric coefficients of the chemical species that define biomass, because there is limited information for the mass fractions of such species as conditions vary, and because of the many combinations of lipid backbones and acyl chains. Therefore, I computed the mass fractions of each biomass component based on the stoichiometries of species in the Yeast8 model, as a back-of-the-envelope calculation for such a coarse-grained modelling effort.<sup>2</sup> Similar calculations were performed by Takhayev et al. (2023).

Here it would help to show the objective function  
 $S = \sum \text{or similar}$  and define "mass fractions" below  
you mention them after.

*isa* reactions define pseudometabolites by having chemical species with known molecular weights as reactants, with their stoichiometric coefficients representing abundance in mmol g<sub>DW</sub><sup>-1</sup>. Therefore, I can treat each pseudometabolite as a chemical species and calculate its molecular weight by assuming mass balance (Chan et al., 2017; Dinh et al., 2022; Takhayev et al., 2023). The resulting molecular weight will thus represent the mass fraction of each biomass component in units of g g<sub>DW</sub><sup>-1</sup>.

*To this end, first*  
~~For this to be possible, I computed molecular weights for pseudometabolites that represent macromolecules and other biomass components in the ecYeast8 model, as the ecYeast8 model does not specify the molecular weights of these bulk metabolites. The pseudometabolites include: lipids, proteins, carbohydrates, RNA, DNA, cofactors, and ions. Namely, I assumed that in reactions that produce the pseudometabolites, there is conservation of mass, and therefore:~~

$$\sum_{s=1}^{m_s} m_s c_s = \sum_{p=1}^{m_p} m_p c_p \quad (1.8)$$

where:

- $m_s$  represents molar mass, *of substrate s*.
- $c_s$  represents stoichiometric coefficient, ..
- $s = 1, \dots, (\text{number of substrates})$  represents substrates of the reaction in question, and
- $p = 1, \dots, (\text{number of products})$  represents products.

To compute the molecular weight of the carbohydrate pseudometabolite, I inspected the pseudoreaction **r\_4048** that creates it:

*subsection*

0.684535 (1->3)-beta-D-glucan + 0.228715 (1->6)-beta-D-glucan  
 + 0.330522 glycogen + 0.650171 mannan + 0.126456 trehalose  
 -> carbohydrate

Here, the molecular weights of all species except for carbohydrate, the pseudometabolite, are represented in the model. Thus, equation 1.8 can be applied to compute the molecular weight of the carbohydrate pseudometabolite. The same process can be applied to compute the molecular weights of the DNA, RNA, cofactor, and ion pseudometabolites. This is because the equations similarly have reactants with molecular weights represented in the model and only the pseudometabolite, the sole product, does not have a molecular weight specified.

*explain how it is applied*

*Put in Table?*

The molecular weight of the DNA pseudometabolite was computed from reaction r\_4050:

0.0036 dAMP + 0.0024 dCMP + 0.0024 dGMP + 0.0036 dTMP  
 -> DNA

The molecular weight of the RNA pseudometabolite was computed from reaction r\_4049:

0.0445348 AMP + 0.0432762 CMP + 0.0445348 GMP + 0.0579921 UMP  
 -> RNA The molecular weight of the cofactor pseudometabolite was computed from reaction r\_4598:

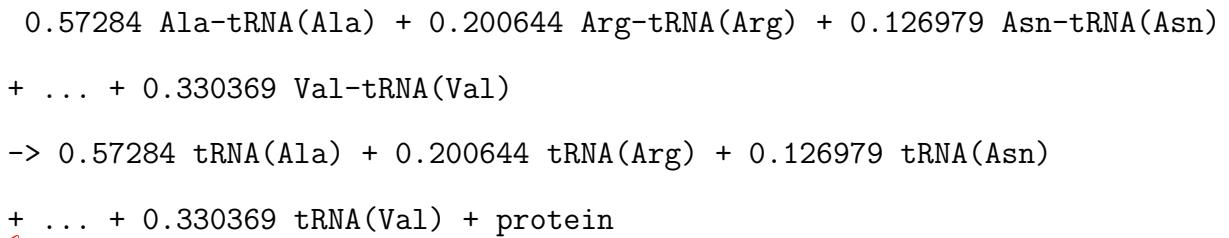
0.00019 coenzyme A + 1e-05 FAD + 0.00265 NAD + 0.00015 NADH  
 + 0.00057 NADP(+) + 0.0027 NADPH + 0.00099 riboflavin  
 + 1.2e-06 TDP + 6.34e-05 THF + 1e-06 heme a  
 -> cofactor

And the molecular weight of the ion pseudometabolite was computed from reaction r\_4599:

3.04e-05 iron(2+) + 0.00363 potassium + 0.00397 sodium + 0.02 sulphate  
 + 0.00129 chloride + 0.00273 Mn(2+) + 0.000748 Zn(2+) + 0.000217 Ca(2+)  
 + 0.00124254 Mg(2+) + 0.000659 Cu(2+)  
 -> ion

Other metabolites were less straightforward and required some judgement calls.

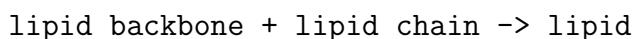
*Subsection* To compute the molecular weight of the protein pseudometabolite, I inspected reaction r\_4047:



*Unclear* In this reaction, the aminoacyl-tRNA reactants are represented in the form of the atoms that make up the aminoacyl residues plus R to represent the tRNA, and the tRNA products are represented as RH. For example, Ala-tRNA(Ala), alanyl-tRNA, is represented as C3H7NOR. The protein pseudoreaction shows how different proportions of each aminoacyl-tRNA combine to form the cell's proteins, so it is safe to discard the R symbol that corresponds to the tRNA from the reaction. Upon doing so, the mass balance represented by equation 1.8 can be applied to compute the molecular weight of the protein pseudometabolite.

*Subsection* Finally, the lipid pseudometabolite is the least straightforward because some of the reactants of the lipid pseudoreaction do not have molecular weights specified.

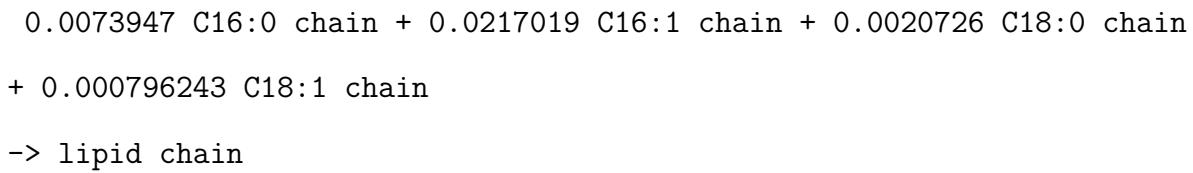
The lipid pseudoreaction is represented in reaction r\_2108:



And both lipid backbone and lipid chain have no molecular weight specified.

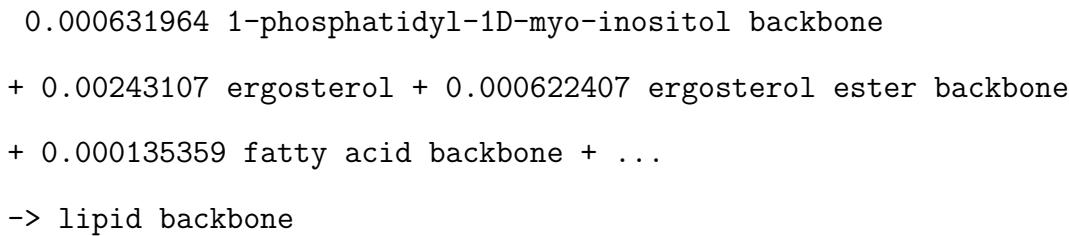
*the model does not specify the*

Reaction r\_4065 specifies a lipid chain pseudoreaction, in which **lipid chain** is generated:



As all reactants have molecular weights defined in the model, the molecular weight of **lipid chain** can be computed from the mass balance of this reaction.

Reaction r\_4063 specifies a lipid backbone pseudoreaction, in which **lipid backbone** is generated:



Within this reaction, all reactants have defined molecular weights except for **fatty acid backbone**. Because of SLIMEr, four reactions in the model produce **fatty acid backbone**:

All species in these reactions except for **fatty acid backbone** have molecular weights specified, and thus I applied mass balance to find the molecular weight of **fatty acid backbone** from each reaction. However, the molecular weights computed from each equation are different, as shown in table 1.1. Since the differences are slight, and ultimately I am making a back-of-the-envelope calculation, I took the mean of the four weights ( $729.53 \text{ g mol}^{-1}$ ) as the molecular weight of **fatty acid backbone** for the purposes of computing the molecular weight of **lipid backbone** (which then yields  $21.31 \text{ g mol}^{-1}$ ). too convolutional

*could be  
made clearer*



ID	Reaction	Computed molecular weight (g mol <sup>-1</sup> )
	palmitate	
r_3975	-> 0.255421 fatty acid backbone 0.256429 C16:0 chain	742.54
	palmitoleate	
r_3976	-> 0.253405 fatty acid backbone 0.254413 C16:1 chain	744.56
	stearate	
r_3977	-> 0.283475 fatty acid backbone 0.284483 C18:0 chain	714.49
	oleate	
r_3978	-> 0.281459 fatty acid backbone 0.282467 C18:1 chain	716.51

**Table 1.1:** ecYeast8 reactions that generate the **fatty acid backbone** metabolite

With the molecular weights of **lipid backbone** and **lipid chain** defined, I can now compute the molecular weight of **lipid** using mass balance.

In summary, the molecular weights of the pseudometabolites are:

Metabolite	Computed molecular weight (g mol <sup>-1</sup> )
Protein	504.37
Carbohydrate	350.37
RNA	64.04
Lipid	31.57
Cofactors	4.83
DNA	3.90
Ions	2.48

**Table 1.2:** Computed molecular weights of bulk metabolites in ecYeast8

As additional validation, the ratio between the molecular weights are similar to the ratio between the mass of each class of macromolecule in the yeast cell dry weight shown by Canelas et al. (2011).

*of what? → we observe that*

*I TALK*

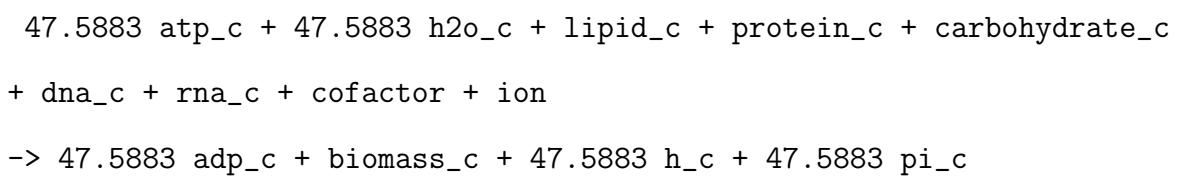
*Idea: Could 1.3.1 - 1.3.2. be moved to an Appendix and you present a summary only here?*

Adding together values in table 1.1 gives a the molecular weight of the biomass pseudometabolite of  $961.57 \text{ g mol}^{-1}$ , close to the  $966 \text{ g mol}^{-1}$  computed by Takhveev et al. (2023) from a different genome-scale model. In theory, this number should be  $1000 \text{ g mol}^{-1}$  because the stoichiometric coefficients of the species that form biomass components are expressed in terms of  $\text{mmol g}_{\text{DW}}^{-1}$  (Thiele and B. Ø. Palsson, 2010; B. Ø. Palsson, 2015), but the deviation from 1000 could be explained by the SLIMER formalism. Plus, the sum of stoichiometric coefficients are not always verified in genome-scale models (Chan et al., 2017).

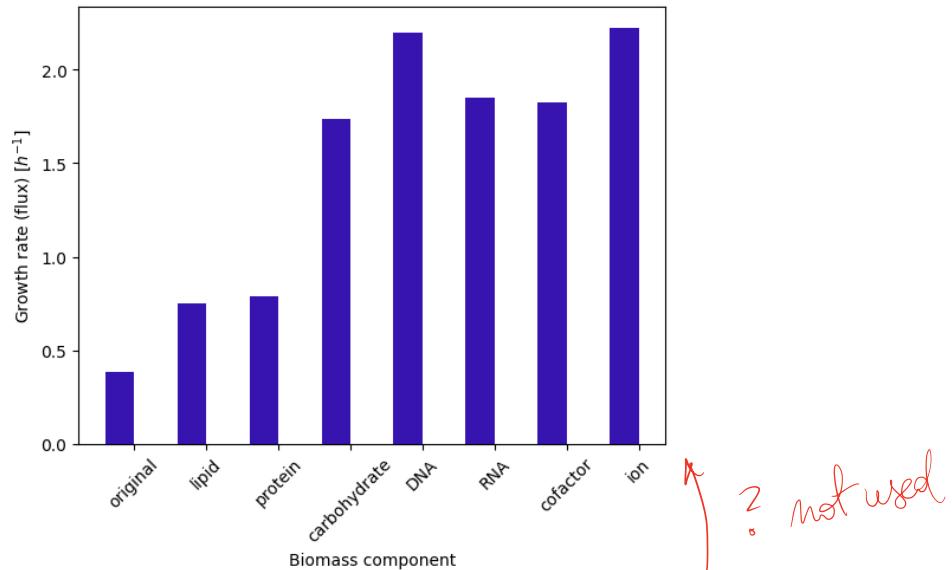
## 1.4 Ablating pseudometabolites from the biomass reaction

To simulate producing each class of biomass component in turn, I take advantage of pseudometabolites in ecYeast8 to remove each of them in turn from the biomass reaction — in other words, ablating pseudometabolites from the biomass reaction. Specifically, I exclude each pseudometabolite from the biomass reaction, in turn, and optimise the model.

