Bayesian and evolutionary temperature dependent genome-scale models display alternative optima due to insufficient constraints

Jakob Peder Pettersen[1] **and Eivind Almaas**[1,2,*]

[1]Department of Biotechnology and Food Science, NTNU- Norwegian University of Science and Technology, Trondheim, Norway
[2]K.G. Jebsen Center for Genetic Epidemiology, Department of Public Health and General Practice, NTNU - Norwegian University of Science and Technology, Trondheim, Norway
[*]Corresponding author: eivind.almaas@ntnu.no

ABSTRACT

The metabolism of all living organisms is dependent on temperature, and therefore, having a good method to predict temperature effects at a systems level is of importance. Here, we investigate the properties and robustness of a recently developed computational framework for enzyme and temperature constrained genome-scale models (etcGEM). The approach predicts the temperature dependence of an organism's metabolic network from thermodynamic properties of the metabolic enzymes, markedly expanding the scope and applicability of constraint-based metabolic modelling. We show that the existing Bayesian algorithm for fitting an etcGEM converges under a range of different priors and random seeds, yet the solutions differ both in parameter space and on a phenotypic level. We quantified the phenotypic consequences by studying the impact of different solutions on six metabolic network signature reactions. While two of these reactions showed little phenotype variation between the solutions, the remainder displayed huge variation in flux-carrying capacity. Furthermore, we made software improvements to reduce the running time of the compute-intensive Bayesian algorithm by a factor of 8.5. In addition, we develop an evolutionary algorithm as an alternative to the Bayesian parameter fitting algorithm. Our analysis shows that it also gives decent convergence, but it yields solutions with lower variability.

Introduction

Temperature is a key effector of life, which is partially due to the effect which temperature has on the catalytic properties of enzymes. For a long time, it has been known that enzymatic reactions slow down at low temperatures, whereas high temperatures destroy enzymes, rendering them non-functional. In recent research[1,2], it has been acknowledged that enzymes also have lower catalytic rates at high temperatures due to changes in heat capacity. The effect of temperature on the behaviour of microorganisms as a whole is evident. Freezing food stops spoilage by inactivating microorganisms, whereas cooking kills the microorganisms. In between these temperature extremes, there there are observable effects which can be utilized commercially. One example of this is yeast production of aroma compounds which has been shown to depend on the temperature[3,4], a finding with potentially great impact on wine and beer brewing.

Until recently no attempt has been made to computationally explain the temperature dependence of microorganism by propagating the temperature dependence of metabolic enzymes to the entire metabolic network of an organism. However, Gang Li *et al.*[5] have come up with an extension of an enzyme constrained genome-scale metabolic model (ecGEM) which can capture the temperature dependence of metabolism. This model is thus called an enzyme and temperature constrained GEM (etcGEM). As most other models, the one by Li *et al.*[5] is based on sets of assumptions and parameters. In particular, this model is based on ecYeast7.6[6] (*Saccharomyces cerevisiae* strain S288C) and contains 764 metabolic enzymes and $2,292$ parameters associated with enzymatic activity. For each of these enzymes, the following parameters must be determined: (1) T_m, the melting temperature; (2) T_{opt}, the temperature optimum; and $\Delta C_p^{\ddagger}$, the change of heat capacity from the ground state to the transitional state.

Given these parameters, it was demonstrated how an enzyme's maximum catalytic rate $k_{cat}(T)$ at a certain temperature T could be estimated[5]. These temperature-dependent maximal catalytic rates were then fed into the enzyme-constrained genome-scale model[6,7], and the metabolic flux rates were predicted using Flux Balance Analysis (FBA)[8,9]. Furthermore, the Li and coworkers[5] used a Bayesian approach to infer the parameters from sets of training data: (1) Maximal growth rates in aerobic batch cultivations[10]; (2) Maximal growth rates in anaerobic batch cultivations[11]; Chemostat cultivations which include measurements of exchange fluxes of carbon dioxide, ethanol and glucose[12].