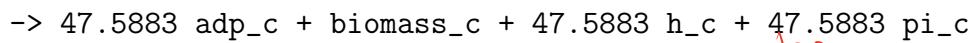
Consider the objective function, the biomass reaction:



There are seven pseudometabolites: lipid, protein, carbohydrate, DNA, RNA, cofactor, and ion. To have the cell prioritise biosynthesis of lipids, I set the stoichiometric coefficients of all pseudometabolites except for lipids to zero in the above equation, giving:



**Figure 1.1:** Growth rates from the original (wild type) model ( $\lambda_0$ ; leftmost bar) and from the ablated versions of the model ( $\lambda_{\text{seq},i}$ , where  $i$  represents each biomass components among lipids, proteins, carbohydrates, DNA, RNA, cofactors, and ions; other bars)



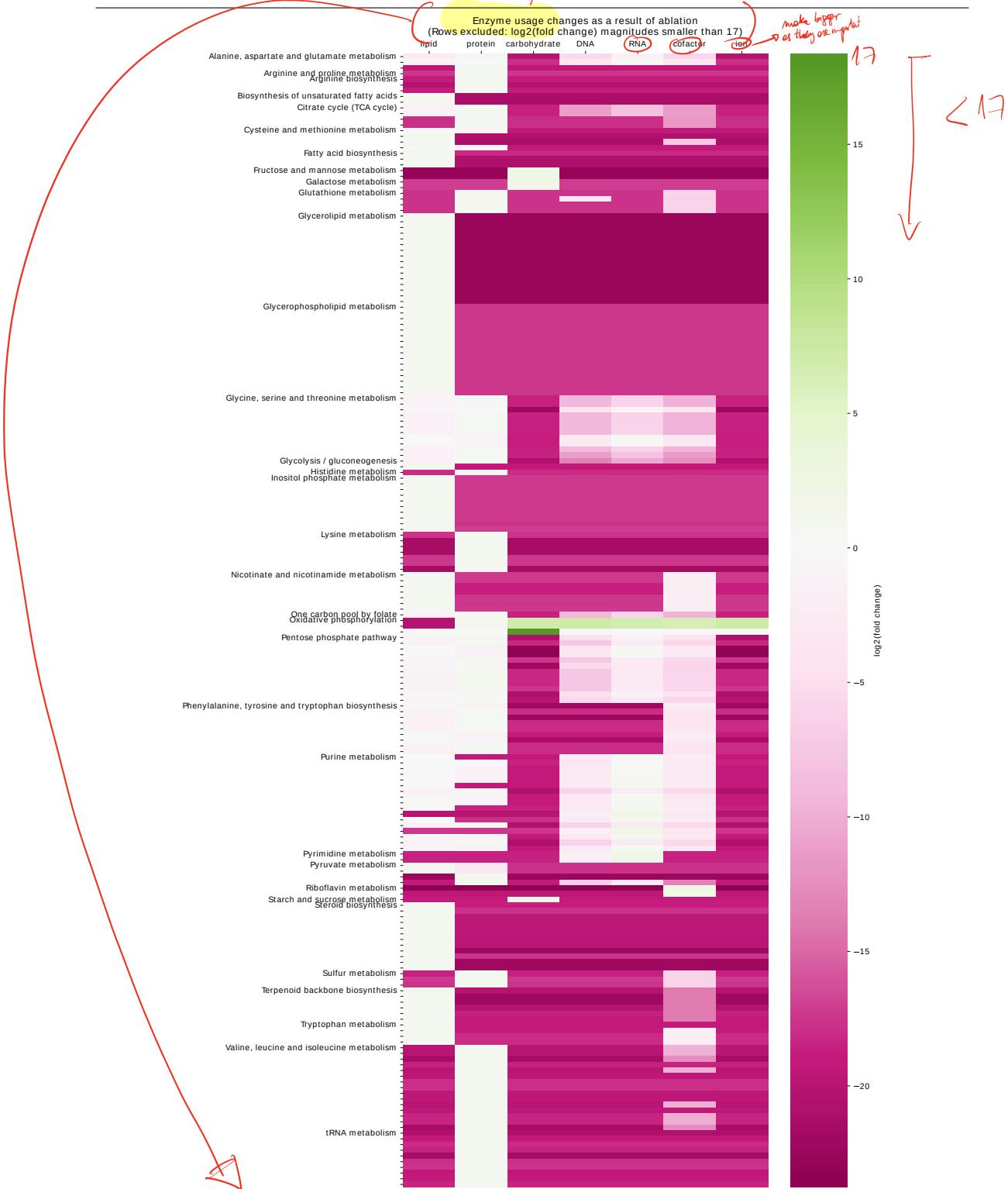
And I repeated this for the other pseudometabolites

Optimising the objective function as each pseudometabolite is prioritised gives changed growth rates (figure 1.1).

When the model prioritises a different biomass component in each round of ablation, the model partitions the limited proteome available for enzyme production differently. In other words, in each round of ablation, the vector of fluxes carried by each enzyme usage pseudoreaction (defined in equation 1.7) is different from each other and from the non-ablated case.

## 1.4. Ablating pseudometabolites from the biomass reaction

18



**Figure 1.2:** The cell changes how it allocates its proteome to enzymes when different components of its biomass are prioritised. Each column shows the component that remains in each round of pseudometabolite ablation. Rows represent enzyme usage reactions, sorted by subsystem. Colours represent how the cell re-allocates its proteome to each enzyme: green shows an increase in reaction flux, while pink shows a decrease in flux. Fluxes with magnitude less than  $\epsilon = 1.11 \times 10^{-11} \text{ mmol g}_{\text{DW}}^{-1}$  are treated as  $\epsilon$  flux, for the purpose of computing fold changes. Cases where  $|\log_2(\text{FC})| > 17$  are excluded to restrict the number of reactions ( $n = 3897$ ) to aid display purposes.

clarify terminology  
and refer to  
(Eq 1.7)

move 2 pages down

?

Subsystem information aids interpretation of these changes. Here, I match the flux carried by each enzyme usage pseudoreaction to the enzyme-catalysed reaction that the enzyme is associated with. If an enzyme usage pseudoreaction is associated with multiple enzyme-catalysed reactions, data entries were duplicated accordingly. Then, the subsystem associated with each enzyme-catalysed reaction is taken from the gene-protein map, and noted.

$$\log_2(FC) = \log_2\left(\frac{X}{Y}\right)$$

where X is ...  
and Y is ...

*quantify*  
~~emphasis~~ To emphasise changes in proteome allocation across rounds of ablation, I compute the  $\log_2(FC)$  — ~~the base-2 logarithm of fold change~~ — of fluxes relative to the non-ablated case. Some reaction fluxes in some cases are zero, making the logarithm of fold change undefined. As a workaround, I defined a threshold small flux  $\epsilon = 1.11 \times 10^{-11} \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$ , which corresponds to one molecule per cell per hour, assuming 15 pg dry weight per cell (Sherman, 2002). My justification for the threshold value is that if a reaction is important enough for the cell, the cell fires the reaction more than once in an hour, which is a large proportion of its doubling time. For the purposes of computing fold changes, fluxes with magnitude less than  $\epsilon$ , including zero flux, are thus treated as  $\epsilon$  flux. Therefore, extremely high  $\log_2(FC) (> 10)$  likely indicates an original zero flux becoming non-zero in ablation, interpreted as an enzyme ‘switching-on’. Extremely low  $\log_2(FC) (< -10)$  likely indicates an original non-zero flux becoming zero in ablation, interpreted as an enzyme ‘switching-off’. Fold changes thus indicate the cell changing priorities: a positive  $\log_2(FC)$  represents more of the proteome allocated to the production of the relevant enzyme, while a negative  $\log_2(FC)$  represents less of the proteome allocated to the production of the relevant enzyme.

Figure 1.2 shows the subsystems and fold changes calculated as described above. In particular, it shows:

too conventional, can you use equations?

1. In most cases, enzymes are ‘switched-off’, though lipid-prioritised biosynthesis is most similar to the parallel case. *you haven't really defined 'parallel case'*
2. Increased fatty acid biosynthesis, glycerolipid metabolism, glycerophospholipid metabolism, inositol phosphate metabolism, steroid biosynthesis, and terpenoid backbone biosynthesis when lipid is prioritised. Decreases in amino acid metabolism, which varied depending on the amino acid. *for example...*
3. Small increases in amino acid metabolism, tRNA metabolism, and oxidative phosphorylation when protein is prioritised. Decrease in glycine, serine, and threonine metabolism is strong in carbohydrate and ion, but weaker in DNA, RNA, and cofactor prioritisation.
4. Increase in fructose and mannose metabolism when carbohydrate is prioritised. Strong increase in riboflavin metabolism when cofactor is prioritised (riboflavin is a cofactor). Increase in pentose phosphate pathway, purine metabolism, and pyrimidine metabolism when RNA is prioritised. However, decrease in pentose phosphate pathway when DNA is prioritised.
5. For carbohydrate, DNA, RNA, cofactor, ion, strong increases in glycolysis/gluconeogenesis and oxidative phosphorylation.

In sum, changes in allocation in rounds of ablation, by subsystem, is biologically relevant and is reasonable given the roles of the enzymes.

*This needs some more explanation.*

## 1.5 Estimating timescale of biosynthesis

I used the ecYeast8.6.0 model with glucose uptake, defined by the bounds of its exchange reaction flux, restricted to within  $0 \text{ mmol g}^{-1}_{\text{DW}}$  to  $16.89 \text{ mmol g}^{-1}_{\text{DW}}$ . The upper limit is based on the glucose uptake rate found by FBA from using the wild type model to optimise growth rate. This value agrees with theoretical

*unclear why you need to say this now.  
Has the model changed?*

*Needs work*

*intro sentences.*

saturation rates of 16 mmol g<sub>DW</sub><sup>-1</sup> to 19 mmol g<sub>DW</sub><sup>-1</sup> (Blank and Sauer, 2004). Using the biomass reaction as the objective function, I optimise the model to obtain the predicted growth rate, and computed the doubling time based on this growth rate as follows:

This is going back to what was already shown in Fig. 1.1

$$t_0 = \frac{\ln 2}{\lambda_0} \quad (1.9)$$

where  $t_0$  is the doubling time and  $\lambda_0$  is the growth rate, equivalent to the optimised flux of the biomass reaction.

I ablate components in the biomass reaction in turn as discussed in section 1.4.

change to active voice and less conversational

In each round, the model is simulated, using the modified biomass reaction as the objective function, and the flux is recorded. Synthesis time is computed, taking into account the mass fraction of each biomass component — i.e. if the component is a smaller fraction of the cell, it takes less time. Specifically, the computation is:

$$t_{\text{seq},i} = f_i \cdot \frac{\ln 2}{\lambda_{\text{seq},i}} \quad (1.10)$$

where:

- $i$  represents each of the biomass components (lipids, proteins, carbohydrates, DNA, RNA, cofactors, and ions),
- $t_{\text{seq},i}$  is the predicted time for synthesis of each biomass component,
- $f_i$  is the mass fraction of each biomass component, and
- $\lambda_{\text{seq},i}$  is the optimal flux of the ablated biomass reaction.

$f_i$  for each biomass component is computed by dividing the molecular weight of the corresponding pseudometabolite by the molecular weight of biomass (table 1.2).

For comparison, I computed estimates of the time for each biomass component, assuming that it is proportional to the mass fraction:

$$t_{\text{par},i} = f_i \cdot t_0 \quad (1.11)$$

where  $t_0$  is the doubling time found in equation 1.9.

To determine whether sequential biosynthesis of biomass components or parallel biosynthesis of biomass components is advantageous, I devise a ratio  $\tau_{\text{seq/par}}$  that represents the ratio between the total time predicted by ablation and the biomass component that is predicted to take the most time, i.e.

*where  $T_{\text{seq}}$  and  $T_{\text{par}}$  ...*

$$\tau_{\text{seq/par}} := \frac{T_{\text{seq}}}{T_{\text{par}}} \quad (1.12)$$

$T_{\text{seq}}$  represents sum of times from focusing on each biomass component — thus representing time predicted, assuming biomass components are synthesised in turn:

$$T_{\text{seq}} := \sum_i t_{\text{seq},i} = \sum_i f_i \cdot \frac{\ln 2}{\lambda_{\text{seq},i}} \quad (1.13)$$

$T_{\text{par}}$  represents the time of synthesis of the biomass component that takes the most time, if time is assumed to be proportional to mass fraction. This is always protein as it accounts for 52.5% of the mass fraction (table 1.2):

$$T_{\text{par}} := \operatorname{argmax}_i t_{\text{par},i} = t_{\text{seq,protein}} = f_{\text{protein}} \cdot \frac{\ln 2}{\lambda_0},$$

$\therefore \operatorname{argmax}_i f_i = f_{\text{protein}}$

(1.14)

$T_{\text{par}}$  thus represents the limiting, slowest process.

Therefore,

$$\begin{aligned}
 \tau_{\text{seq/par}} &= \frac{T_{\text{seq}}}{T_{\text{par}}} \\
 &= \left( \sum_i \frac{f_i}{\lambda_{\text{seq},i}} \right) \cdot \frac{\lambda_0}{f_{\text{protein}}} \\
 &= \left( \frac{f_{\text{lipid}}}{\lambda_{\text{seq,lipid}}} + \frac{f_{\text{protein}}}{\lambda_{\text{seq,protein}}} + \dots + \frac{f_{\text{ion}}}{\lambda_{\text{seq,ion}}} \right) \cdot \frac{\lambda_0}{f_{\text{protein}}}
 \end{aligned} \tag{1.15}$$

This expression means that the definition of the  $\tau_{\text{seq/par}}$  ratio does not reduce to a trivial expression and depends on the  $\lambda_0$  and the  $\lambda_{\text{seq},i}$  values, which are all independent of each other. (This relationship between  $\tau_{\text{seq/par}}$  and  $\lambda_0$  and  $\lambda_{\text{seq},i}$  is explored in section 1.6.) ?

A  $\tau_{\text{seq/par}} < 1$  means that synthesising biomass components in sequence saves more time, while  $\tau_{\text{seq/par}} > 1$  indicates that parallel synthesis of biomass components is favoured as synthesising biomass components in sequence does not save time.

Figure 1.3 summarises the variables discussed in this section.

In figure 1.3, the  $\tau_{\text{seq/par}} = 0.70$ . The relationship  $\tau_{\text{seq/par}} < 1$  also holds for auxotrophs and deletion strains (figure 1.4). This supports experimental evidence that shows that auxotrophs and deletion strains have YMCs. [ref]



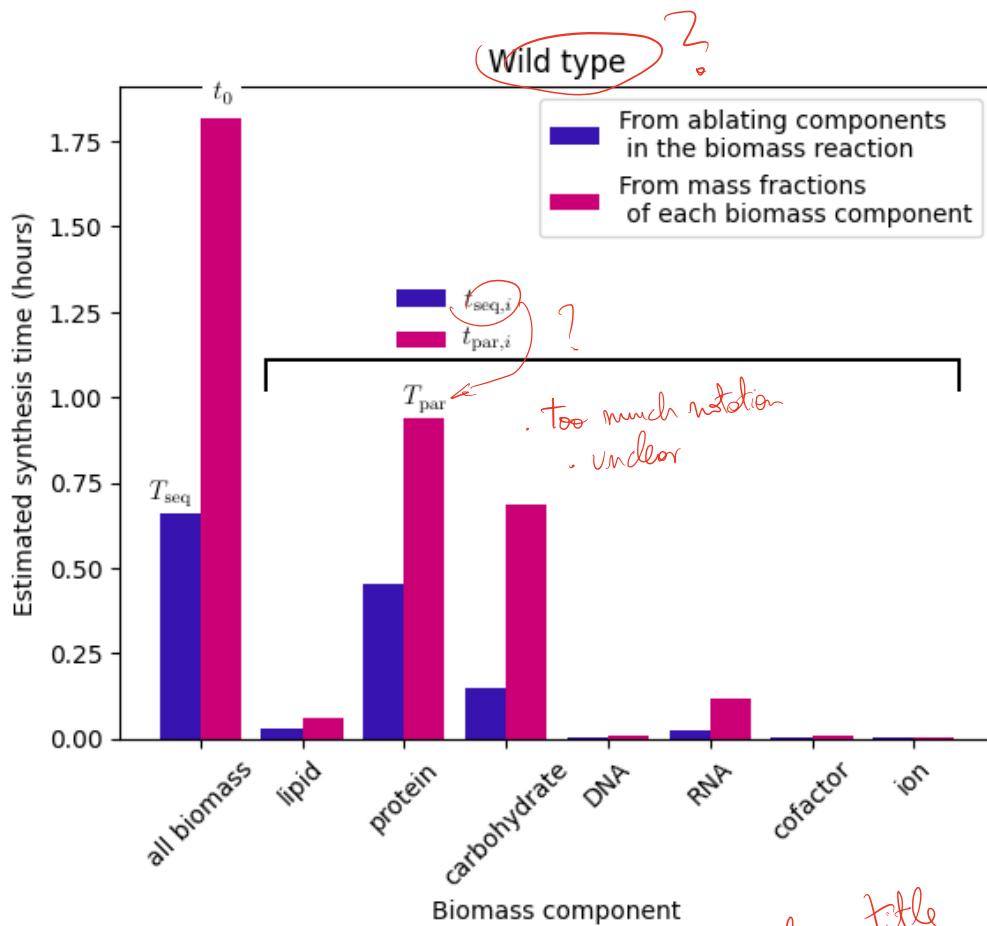
NEEDS WORK

## 1.6 Effect of restricting the enzyme pool

There are multiple approaches to evaluate the hypothesis of whether a restriction of the proteome pool favours sequential biosynthesis of biomass components. Here, I discuss three commonly-used approaches — parsimonious FBA, regularised FBA, and constraining the sum of absolute values of fluxes — before justifying the use of directly varying a parameter in the ecYeast8 model to take advantage of GECKO.

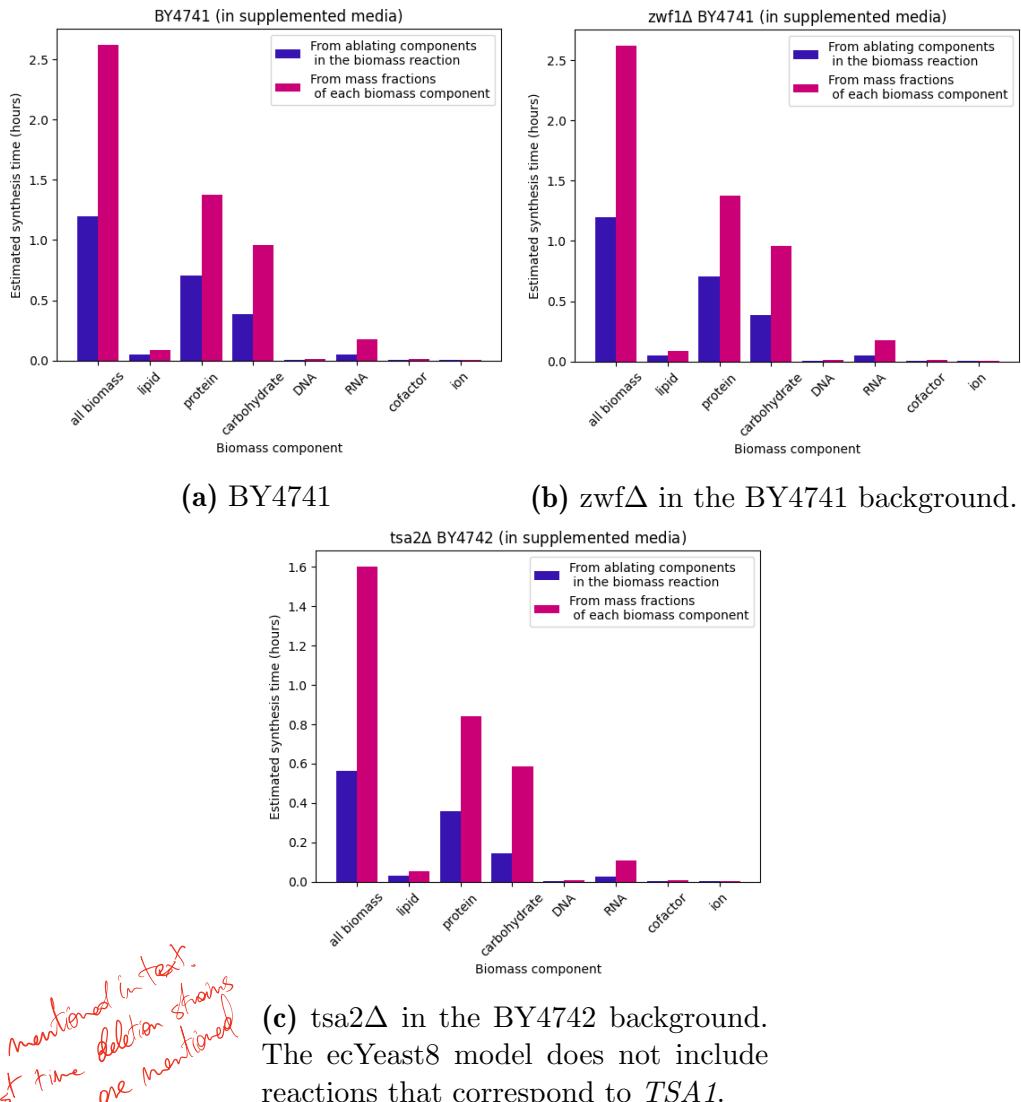
needs to be motivated.  
Not a hypothesis you  
have motives  
before

why is this important/interesting?



**Figure 1.3:** Comparing time scales derived from ablating components of the biomass reaction (blue) and from assuming that synthesis time is proportional to the mass fraction of the biomass component (red). For each biomass component, the time  $t_{\text{seq},i}$  was computed according to equation 1.10 (blue bars apart from leftmost column), and the time  $t_{\text{par},i}$  was computed according to equation 1.11 (red bars apart from leftmost column).  $T_{\text{par}}$  is defined as the greatest  $t_{\text{par},i}$  (equation 1.14). Blue under ‘all biomass’ is  $T_{\text{seq}}$  (equation 1.13), the sum of times derived from ablation (other blue bars). Red under ‘all biomass’ is the doubling time  $t_0$  (equation 1.9).

figures need a title  
TALK



**Figure 1.4:** Comparing time scales from sequential and parallel biosynthesis in auxotrophs and deletion strains, as in figure 1.3. For BY4741-background strains, supplements are simulated by allowing uptake of histidine, leucine, tryptophan, methionine and uracil. The same applied to BY4742-background strains, but lysine uptake replaces methionine uptake.

- Parsimonious FBA (Lewis et al., 2010) first uses FBA to compute the optimal growth rate, fixes this value, and then minimises the sum of gene-associated reaction fluxes while maintaining optimal growth. Depending on the software package, this minimisation either minimises the sum of fluxes (COBRA, for MATLAB), the sum of the absolute values of each flux (*cobraPy*, for the Python programming language) or squared sum (COBREXA, for the Julia programming language). I reject this approach because it fixes the growth rate, but I aim to see how constraints affect cell strategies, including the growth rate, as the constraints vary along a spectrum. Additionally, parsimonious FBA relies on reducing the subset of genes that contribute to the solution. In other words, it modifies the model and it also relies on good gene-protein annotations, with the latter not always guaranteed.

- Regularised FBA is defined as adding a regularisation parameter to the objective function. Vijayakumar et al. (2020) describe a quadratic program for solving a regularised two-level FBA:

First time the max problem  
is shown

$$\max g^\top v - \frac{\sigma}{2} v^\top v \quad (1.16)$$

where  $g$  is the objective function,  $v$  is the flux vector, and  $\sigma$  is a regularisation parameter than can be tuned.

I reject this approach because it requires a quadratic solver, making it difficult and computationally expensive to implement. Additionally, the behaviour is not as expected when I implemented it: the growth rate does not change as the regularisation parameter  $\sigma$  varies. I expected a trade-off between growth and reaction fluxes and the balance between these two quantities to change as this parameter varies.

This is one of the first times the max problem is shown

Constraining the sum of absolute values of fluxes is simply defined as fixing

$$\sum_i |v_i| < c \quad (1.17)$$

where  $v_i$  represents each flux of each reaction, and  $c$  is a constant to be varied.

This is reasonable for the original Yeast8 model without the enzyme constraint (as opposed to ecYeast8). As  $c$  decreases to 0, the original growth rate  $\lambda_0$  and ablated growth rates  $\lambda_{\text{seq},i}$  decrease to 0 because at  $c$  values near 0, flux values, including that of the biomass reaction, can only take small values. However, if this approach is applied to ecYeast8, there is double imposition of constraints: on (a) constraints on enzyme usage imposed on the enzyme-usage pseudoreactions created by GECKO, and on (b) constraints on sum of the absolute values of fluxes. This will confuse interpretation.

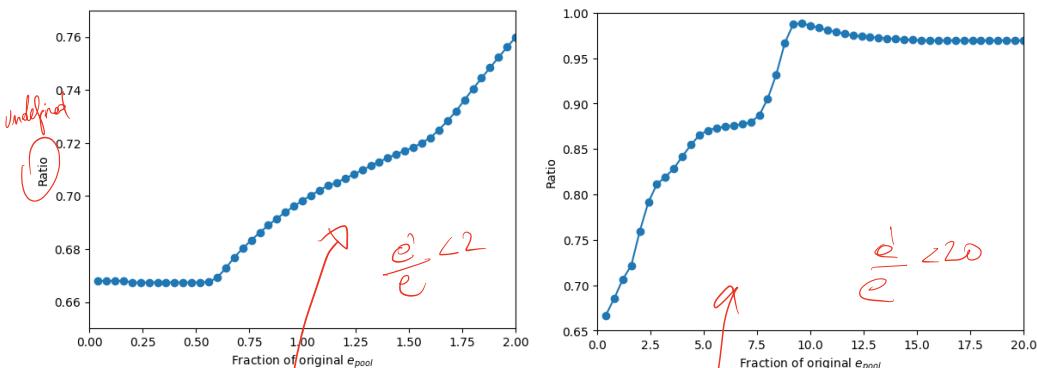
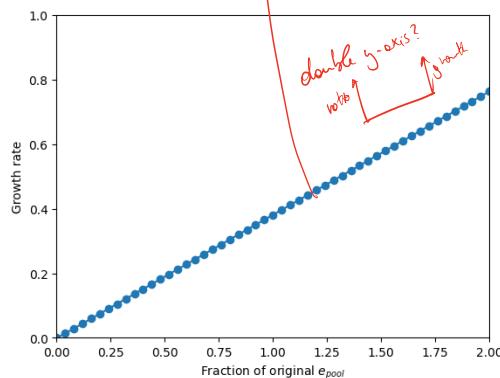
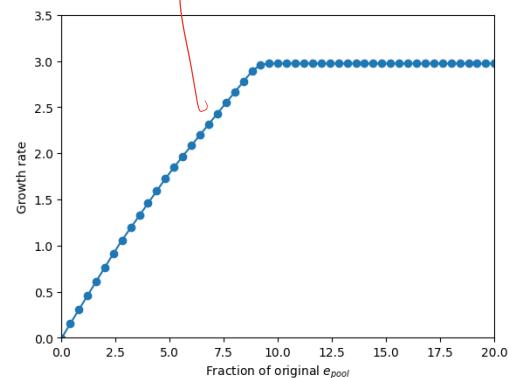
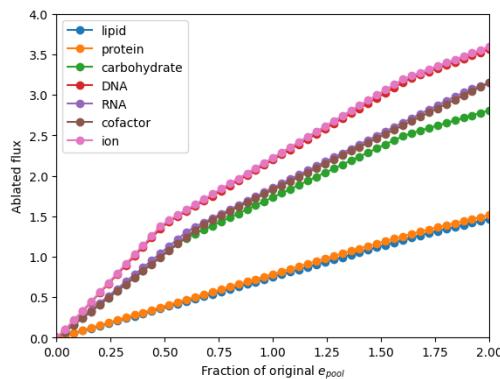
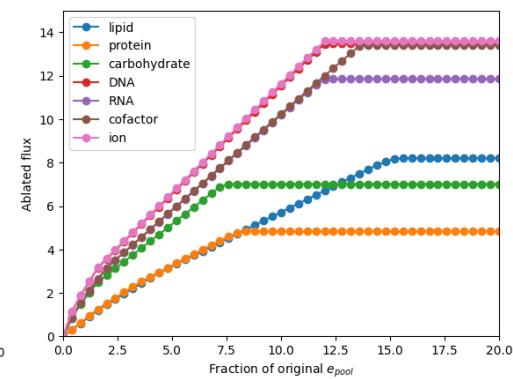
Therefore, I decide to vary the enzyme-available proteome pool to study proteomic constraints. This takes advantage of a GECKO formalism that is easy to modify and interpret.