In the training data, the experimentally determined fluxes were recorded for a range of temperatures, thus generating a set of growth scenarios. The performance of a parameter set was assessed by predicting the flux rates in these growth scenarios and comparing these fluxes rates with the experimental results. Hence, the R^2 score between the experimental and modelled fluxes were used to assess the model's goodness of fit.

For initiating a Bayesian inference algorithm, prior parameters have to be chosen for the enzymes. Li *et al.*[5] did this through a custom heuristic which was partially based on measured temperature optima and denaturation temperatures for enzymes. By training the model with the Sequential Monte Carlo based Approximate Bayesian algorithm, a posterior distribution of parameter sets was found[5]. However, Li *et al.* did not investigate whether this posterior distribution was the only one matching the data, nor whether stochastic effects or their choice of priors had any affect on convergence and results. Thus, it is unclear how metabolic flux results from the etcGEM can be interpreted.

In this paper, we will investigate these properties both by running the Approximate Bayesian algorithm with multiple different random seeds and priors. Additionally, we explore parameter sets resulting from a very different algorithmic approach, that of an evolutionary algorithm, and we compare them with the posterior parameter distributions of the Approximate Bayesian algorithm. Finally, we have improved the execution time of the original software package by almost a factor 10, making it feasible to execute on smaller-scale computational infrastructures.

Results

Improvements to the running time of the algorithm

Running the Bayesian inference algorithm once for 500 iterations with the chosen parameters consumed approximately $17,000$ CPU hours (approximately corresponding to two weeks on a 48 core computer) using the framework from Li *et al.* Profiling showed that the particle evaluation procedure (see Methods) was the bottleneck and was caused by COBRApy's[13] internal routines to modify metabolic models prior to optimization. Hence, preparing the models for optimization consumed far more time than the optimization proper. For this reason, we modified the implementation to use the ReFramed package (`https://github.com/cdanielmachado/reframed`) for handling the genome-scale model. We benchmarked the two versions on a computer running Intel Core i7-8565U using a single core (Table 1). With our code improvements, the performance was boosted by factor of 8.5. As a consequence, the results of the Bayesian inference algorithm could be obtained the day after starting it when running on a computer server. Still, only one fifth of the particle evaluation time was spent on optimization, so improvements within the ReFramed package could increase performance even more.

Running the Bayesian inference algorithm on permuted priors

In order to investigate the effect of priors, we randomly permuted the prior parameters (see Methods for details) from Ref.[5], and we ran the Bayesian fitting algorithm on the three training datasets. We tried out in total three different permutations of the prior parameters in addition to the original priors. Also, we studied the results of stochastic effects by running two fitting simulations for *each* prior by using different random seeds, termed simulation 1 and 2. Thus in total, we ran the Bayesian fitting algorithm eight times, yielding eight different populations of solutions. The permuted priors yielded approximately the same rate of fitness convergence as the unpermuted priors (Figure 1 **A** and **B** and Supplementary Figure S1). Between simulation 1 and 2 for the same priors, the differences were negligible.

Having observed that all priors do indeed give good solutions after fitting, we next investigated whether these solutions were the same. For this, we created a Principal Component Analysis[14] (PCA) plot of the parameter sets obtained under fitting (Figure 1 **C** and **D** with a more complete overview in Supplementary Figure S2). The posterior distributions, defined as the collection of particles having $R^2 > 0.9$ were different even though the convergence properties were similar. This means that there are multiple parameter sets fitting the data equally well. Furthermore, stochastic effects caused by different random seeds resulted in convergence to different posterior distributions for simulation 1 and 2, even though the priors were identical.

Evolutionary algorithm

In order to further explore the fitness landscape of the enzyme parameters when mapped on to the genome-scale metabolic model, we constructed an evolutionary algorithm[15–17] as an alternative to the Approximate Bayesian algorithm (see Methods).

Table 1. Comparing benchmark results for a full evaluation of a particle using all three conditions (aerobic, anaerobic and chemostat). The numbers were averaged over 10 iterations.