To vary the enzyme-available proteome pool in the ecYeast8 model, I vary the value of the upper limit of the flux  $e_{\text{pool}}$  of the enzyme pool pseudoreaction (equation 1.5) in order to change the flux available for enzyme usage pseudoreactions (equation 1.7). My reasoning is that with a smaller  $e_{\text{pool}}$ , the sum of fluxes of enzyme usage pseudoreactions must decrease, and the model must decide which enzyme usage pseudoreactions to allocate a higher flux to. This models the biological situation in which the cell has a smaller enzyme-available proteome pool, so the cell must decide which enzymes to allocate the greatest proportions of the pool to, for enzyme synthesis.

This is the key point.  
Jump right in

Show some equations

This is the motivation for this section

(a) Effect on ratio ( $\tau_{\text{seq/par}}$ ),  $e'_\text{pool} \leq 2e_\text{pool}$ .(b) Effect on ratio ( $\tau_{\text{seq/par}}$ ),  $e'_\text{pool} \leq 20e_\text{pool}$ .(c) Effect on wild type growth rate ( $\lambda_0$ ),  $e'_\text{pool} \leq 2e_\text{pool}$ .(d) Effect on wild type growth rate ( $\lambda_0$ ),  $e'_\text{pool} \leq 20e_\text{pool}$ .(e) Effect on ablated growth rates ( $\lambda_{\text{seq},i}$ ),  $e'_\text{pool} \leq 2e_\text{pool}$ .(f) Effect on ablated growth rates ( $\lambda_{\text{seq},i}$ ),  $e'_\text{pool} \leq 20e_\text{pool}$ .

**Figure 1.5:** Constraining the proteome pool available for synthesis enzymes leads to a greater advantage of sequential biosynthesis of biomass components over parallel biosynthesis, as evidenced by a decreasing  $\tau_{\text{seq/par}}$  ratio if  $e'_\text{pool}$  decreases (1.5a). Concurrently, as  $e'_\text{pool}$  decreases, the wild type growth rate ( $\lambda_0$ ) decreases linearly to zero (1.5c) and ablated growth rates ( $\lambda_{\text{seq},i}$ ) decrease in linear segments independently of each other and of the growth rate (1.5c). These values determine the ratio according to equation 1.15.

- △ Add figure title
- Explain what is shown, and the what it means

Figure 1.5 shows the results of my investigation. I denote  $e_{\text{pool}}$  as the default enzyme-available proteome pool ( $0.104 \text{ g g}_{\text{DW}}^{-1}$ ) and  $e'_{\text{pool}}$  as the enzyme-available proteome pool I set in the model. When  $0 \leq e'_{\text{pool}} \leq 2e_{\text{pool}}$ , the model gives realistic growth rates  $0 \text{ h}^{-1}$  to  $0.8 \text{ h}^{-1}$  (figure 1.5c). In this region, growth rate increases linearly as  $e_{\text{pool}}$  increases. With higher  $e'_{\text{pool}}$  values — that is, less of a constraint on the enzyme pool — the  $\tau_{\text{seq/par}}$  ratio increases, thus indicating that sequential biosynthesis gives less of an advantage (figure 1.5a). As  $e'_{\text{pool}}$  increases, ablated growth rates  $\lambda_{\text{seq},i}$  increases linearly at low  $e'_{\text{pool}}$  (figure 1.5e). But then, at higher  $e'_{\text{pool}}$ , these linear relationships decrease in gradient, all going to a plateau at very high  $e'_{\text{pool}}$  (figure 1.5f). This behaviour shows that the relationship between  $\lambda_{\text{seq},i}$  and  $e'_{\text{pool}}$  is independent of the growth rate  $\lambda_0$  and of each other. The  $\lambda_{\text{seq},i}$  of different components plateau at different  $e'_{\text{pool}}$  sizes, with carbohydrate reaching a plateau first, followed by protein. This may indicate that the enzyme-available proteome pool is limiting for these components, which may be because the cell needs more enzyme mass to catalyse the reactions needed for the synthesis of these components.

To explain the increase in  $\tau_{\text{seq/par}}$  as  $e'_{\text{pool}}$  increases, I consider the behaviour of  $\lambda_0$  and  $\lambda_{\text{seq},i}$  values with respect to  $e'_{\text{pool}}$  in intervals. As a simplification, I express the size of  $e'_{\text{pool}}$  in multiples of the original  $e_{\text{pool}}$  and denote the size of  $e'_{\text{pool}}$  as  $x$ .

Let  $\tau_{\text{seq/par}}$ , given by equation 1.15, depend on  $x$ :

$$\tau_{\text{seq/par}}(x) = \left( \sum_i \frac{f_i}{\lambda_{\text{seq},i}(x)} \right) \cdot \frac{\lambda_0(x)}{f_{\text{protein}}} \quad (1.18)$$

This expression takes into account how  $\lambda_0$  and  $\lambda_{\text{seq},i}$  values vary with  $x$ , and how  $f_i$  values are constants.

We thus obtain:

$$\begin{aligned}
 \frac{d\tau_{\text{seq/par}}(x)}{dx} &= \frac{1}{f_{\text{protein}}} \frac{d}{dx} \left[ \left( \sum_i \frac{f_i}{\lambda_{\text{seq},i}(x)} \right) \cdot \lambda_0(x) \right] \\
 &= \frac{1}{f_{\text{protein}}} \left[ \left( \sum_i \frac{f_i}{\lambda_{\text{seq},i}(x)} \right) \cdot \frac{d\lambda_0(x)}{dx} + \lambda_0(x) \frac{d}{dx} \left( \sum_i \frac{f_i}{\lambda_{\text{seq},i}(x)} \right) \right] \\
 &= \frac{1}{f_{\text{protein}}} \left[ \left( \sum_i \frac{f_i}{\lambda_{\text{seq},i}(x)} \right) \cdot \frac{d\lambda_0(x)}{dx} - \lambda_0(x) \sum_i \left( \frac{f_i}{\lambda_{\text{seq},i}(x)^2} \cdot \frac{d\lambda_{\text{seq},i}(x)}{dx} \right) \right]
 \end{aligned} \tag{1.19}$$

Reference to Fig 1.5.

make clear this is observed from the simulation.

Consider  $0 \leq x \leq 0.5$ . In this region of  $x$ , let  $\lambda_0 = k_0x$  and  $\lambda_{\text{seq},i} = k_i x$ , where constants  $k_0, k_i > 0$ . This models how these values initially increase linearly in figure 1.5. Equation 1.19 thus becomes:

$$\begin{aligned}
 \frac{d\tau_{\text{seq/par}}(x)}{dx} &= \frac{1}{f_{\text{protein}}} \left[ \left( \sum_i \frac{f_i}{k_i x} \right) \cdot k_0 - k_0 x \sum_i \left( \frac{f_i}{(k_i x)^2} \cdot k_i \right) \right] \\
 &= \frac{1}{f_{\text{protein}}} \left[ \frac{k_0}{x} \sum_i \frac{f_i}{k_i} - k_0 x \left( \sum_i \frac{f_i}{k_i x^2} \right) \right] \\
 &= \frac{1}{f_{\text{protein}}} \left[ \frac{k_0}{x} \sum_i \frac{f_i}{k_i} - \frac{k_0}{x} \sum_i \frac{f_i}{k_i} \right] \\
 &= 0
 \end{aligned} \tag{1.20}$$

And this explains the constant  $\tau_{\text{seq/par}}$  in this region.

To move on Appendix 2

Now, consider  $0.5 < x \leq 9$ . In this region, the trajectories of  $\lambda_{\text{seq},i}$  with respect to time remain linear, but some with changes in slope. In other words, in a sub-region where the slopes of all  $\lambda_{\text{seq},i}$  are constant, we can let:  $\lambda_0 = k_0x$  and  $\lambda_{\text{seq},i} = m_i x + c_i$ , where  $k_0, m_i, c_i > 0$ . Equation 1.19 thus becomes:

$$\begin{aligned}\frac{d\tau_{\text{seq/par}}(x)}{dx} &= \frac{1}{f_{\text{protein}}} \left[ \left( \sum_i \frac{f_i}{m_i x + c_i} \right) \cdot k_0 - k_0 x \sum_i \left( \frac{f_i}{(m_i x + c_i)^2} \cdot m_i \right) \right] \\ &= \frac{k_0}{f_{\text{protein}}} \left[ \left( \sum_i \frac{f_i}{m_i x + c_i} \right) - x \left( \sum_i \frac{f_i m_i}{(m_i x + c_i)^2} \right) \right] \\ &= \frac{k_0}{f_{\text{protein}}} \sum_i \left[ \frac{f_i}{m_i x + c_i} - \frac{x f_i m_i}{(m_i x + c_i)^2} \right] \\ &= \frac{k_0}{f_{\text{protein}}} \sum_i \left[ \frac{f_i c_i}{(m_i x + c_i)^2} \right]\end{aligned}\tag{1.21}$$

As  $f_i, c_i, m_i > 0$  for all biomass components  $i$ , and  $k_0 > 0$ , we get  $\frac{d\tau_{\text{seq/par}}(x)}{dx} > 0$  regardless of the values that these constants take. Because  $k_0$  does not change over the region of  $x$  considered,  $m_i$ ,  $c_i$ , and  $x$  values thus determine the magnitude of  $\frac{d\tau_{\text{seq/par}}(x)}{dx}$ . If within a region of  $x$ ,  $m_i$  and  $c_i$  values remain constant for all  $i$ , then as  $x$  increases,  $\frac{d\tau_{\text{seq/par}}(x)}{dx}$  should decrease — this is certainly the case, as can be observed in figure 1.5.

Lastly, consider  $x > 9$ . In this region,  $\lambda_0$  becomes constant, thus we let  $\lambda_0 = k_0$ . We keep  $\lambda_{\text{seq},i} = m_i x + c_i$ , and as before,  $k_0, m_i, c_i > 0$ . Equation 1.19 thus becomes:

$$\begin{aligned}\frac{d\tau_{\text{seq/par}}(x)}{dx} &= \frac{1}{f_{\text{protein}}} \left[ 0 - k_0 \sum_i \left( \frac{f_i}{(m_i x + c_i)^2} \cdot m_i \right) \right] \\ &= -\frac{k_0}{f_{\text{protein}}} \sum_i \left[ \frac{f_i m_i}{(m_i x + c_i)^2} \right]\end{aligned}\tag{1.22}$$

This predicts decreasing  $\tau_{\text{seq/par}}$  as  $x$  increases in this region. Because  $k_0$  is constant in this region, the rate of this decrease is thus controlled by  $m_i$  and  $c_i$  values. As each  $\lambda_{\text{seq},i}$  trajectory becomes flat as  $x$  increases, each  $\frac{f_i m_i}{(m_i x + c_i)^2}$  term becomes zero, thus shrinking the magnitude of  $\frac{d \tau_{\text{seq/par}}(x)}{dx}$ . Finally, as all  $\lambda_{\text{seq},i}$  trajectories become flat at  $x > 15$ ,  $\frac{d \tau_{\text{seq/par}}(x)}{dx} = 0$ .

## 1.7 Effect of carbon and nitrogen sources

### 1.7.1 Exchange reaction saturation of

Genome-scale metabolic models typically include nutrient exchange reactions to simulate the presence of certain nutrients in the growth medium. Because FBA does not account for substrate concentrations, studying the effect of nutrient sources using FBA involves constraining the flux of exchange reactions so that simulated growth rate match experimental observations, as performed by Elsemman et al. (2022) to study the effect of glucose.) only by them?

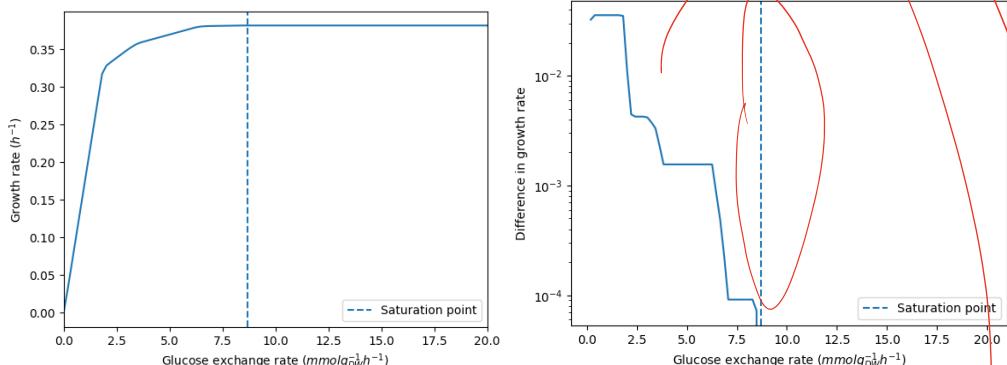
Using the ecYeast8.6.0 model, I varied the flux bounds of some exchange reactions and recorded how they affected simulated growth rate. I did this with two carbon sources, glucose and pyruvate, and with ammonium as a nitrogen source (figure 1.6). Define/explain?

The growth saturation curve for glucose (figure 1.6a) is similar to that simulated by Elsemman et al. (2022) using another derivative of the Yeast8 model. The saturation point is well below the maximal glucose consumption rate of 16 mmol g<sub>DW</sub><sup>-1</sup> h<sup>-1</sup> to 19 mmol g<sub>DW</sub><sup>-1</sup> h<sup>-1</sup> as determined by GC-MS-based metabolic flux ratio analysis (Blank and Sauer, 2004). In addition, the saturation curves show that the maximum growth rate on glucose is 0.38 h<sup>-1</sup> while the maximum growth rate on pyruvate is 0.25 h<sup>-1</sup> (figure 1.6c). The maximum growth rate on

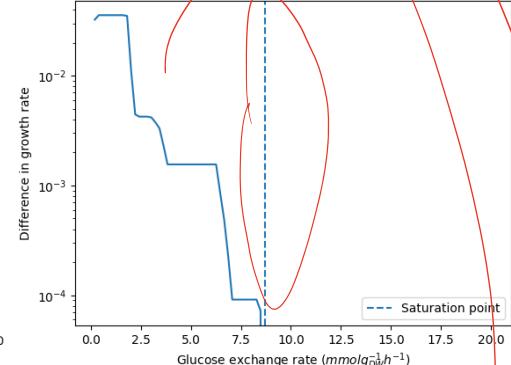
conventional

motivate why  
you are talking  
about this.

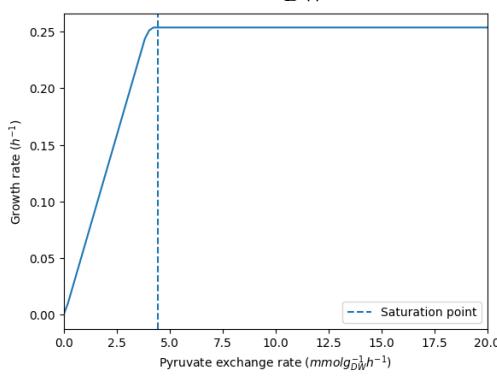
## 1.7. Effect of carbon and nitrogen sources



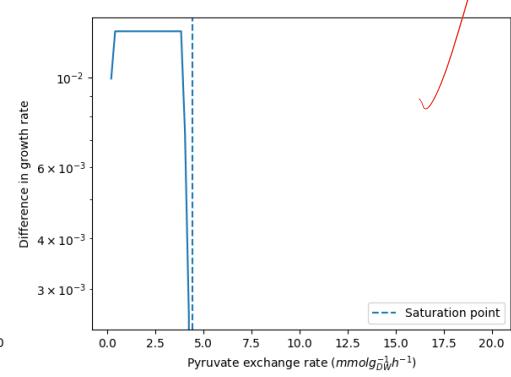
(a) Effect of glucose exchange on growth rate, with ammonium exchange flux unrestricted. Growth rate saturation is at  $8.69 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$ .



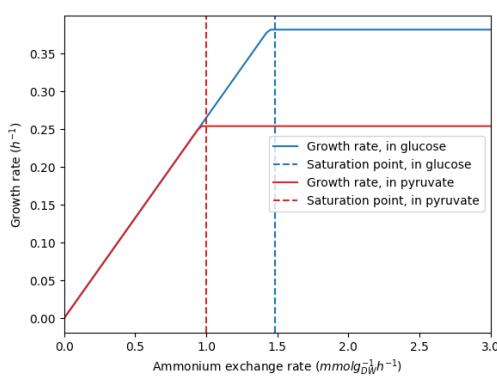
(b)



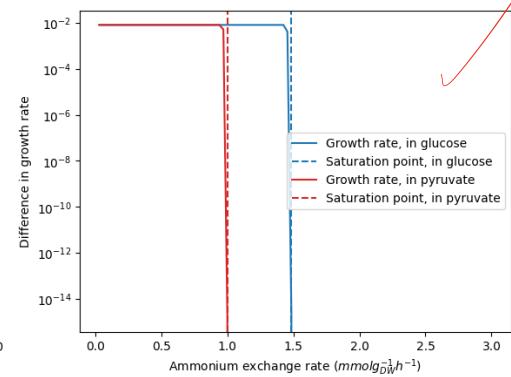
(c) Effect of pyruvate exchange on growth rate, with ammonium exchange flux unrestricted. Growth rate saturation is at  $4.44 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$ .



(d)



(e) Effect of ammonium exchange on growth rate, with exchanges of carbon sources set to growth rate saturation based on figures 1.6a and 1.6c. Growth rate saturation is at  $1.48 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$  in glucose, and at  $1.00 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$  in pyruvate.



(f)

**Figure 1.6:** Effect of nutrient exchange reactions on growth rate, predicted by the ecYeast8 model (left panels). Saturation values were found by computing differences of growth rates with respect to exchange rates (right panels 1.6b, 1.6d, 1.6f), and saturation was termed as the exchange rate at which the differences dropped below a small tolerance value  $\epsilon = 1 \times 10^{-8}$ .

glucose agrees with Domenzain et al. (2022), which used GECKO 2 to create the ecYeast7 model to predict maximum growth rates on various carbon sources. This paper did not predict growth rate on pyruvate, but a lower maximum growth rate on the non-fermentable carbon source compared to glucose agrees with the biochemical basis and is also consistent with my experimental observations. Finally, the growth rate saturation point for ammonium is determined by the maximal growth rate on the carbon source (figure 1.6e). For my subsequent investigation of the effect of carbon and nitrogen sources on the model, I use the exchange rates found in this investigation as saturation values.

## 1.7.2 Effect of carbon and nitrogen sources on biomass synthesis strategies

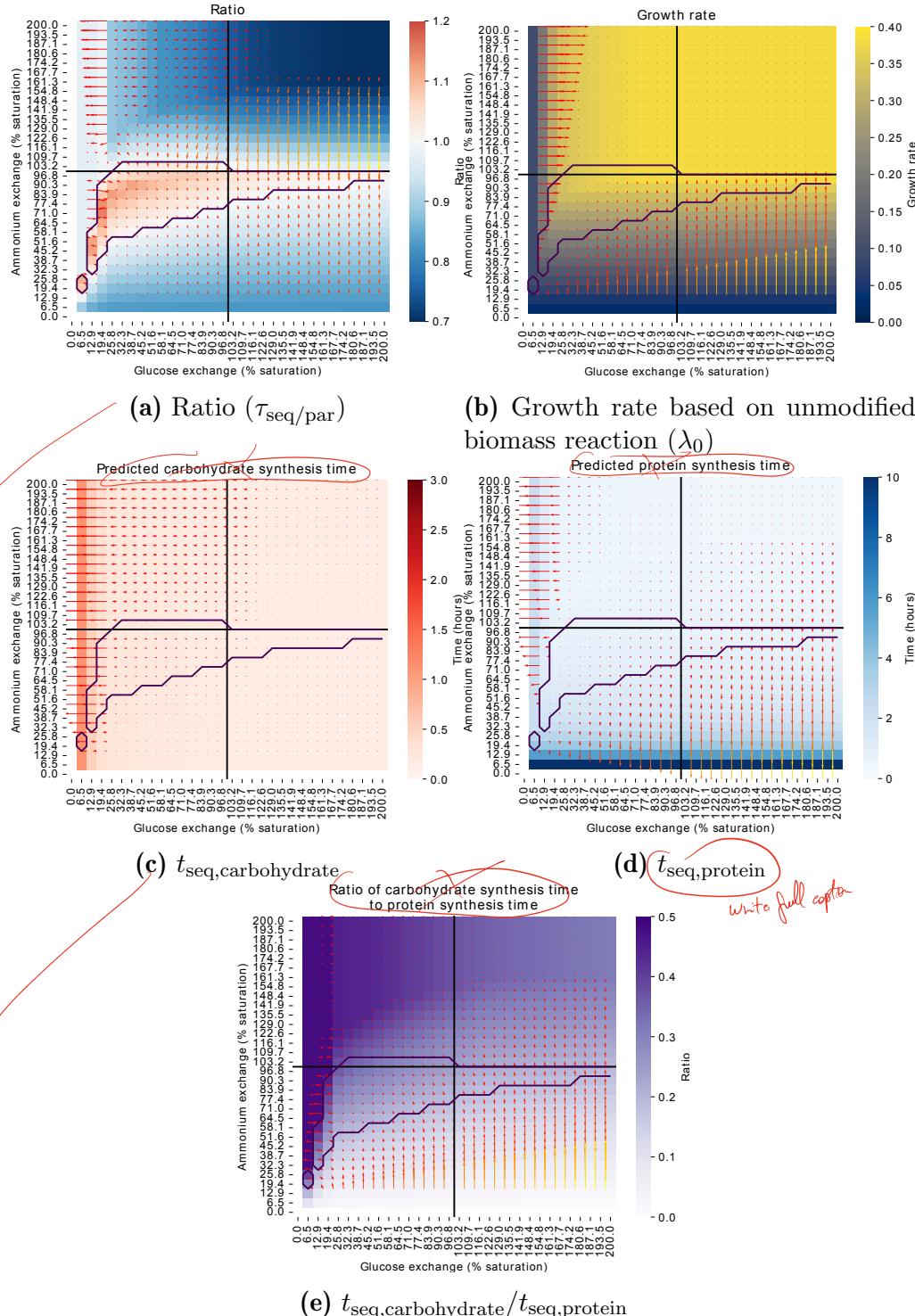
To assess the effect of carbon and nitrogen sources on biomass synthesis strategies,

I first varied the upper flux bounds of exchange reactions that corresponded to glucose exchange (denoted  $R_{\text{glucose}}$ , or  $R_{\text{glc}}$ ) and to ammonium exchange rate (denoted  $R_{\text{ammonium}}$ , or  $R_{\text{ammonium}}$ ). Each of these bounds took 32 equally-spaced values from 0% to 200% saturation, with glucose saturation being at  $8.69 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$  and ammonium saturation being at  $1.48 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$ , to produce a grid of 1024 conditions. At each combination of glucose and ammonium exchange, I ablated components in the biomass reaction according to section 1.5 to obtain  $\tau_{\text{seq/par}}$ ,  $\lambda_0$ ,  $t_{\text{seq,carbohydrate}}$ , and  $t_{\text{seq,protein}}$ .  $\tau_{\text{seq/par}}$  indicated whether the sequential or parallel strategy is advantageous for each condition, while  $\lambda_0$  indicated which nutrient source was limiting in each condition.

has?

. too detailed  
. for caption

which one



**Figure 1.7:** Effect of glucose ( $R_{\text{glucose}}$ ) and ammonium exchange rates ( $R_{\text{ammonium}}$ ) on various quantities. Exchange rates are expressed in percentages of growth saturation values shown in <sup>1.6</sup>: glucose saturation being at  $8.69 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$  and ammonium saturation being at  $1.48 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$ . Black straight lines indicate saturation values. Contours show regions in which  $\tau_{\text{seq/par}} > 1$ . Arrows indicate susceptibility of the quantity displayed in the heatmap, relative to  $R_{\text{glucose}}$  and  $R_{\text{ammonium}}$ .

*explain here*

Synthesis times  $t_{\text{seq},i}$  were computed to assess whether the ratio between the synthesis times differed in different conditions, in particular, conditions in which parallel biomass synthesis is favoured ( $\tau_{\text{seq}/\text{par}} > 1$ ). Specifically, I chose carbohydrate and protein synthesis times because:

1. Of all biomass components, predicted synthesis times of these biomass components varied the most as the glucose and ammonium exchange rates were varied.
2. Each accounts for a large proportion of biomass: protein accounts for 52.5% and carbohydrate accounts for 36.4% (table 1.2).
3. Carbohydrate synthesis has a clear biochemical relationship with the level of a carbon source. And, because amino acids contain amino groups, protein synthesis has a biochemical relationship with ammonium as a nitrogen source.

And thus the ratio  $\frac{t_{\text{seq,carbohydrate}}}{t_{\text{seq,protein}}}$  serves as the principal measure to quantify how  $t_{\text{seq},i}$  values change as carbon and nitrogen source concentrations change.

To determine whether the carbon or nitrogen source is limiting for each of the quantities calculated, I computed the susceptibility of the quantity with respect to each axis, carbon source or nitrogen source.

*To do X, I defined ...*  
Susceptibility at each nutrient condition ( $R_{\text{glc}}, R_{\text{amm}}$ ), with respect to each axis  $i$  is given by:

$$s_i(R_{\text{glc}}, R_{\text{amm}}) = \frac{R_i}{y(R_{\text{glc}}, R_{\text{amm}})} \cdot \frac{\partial y(R_{\text{glc}}, R_{\text{amm}})}{\partial R_i} \quad (1.23)$$

where:

- $i$  indicates the axis: glucose (glc) or ammonium (amm)

→ First tell them what you did  
 → Then how you got there and the decisions you made along the way  
 → At the moment it's written like a log book in chronological order

- $R_i$  indicates the exchange rate, glucose ( $R_{\text{glc}}$ ) or ammonium ( $R_{\text{amm}}$ ). An  $(R_{\text{glc}}, R_{\text{amm}})$  pair defines a nutrient condition.
- $y(R_{\text{glc}}, R_{\text{amm}})$  represents the quantity of interest  $\tau_{\text{seq/par}}, \lambda_0, t_{\text{seq,carbohydrate}}$ ,  $t_{\text{seq,protein}}$ , or  $\frac{t_{\text{seq,carbohydrate}}}{t_{\text{seq,protein}}}$  at each nutrient condition

For computational use, the values of  $R_i$  are equally-spaced discrete values from a vector with an interval  $\Delta R_i$ . The differential  $\frac{\partial y(R_{\text{glc}}, R_{\text{amm}})}{\partial R_i}$  is thus approximated by central differences:

$$\begin{aligned} \frac{\partial y(R_{\text{glc}}, R_{\text{amm}})}{\partial R_{\text{glc}}} &\approx \frac{y(R_{\text{glc}} + \Delta R_{\text{glc}}, R_{\text{amm}}) - y(R_{\text{glc}} - \Delta R_{\text{glc}}, R_{\text{amm}})}{2\Delta R_{\text{glc}}} \\ \frac{\partial y(R_{\text{glc}}, R_{\text{amm}})}{\partial R_{\text{amm}}} &\approx \frac{y(R_{\text{glc}}, R_{\text{amm}} + \Delta R_{\text{amm}}) - y(R_{\text{glc}}, R_{\text{amm}} - \Delta R_{\text{amm}})}{2\Delta R_{\text{amm}}} \end{aligned} \quad (1.24)$$

*not needed*

Susceptibility gives an advantage over simply computing the differential of the quantity of interest with respect to the exchange rate, i.e.  $\frac{\partial y}{\partial R_i}$ , because it accounts for the changing magnitude of  $R_i$ . At greater values of  $R_i$ , the susceptibility is decreased by a greater value with respect to  $\frac{\partial y}{\partial R_i}$ . This accounts for how at such high values of  $R_i$ , the proportional change in  $R_i$  as it is increased or decreased by one step  $\Delta R_i$  is smaller. It also accounts for the different step sizes of  $R_{\text{glc}}$  and  $R_{\text{amm}}$ , owing to the different saturation concentrations of the two nutrients.