Implementation	Overall time [s]	Percentage time in optimization [%]
Original version from Li *et al.*	106	0.837
Updated version with ReFramed	13.4	19.7

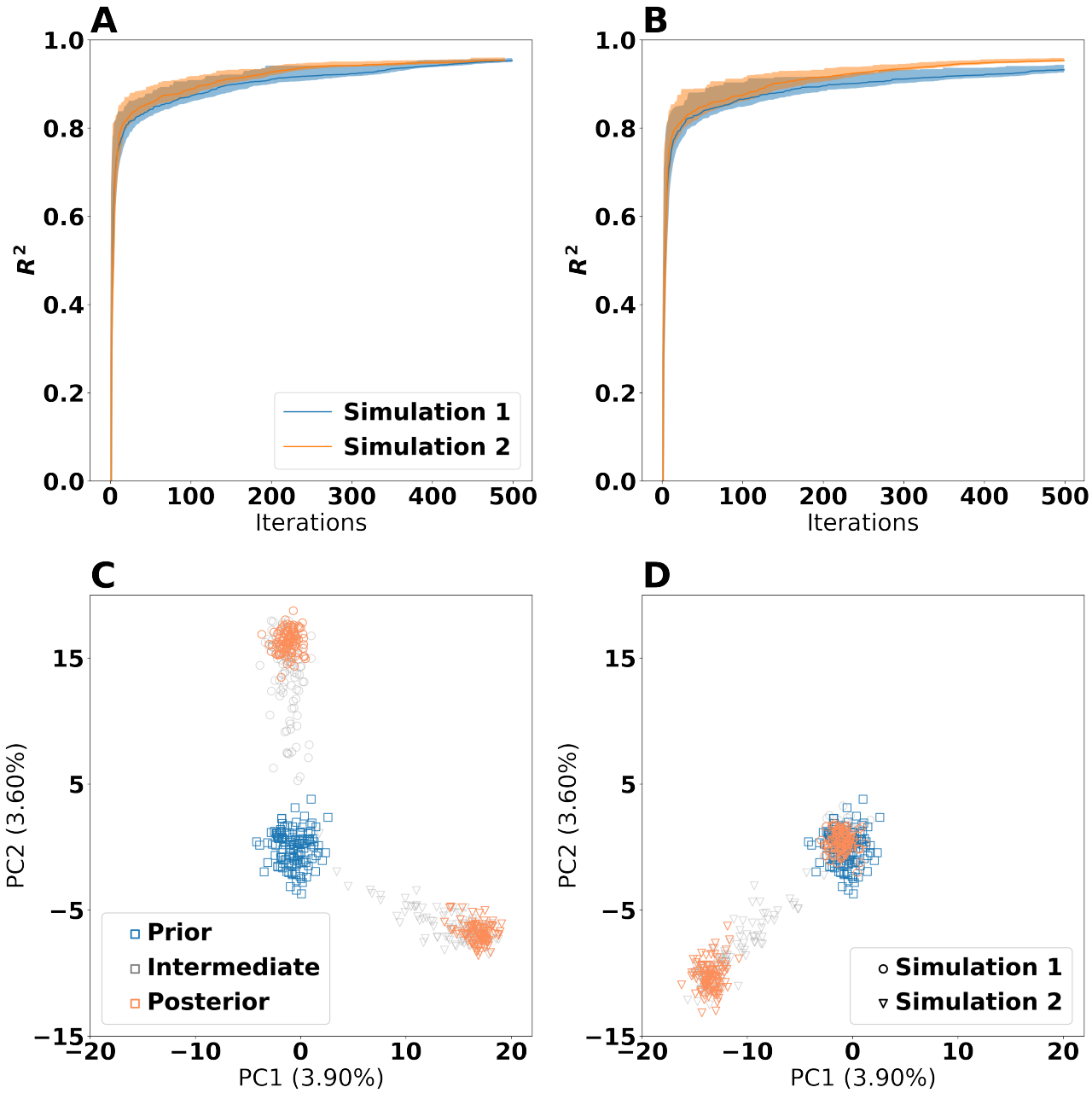

Figure 1. Comparison of the effect of prior and random seed on the Bayesian algorithm results. Panel **A** and **B** show the training R^2 values for the unpermuted priors and permuted prior 1, respectively. An R^2 value of 1 corresponds to exact correspondence between the training data and the model predictions. The shaded regions indicate the 5th and 95th percentiles, whereas the solid lines indicate the median (50th percentile). Panel **C** and **D** show Principal Component Analysis (PCA) plots of the parameter sets from the unpermuted priors and permuted prior 1, respectively. Each point is a candidate parameter set. The prior points are the ones which served as a starting point for the algorithm, the posterior points are the ones which had $R^2 > 0.9$, whereas all other points are intermediate points stemming from the simulations. The axes are identical for both panels and use the same ordination such that the panels are directly comparable. For clarity, random thinning of the points was applied to the posterior and intermediate points such that only 100 points are shown for each combination of panel, shape and colour.

We used the same priors as Li *et al.*[5] to initiate the evolutionary process, and we compared the results from the Bayesian fitting (unpermuted prior, simulation 1) and the evolutionary algorithm (see Figure 2). The fitness convergence properties are different for the two methods. As expected, the evolutionary algorithm improved fitness in leaps and bounds where it took some iterations before more fit individuals entered the population and out-competed the less fit individuals. At about 50 iterations, the evolutionary algorithm suffered from genetic bottlenecking and only slowly improved in fitness thereafter. On the other hand, the fitness from the Bayesian algorithm converged more steadily having a consistent span of fitness values. Even if the Bayesian algorithm lagged behind the evolutionary one at 100 generations, it slowly caught up, so that at the final generation (number 500), it had a median $R^2 = 0.9531$ compared to $R^2 = 0.9511$ for the evolutionary algorithm. Even though the fitness in the evolutionary algorithm