At a specific nutrient condition  $(R_{\text{glc}}, R_{\text{amm}})$ , if  $s_{\text{glc}}(R_{\text{glc}}, R_{\text{amm}}) > s_{\text{amm}}(R_{\text{glc}}, R_{\text{amm}})$  for a quantity of interest  $y$ , then it means that the quantity is more susceptible to — or more limited by — glucose exchange at this condition. The reverse is true if  $s_{\text{glc}}(R_{\text{glc}}, R_{\text{amm}}) < s_{\text{amm}}(R_{\text{glc}}, R_{\text{amm}})$ . To visualise the extent to which each exchange rate limits each quantity of interest, I show the susceptibility values as arrows (figures 1.7 and 1.8) with  $s_{\text{glc}}$  and  $s_{\text{amm}}$  as two components of the

*susceptibility = sensitivity?*  
*↳ more intuitive*

*Don't understand.  
Do you need to explain this?*

vector that defines these arrows. This way, horizontal arrows indicate a strong susceptibility to glucose and vertical arrows indicate a strong susceptibility to ammonium, while the arrows point towards increasing values of the quantity of interest.

*"we see two regions of interest..."*

Figures 1.7a and 1.7b show that the region where  $\tau_{\text{seq/par}} > 1$  corresponds to a region around the boundary between the region where glucose more limits growth rate and the region where ammonium more limits growth rate, plus a region where ammonium exchange is a saturation while glucose exchange is over saturation. In contrast, when the growth rate is near its maximum, where neither glucose nor ammonium is limiting, when glucose or ammonium exchange increases, sequential biosynthesis becomes more advantageous.

*unclear*  
*good sentence structure ✓*

To explain why  $\tau_{\text{seq/par}} > 1$  when ammonium exchange is under saturation and glucose is over saturation, consider equation 1.15. If we assume that  $\frac{f_i}{\lambda_{\text{seq},i}}$  for  $i$  other than carbohydrate and protein changes negligibly — justified by the small  $f_i$  values for these other biomass components — we write:

$$\tau_{\text{seq/par}} \approx \left( k + \frac{f_{\text{protein}}}{\lambda_{\text{seq,protein}}} + \frac{f_{\text{carbohydrate}}}{\lambda_{\text{seq,carbohydrate}}} \right) \cdot \frac{\lambda_0}{f_{\text{protein}}} \quad (1.25)$$

where  $k$  is a constant.

In the glucose-limiting region, as  $R_{\text{glc}}$  increases, under saturation,

- $\lambda_0$  increases,
- $\lambda_{\text{seq,carbohydrate}}$  increases, and
- $\lambda_{\text{seq,protein}}$  increases, while
- $\tau_{\text{seq/par}}$  stays constant.

In the nitrogen-limiting region, as  $R_{\text{amm}}$  increases, under saturation,

- $\lambda_0$  increases,

*Not needed.*

- $\lambda_{\text{seq,carbohydrate}}$  stays constant, and
- $\lambda_{\text{seq,protein}}$  increases, while
- $\tau_{\text{seq/par}}$  increases.

Based on the glucose-limiting region, the increases in  $\lambda_{\text{seq,carbohydrate}}$  and  $\lambda_{\text{seq,protein}}$  in the carbon-limiting case are just enough to balance the increase in  $\lambda_0$ . Then, in the nitrogen-limiting case, if there is no increase in  $\lambda_{\text{seq,carbohydrate}}$ , then the combined effect of  $\lambda_{\text{seq,carbohydrate}}$  and  $\lambda_{\text{seq,protein}}$  no longer balance the increase in  $\lambda_0$ . This thus explains why  $\tau_{\text{seq/par}}$  increases in this case.

If both nutrients are limiting, as  $(R_{\text{glc}}, R_{\text{amm}})$  varies along the curve from  $(0, 0)$  to  $(R_{\text{glc,saturation}}, R_{\text{amm,saturation}})$  that goes through such conditions, both  $\lambda_{\text{seq,carbohydrate}}$  and  $\lambda_{\text{seq,protein}}$  increase, explaining why  $\tau_{\text{seq/par}}$  remain roughly at the same high values greater than 1. Thus, the region where  $\tau_{\text{seq/par}} > 1$  where both glucose and ammonium are near-limiting can be seen as where the behaviour of the glucose-limiting and ammonium-limiting cases meet.

Next, I investigate  $\frac{t_{\text{seq,carbohydrate}}}{t_{\text{seq,protein}}}$ .

Based on equation 1.10,

? doubling  
this has not been mentioned

$$\frac{t_{\text{seq,carbohydrate}}}{t_{\text{seq,protein}}} = \frac{f_{\text{carbohydrate}}}{f_{\text{protein}}} \cdot \frac{\lambda_{\text{seq,protein}}}{\lambda_{\text{seq,carbohydrate}}} \quad (1.26)$$

In the glucose-limiting region,  $\frac{t_{\text{seq,carbohydrate}}}{t_{\text{seq,protein}}}$  remains at a high value because, as  $R_{\text{glc}}$  increases in this region, both  $\lambda_{\text{seq,carbohydrate}}$  and  $\lambda_{\text{seq,protein}}$  increase. However, in the ammonium-limited region,  $\frac{t_{\text{seq,carbohydrate}}}{t_{\text{seq,protein}}}$  increases as  $R_{\text{amm}}$  increases because then  $\lambda_{\text{seq,protein}}$  increases while  $\lambda_{\text{seq,carbohydrate}}$  remains constant. If both nutrients are limiting, as  $(R_{\text{glc}}, R_{\text{amm}})$  varies along the curve from  $(0, 0)$  to  $(R_{\text{glc,saturation}}, R_{\text{amm,saturation}})$  that goes through such conditions, both  $\lambda_{\text{seq,carbohydrate}}$  and  $\lambda_{\text{seq,protein}}$  increase, explaining why  $\frac{t_{\text{seq,carbohydrate}}}{t_{\text{seq,protein}}}$  remain roughly at the same.

Considering  $t_{\text{seq,carbohydrate}}$  and  $t_{\text{seq,protein}}$ , glucose exchange limits carbohydrate synthesis time for virtually every nutrient condition. In contrast, ammonium exchange limits protein synthesis time for most nutrient conditions, except for conditions in which glucose exchange is very low, owing to the fact that both carbon and nitrogen sources contribute to protein biosynthesis. Taken together with the discussion above, these effects of nutrient exchange on protein and carbohydrate synthesis times thus explain the patterns of  $\tau_{\text{seq/par}}$  and  $\frac{t_{\text{seq,carbohydrate}}}{t_{\text{seq,protein}}}$  as both exchange reactions change in flux.

unclear

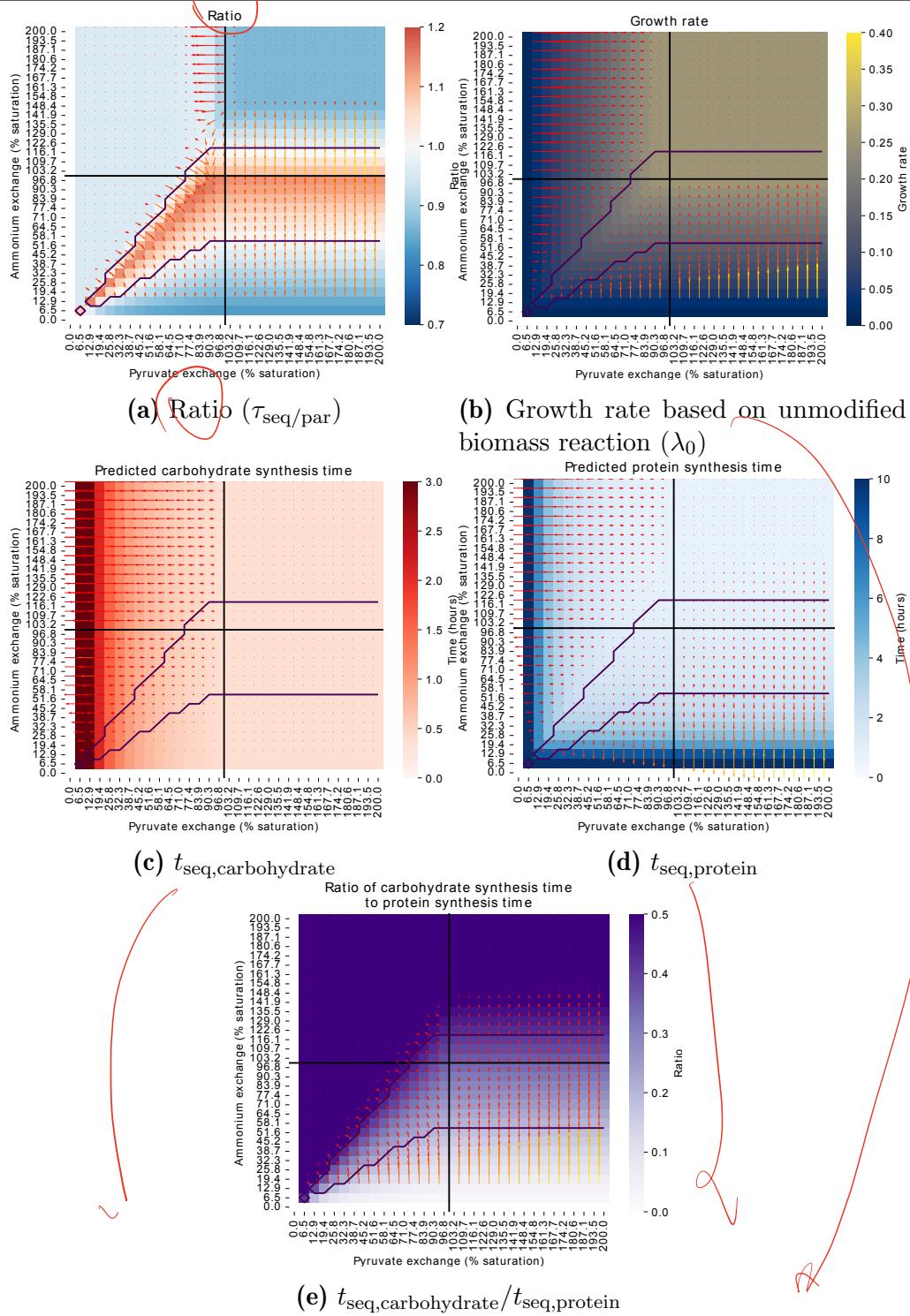
**Subsection** To investigate the effect on a non-fermentable carbon source, I repeated the investigation using pyruvate as the carbon source, and adjusted the saturation exchange rates accordingly: pyruvate saturation being at  $4.44 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$  and ammonium saturation being at  $1.00 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$  (figure 1.8). Overall, pyruvate seems to represent an extreme case of the glucose investigation. The previous discussion of how the quantities of interest vary as exchange rates vary applies. Though, the change of behaviour can be explained by how the growth saturation curve under pyruvate has a different shape from the growth saturation curve under glucose.

?  
?

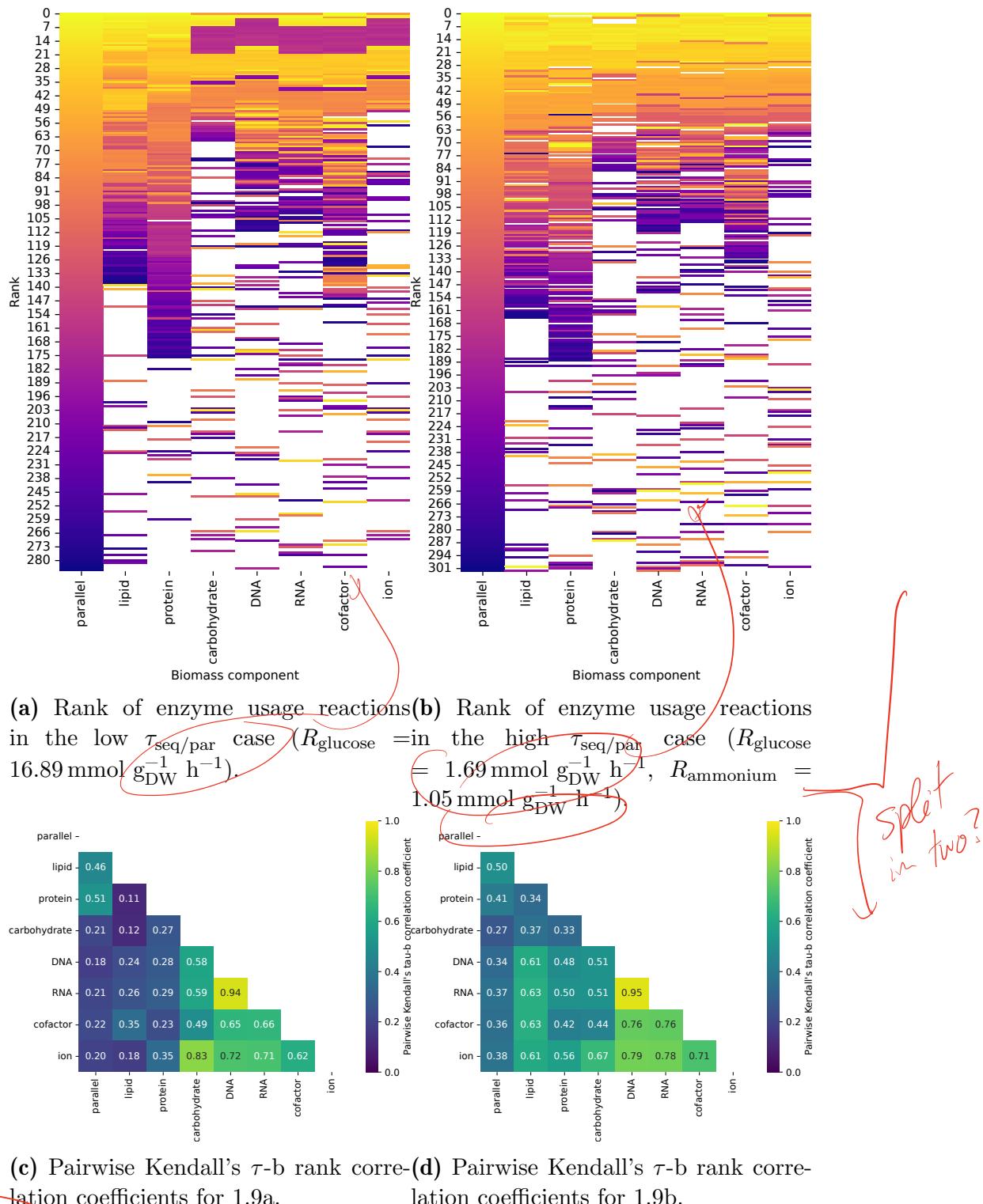
In both the glucose-ammonium and pyruvate-ammonium conditions, I observe that the highest  $\tau_{\text{seq/par}}$  occurs at the boundary at which both the carbon source and the nitrogen source are limiting. This leads to the hypothesis that the cell favours parallel biosynthesis of biomass components in such conditions, especially if the associated biomass components share metabolic pathways and if the conditions dictate similar levels of enzymes.

This is  
the main  
point.

Needs to stand out



**Figure 1.8:** Effect of pyruvate ( $R_{\text{pyruvate}}$ ) and ammonium exchange rates ( $R_{\text{ammonium}}$ ) on various quantities. Exchange rates are expressed in percentages of growth saturation values shown in 1.6: pyruvate saturation being at  $4.44 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$  and ammonium saturation being at  $1.00 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$ . Black straight lines indicate saturation values. Contours show regions in which  $\tau_{\text{seq}}/\text{par} > 1$ . Arrows indicate susceptibility of the quantity displayed in the heatmap, relative to  $R_{\text{pyruvate}}$  and  $R_{\text{ammonium}}$ .



**Figure 1.9:** Proteome allocation favouring sequential and parallel biosynthesis strategies, with glucose as carbon source. **(Top panels: 1.9a, 1.9b)** In the parallel case (no change to biomass equation), enzyme usage reactions are ranked, in descending order, by magnitude of flux and each assigned a colour. When biomass components are ablated, the ranks change, as shown by how the order of colours change from column to column. White indicates reactions that carry zero flux in the parallel case. **(Bottom panels: 1.9c, 1.9d)** Kendall's  $\tau$ -b rank correlation coefficient was computed for each pair of columns in the top panels to quantify the similarity between each case.

To evaluate this hypothesis, I investigated how the enzyme usage fluxes change across each round of ablation in the glucose-ammonium exchange combinations that produce the lowest and greatest  $\tau_{\text{seq/par}}$  (figure 1.9), and performed the same investigation for pyruvate-ammonium exchange combinations (figure 1.10). As a visualisation aid, I ranked the enzyme usage reactions by magnitude of flux to emphasise how the allocation of enzymes changes in each round. I also computed the Kendall's  $\tau$ -b rank correlation coefficient (KENDALL, 1945) to quantify how similar the enzyme usage flux vectors were between each pair of rounds.

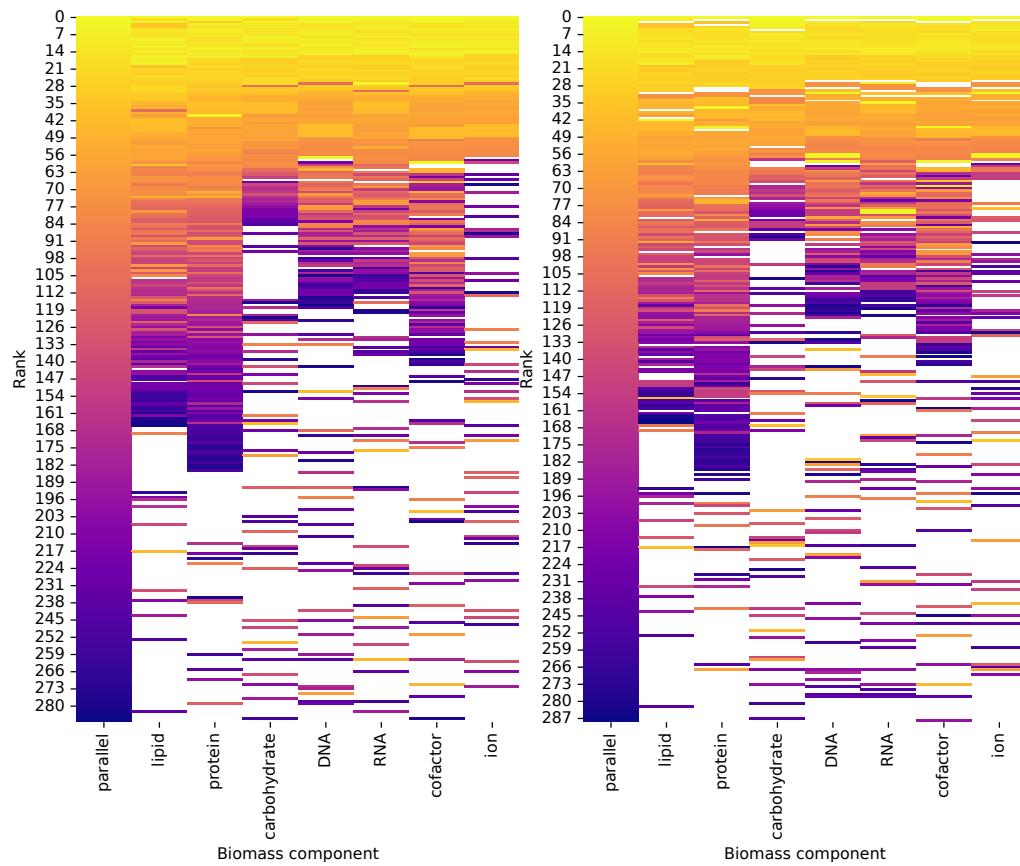
• Needs more structure  
• It's chronological

Figure 1.9 suggests that in the low  $\tau_{\text{seq/par}}$  condition, in which both glucose and ammonium are abundant, enzyme-available proteome allocation patterns for lipid- and protein-prioritised simulations most resemble the parallel case. In contrast, a set of enzymes that originally had low usage fluxes in the parallel case became enzymes with the highest levels when carbohydrates, DNA, RNA, cofactors, or ions are prioritised. This separation of biomass components into two groups according to enzyme-available proteome allocation patterns can explain how sequential biosynthesis is advantageous. At least, the lipids and proteins can be synthesised together, followed by the other components; the order by which these two groups are synthesised can be reversed. The high  $\tau_{\text{seq/par}}$  condition, in which both glucose and ammonium are near-limiting, gives the opposite picture. Here, allocation of the proteome to enzymes is similar across all situations, suggesting that parallel biosynthesis of biomass components may be advantageous in this situation.

Message

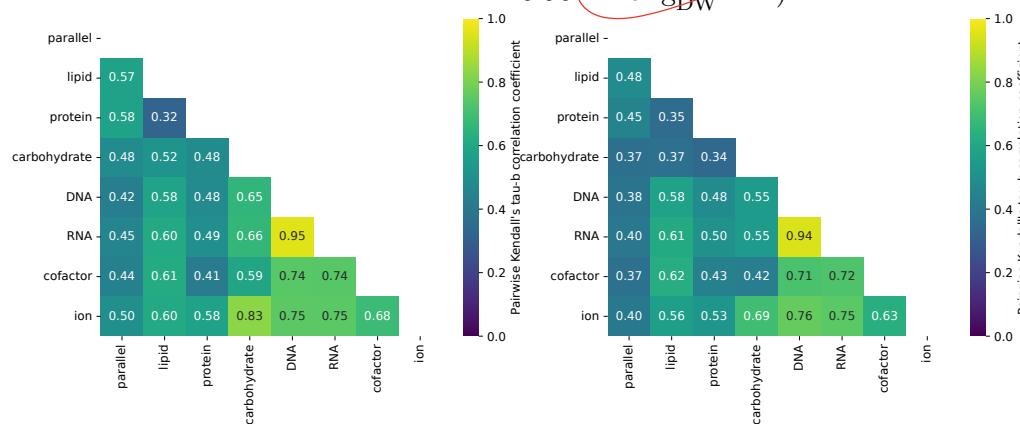
However, figure 1.10 suggests that the contrast between each biomass component is lessened when pyruvate is the carbon source. ... and?

Make it clear what / why / how  
(In this order)



(a) Rank of enzyme usage reactions in the low  $\tau_{\text{seq/par}}$  case ( $R_{\text{pyruvate}} = 8.89 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$ ).

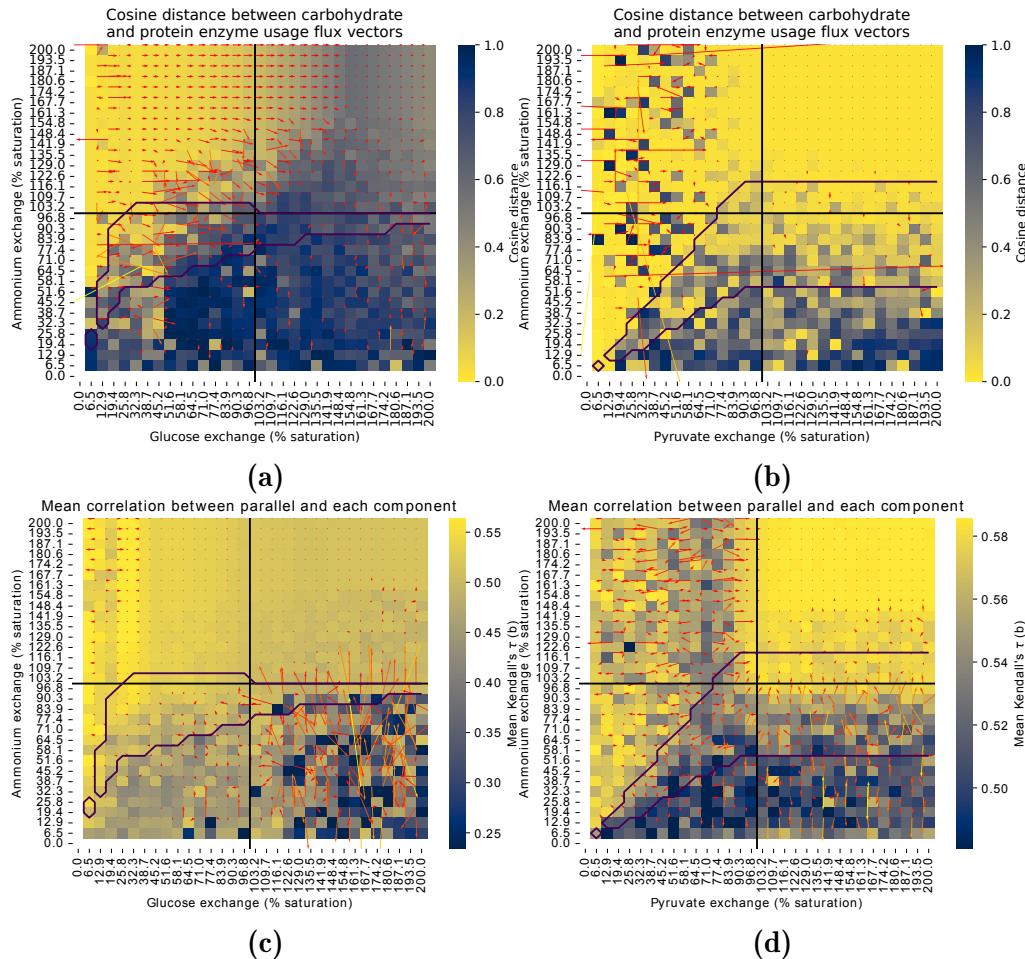
(b) Rank of enzyme usage reactions in the high  $\tau_{\text{seq/par}}$  case ( $R_{\text{pyruvate}} = 3.73 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$ ,  $R_{\text{ammonium}} = 0.90 \text{ mmol g}_{\text{DW}}^{-1} \text{ h}^{-1}$ ). Conversational



(c) Pairwise Kendall's  $\tau$ -b rank corre-  
lation coefficients for 1.10a.

(d) Pairwise Kendall's  $\tau$ -b rank corre-  
lation coefficients for 1.10b. Conversational

**Figure 1.10:** Proteome allocation favouring sequential and parallel biosynthesis strategies, with pyruvate as carbon source. **(Top panels: 1.10a, 1.10b)** In the parallel case (no change to biomass equation), enzyme usage reactions are ranked, in descending order, by magnitude of flux and each assigned a colour. When biomass components are ablated, the ranks change, as shown by how the order of colours change from column to column. White indicates reactions that carry zero flux in the parallel case. **(Bottom panels: 1.10c, 1.10d)** Kendall's  $\tau$ -b rank correlation coefficient was computed for each pair of columns in the top panels to quantify the similarity between each case.



**Figure 1.11:** Measures of similarity between enzyme usage fluxes in rounds of ablation, when glucose and ammonium exchange are varied (left panels), and when pyruvate and ammonium exchange are varied (right panels). **(Top panels)** Cosine distance between the vector of enzyme usage fluxes when protein is prioritised and the equivalent vector when carbohydrate is prioritised, computed for all nutrient conditions. **(Bottom panels)** For each nutrient condition, the Kendall's  $\tau$ - $b$  rank correlation coefficient between the parallel case and each biomass-prioritised case was computed and the mean was found. Higher mean correlations suggest an advantage to parallel biosynthesis.

Upon investigation of all 1024 nutrient conditions in each carbon source-nitrogen source pair, it appears that the relationship between exchange rate and any measure of similarity between ablation rounds that is based on enzyme usage reactions is not continuous (figure 1.11). This noise complicates drawing conclusions. Although, in the pyruvate-ammonium case, there is a general trend that suggests that proteome allocation when carbohydrates are prioritised and when proteins are prioritised are more similar in high-pyruvate high-ammonium conditions. There is also a trend that suggests, contrary to figure 1.8, that parallel biosynthesis is more similar to ablated cases in high-pyruvate high-ammonium conditions, but not by a high degree. On the contrary, in the glucose-ammonium case, there is a general trend that suggests that, at high ammonium exchange rates, increasing glucose exchange rates lead to a greater difference between the enzyme usage flux vectors when carbohydrate is prioritised and when protein is prioritised. In all cases, there seems to be no clear relationship between the distance measures and whether  $\tau_{\text{seq/par}} > 1$ .

The noise in figure 1.11 is likely explained by the multiplicity of solutions in FBA. This phenomenon means that FBA finds the optimal value of the objective function, such as growth, but the flux vector that gives rise to this objective function may differ across each simulation or each linear optimisation solver. This is not a problem when ablation-related quantities were computed earlier in the section because they were based on the flux of the objective function, defined as the biomass reaction, even if the biomass reaction itself was modified during ablation. In other words, although choosing representative nutrient conditions leads to an attractive picture that may confirm the hypothesis of this section, the computational limitations of FBA may render any conclusion flawed.

2  
THINK

## 1.8. Conclusion

the chapter

47

### 1.8 Conclusion

Needs work

first summary

Enzyme-constrained models like GECKO are useful. However, an appropriate method of restricting fluxes must be used, given the formalisms built in the model.

I found that / Results suggest

Ablation of components of the biomass reaction gives changes to enzyme allocation that are biologically relevant and also gives a way to estimate timescale of biosynthesis. Such timescales give rise to a  $\tau_{\text{seq/par}}$  ratio that can be used to assess whether sequential or parallel biosynthesis of biomass components is favoured.

A restricted proteome-available protein pool, as long as the growth rate remains realistic, makes sequential biosynthesis more advantageous. In addition, the advantage is retained in auxotrophs and deletion strains, confirming the robustness (presence) of sequential biomass synthesis as a resource allocation strategy, and thus may explain why the metabolic cycle is still present in such strains.

"Results show/suggest that"

Nitrogen source availability affects protein synthesis time in the sequential case, while carbon source availability affects both carbohydrate and protein synthesis times. Both additively promote (wild type) growth rate. These effects explain why parallel biosynthesis is advantageous when carbon and nitrogen sources are both near-limiting or when the nitrogen source is near saturation while the glucose source is over saturation. Such patterns are dependent on the growth saturation curves on each nutrient source.

When the cell prioritises biosynthesis of each biomass component, it allocates its proteome to enzymes differently, but the allocation is similar when the cell prioritises carbohydrates, DNA, RNA, cofactors, and ions. This could explain why parallel biosynthesis is advantageous in some conditions. However, the multiplicity of FBA solutions complicates analysis based of fluxes.

is this a result?

How the cell changes its allocation<sup>?</sup> of its proteome to subsystems may explain some experimental observations. It is possible that the cell synthesises carbohydrate, DNA, and RNA around the same time so that use of oxidative phosphorylation occur at around the same time. This may explain cycles in dissolved oxygen concentrations. In addition, lipid biosynthesis often requires different enzymes than the other components, and this may explain cycling of lipid stores. [ref]

The extent to which sequential biosynthesis is advantageous in different nutrient conditions and the synthesis timescales predicted may explain why yeast cells show metabolic cycles in certain nutrient conditions. The model may provide a weak explanation of why yeast cells continue to have metabolic cycles when abruptly starved of glucose: if the glucose exchange is near zero, the ratio is less than one. Though, a better investigation would account for switching between nutrient conditions, which the most basic forms of FBA is not built for — it only looks at steady state, and does not ‘remember’ past states.

The model does not account for varying protein fractions and other parameters that affect the size of  $e_{\text{pool}}$  during growth and division, and across different conditions. For instance, Elsemann et al. (2022) state that growth rate affects proteome fractions ( $f$ ) and the saturation factor ( $\sigma$ ). However, because I find that  $e_{\text{pool}}$  affects the growth rate, there may be a circular relationship, and there is no guarantee that parameter values will converge if I tune the  $e'_{\text{pool}}$  to obtain a  $\lambda_0$  that in turn affects  $e'_{\text{pool}}$ . Furthermore, the data on biomass component fractions are old, sparse, and coarse — as a back-of-the-envelope calculation, my investigation is perhaps sufficient.

---

In summary, my results show that the yeast cell may synthesise its biomass components in sequence or in parallel, subject to proteomic and nutrient availability constraints. Although it is unrealistic to assume that synthesis of one class macromolecule excludes all others, this approach is still instructive. It gives a back-of-the-envelope calculation to support the notion that the cell partitions biosynthesis temporally. It also gives weight to the idea that this may be one of the rationales of the existence of the yeast metabolic cycle.

---

---

## Appendix A

---

### **Insert appendix title here**

---

Insert content here.

---

## Bibliography

---

- Becker, S. A. and B. O. Palsson (2008). “Context-Specific Metabolic Networks Are Consistent with Experiments”. In: *PLOS Computational Biology* 4.5, e1000082.
- Blank, L. M. and U. Sauer (2004). “TCA Cycle Activity in *Saccharomyces Cerevisiae* Is a Function of the Environmentally Determined Specific Growth and Glucose Uptake Rates”. In: *Microbiology* 150.4, pp. 1085–1093.
- Canelas, A. B., C. Ras, A. ten Pierick, W. M. van Gulik and J. J. Heijnen (2011). “An in Vivo Data-Driven Framework for Classification and Quantification of Enzyme Kinetics and Determination of Apparent Thermodynamic Data”. In: *Metabolic Engineering* 13.3, pp. 294–306.
- Cesur, M. F., T. Çakır and P. Pir (2022). “Genome-Wide Analysis of Yeast Metabolic Cycle through Metabolic Network Models Reveals Superiority of Integrated ATAC-seq Data over RNA-seq Data”. In: *mSystems* 0.0 (), e01347–21.
- Chan, S. H. J., J. Cai, L. Wang, M. N. Simons-Senftle and C. D. Maranas (2017). “Standardizing Biomass Reactions and Ensuring Complete Mass Balance in Genome-Scale Metabolic Models”. In: *Bioinformatics* 33.22, pp. 3603–3609.
- Chang, A., L. Jeske, S. Ulbrich, J. Hofmann, J. Koblitz, I. Schomburg, M. Neumann-Schaal, D. Jahn and D. Schomburg (2021). “BRENDA, the ELIXIR Core Data Resource in 2021: New Developments and Updates”. In: *Nucleic Acids Research* 49.D1, pp. D498–D508.

- Dinh, H. V., D. Sarkar and C. D. Maranas (2022). “Quantifying the Propagation of Parametric Uncertainty on Flux Balance Analysis”. In: *Metabolic Engineering* 69, pp. 26–39.
- Domenzain, I., B. Sánchez, M. Anton, E. J. Kerkhoven, A. Millán-Oropeza, C. Henry, V. Siewers, J. P. Morrissey, N. Sonnenschein and J. Nielsen (2022). “Reconstruction of a Catalogue of Genome-Scale Metabolic Models with Enzymatic Constraints Using GECKO 2.0”. In: *Nature Communications* 13.1, p. 3766.
- Elsemman, I. E., A. Rodriguez Prado, P. Grigaitis, M. Garcia Albornoz, V. Harman, S. W. Holman, J. van Heerden, F. J. Bruggeman, M. M. M. Bisschops, N. Sonnenschein, S. Hubbard, R. Beynon, P. Daran-Lapujade, J. Nielsen and B. Teusink (2022). “Whole-Cell Modeling in Yeast Predicts Compartment-Specific Proteome Constraints That Drive Metabolic Strategies”. In: *Nature Communications* 13.1, p. 801.
- Heavner, B. D., K. Smallbone, B. Barker, P. Mendes and L. P. Walker (2012). “Yeast 5 – an Expanded Reconstruction of the *Saccharomyces Cerevisiae* Metabolic Network”. In: *BMC Systems Biology* 6.1, p. 55.
- Kanehisa, M., M. Furumichi, Y. Sato, M. Kawashima and M. Ishiguro-Watanabe (2023). “KEGG for Taxonomy-Based Analysis of Pathways and Genomes”. In: *Nucleic Acids Research* 51.D1, pp. D587–D592.
- KENDALL, M. G. (1945). “THE TREATMENT OF TIES IN RANKING PROBLEMS”. In: *Biometrika* 33.3, pp. 239–251.
- Lewis, N. E., K. K. Hixson, T. M. Conrad, J. A. Lerman, P. Charusanti, A. D. Polpitiya, J. N. Adkins, G. Schramm, S. O. Purvine, D. Lopez-Ferrer, K. K. Weitz, R. Eils, R. König, R. D. Smith and B. Ø. Palsson (2010). “Omic Data from Evolved *E. Coli* Are Consistent with Computed Optimal Growth from Genome-scale Models”. In: *Molecular Systems Biology* 6.1.

- Lu, H., F. Li, B. J. Sánchez, Z. Zhu, G. Li, I. Domenzain, S. Marciauskas, P. M. Anton, D. Lappa, C. Lieven, M. E. Beber, N. Sonnenschein, E. J. Kerkhoven and J. Nielsen (2019). “A Consensus *S. Cerevisiae* Metabolic Model Yeast8 and Its Ecosystem for Comprehensively Probing Cellular Metabolism”. In: *Nature Communications* 10.1, p. 3586.
- Mahadevan, R., J. S. Edwards and F. J. Doyle (2002). “Dynamic Flux Balance Analysis of Diauxic Growth in *Escherichia Coli*”. In: *Biophysical Journal* 83.3, pp. 1331–1340.
- Nilsson, A. and J. Nielsen (2016). “Metabolic Trade-offs in Yeast Are Caused by F1F0-ATP Synthase”. In: *Scientific Reports* 6.1, p. 22264.
- Norsigian, C. J., N. Pusarla, J. L. McConn, J. T. Yurkovich, A. Dräger, B. O. Palsson and Z. King (2020). “BiGG Models 2020: Multi-Strain Genome-Scale Models and Expansion across the Phylogenetic Tree”. In: *Nucleic Acids Research* 48.D1, pp. D402–D406.
- Orth, J. D., I. Thiele and B. Ø. Palsson (2010). “What Is Flux Balance Analysis?” In: *Nature Biotechnology* 28.3, pp. 245–248.
- Palsson, B. Ø. (2015). *Systems Biology: Constraint-based Reconstruction and Analysis*. Cambridge: Cambridge University Press.
- Reimers, A.-M., H. Knoop, A. Bockmayr and R. Steuer (2017). “Cellular Trade-Offs and Optimal Resource Allocation during Cyanobacterial Diurnal Growth”. In: *Proceedings of the National Academy of Sciences* 114.31, E6457–E6465.
- Sánchez, B. J., C. Zhang, A. Nilsson, P.-J. Lahtvee, E. J. Kerkhoven and J. Nielsen (2017). “Improving the Phenotype Predictions of a Yeast Genome-Scale Metabolic Model by Incorporating Enzymatic Constraints”. In: *Molecular Systems Biology* 13.8, p. 935.

- Sánchez, B. J., F. Li, E. J. Kerkhoven and J. Nielsen (2019). “SLIMER: Probing Flexibility of Lipid Metabolism in Yeast with an Improved Constraint-Based Modeling Framework”. In: *BMC Systems Biology* 13.1, p. 4.
- Sherman, F. (2002). “Getting Started with Yeast”. In: *Methods in Enzymology*. Ed. by C. Guthrie and G. R. Fink. Vol. 350. Guide to Yeast Genetics and Molecular and Cell Biology - Part B. Academic Press, pp. 3–41.
- Takhaveev, V., S. Özsezen, E. N. Smith, A. Zylstra, M. L. Chaillet, H. Chen, A. Papagiannakis, A. Milias-Argeitis and M. Heinemann (2023). “Temporal Segregation of Biosynthetic Processes Is Responsible for Metabolic Oscillations during the Budding Yeast Cell Cycle”. In: *Nature Metabolism* 5.2, pp. 294–313.
- The UniProt Consortium (2023). “UniProt: The Universal Protein Knowledgebase in 2023”. In: *Nucleic Acids Research* 51.D1, pp. D523–D531.
- Thiele, I. and B. Ø. Palsson (2010). “A Protocol for Generating a High-Quality Genome-Scale Metabolic Reconstruction”. In: *Nature Protocols* 5.1, pp. 93–121.
- Vijayakumar, S., P. K. S. M. Rahman and C. Angione (2020). “A Hybrid Flux Balance Analysis and Machine Learning Pipeline Elucidates Metabolic Adaptation in Cyanobacteria”. In: *iScience* 23.12, p. 101818.
- Weiße, A. Y., D. A. Oyarzún, V. Danos and P. S. Swain (2015). “Mechanistic Links between Cellular Trade-Offs, Gene Expression, and Growth”. In: *Proceedings of the National Academy of Sciences* 112.9, E1038–E1047.