converged, the evolution population in parameter space did not (Fig. 2 panel **B**). Instead, it continuously drifted away from the priors. We also attempted t-SNE[18] ordination of the particles from the Bayesian and evolutionary approach to detect higher-dimensional clustering (Supplementary Figure S3), but we were unable to find any reasonable clustering of the results beyond what was observed in the PCA ordination. Thus, the particles are scattered in parameter space.

Finally, we looked into the normalized Root Mean Squared Distance (RMSD) between each particle in the posterior distributions and population centroid (Figure 3), see Methods for details. The evolutionary algorithm consistently showed lower RMSD than the Bayesian algorithm, meaning that the evolutionary algorithm produced posterior distribution points which were closer to the centroid than the Bayesian algorithm. The distribution of RMSD values for the Bayesian algorithm was not influenced by the choice of prior or random seed to any large degree.

Differences on enzyme level

Since the particles appear to be spread in parameter space, it is interesting to investigate whether the parameter variation causes observable differences in simulated properties of the genome-scale metabolic model. To systematically study the phenotypic behaviour of posterior distribution particles, we performed Flux Variability Analysis (FVA)[19], see Methods for more details. While it is impossible to lock down a specific flux distribution due to an infinite number of alternative optimal solutions, FVA uncovers the flux bounds of each individual reaction that are capable of supporting optimal metabolic behavior. We decided to focus the investigation of FVA results on six reactions that have important biochemical roles in the metabolic network:

- Pyruvate dehydrogenase: A key reaction in connecting glycolysis to the TCA cycle and fatty acid synthesis
- Fructose-bisphosphate aldolase: An intermediate reaction in glycolysis
- Ferrocytochrome-c:oxygen oxidoreductase: The oxygen consuming reaction in the respiratory electron transport chain
- Phosphoserine phosphatase: Reaction producing the amino acid serine from intermediates in the glycolysis
- Shikimate kinase: Intermediate reaction in synthesis of aromatic amino acids (phenylalanine, tyrosine and tryptophan)
- Growth: The growth (biomass) reaction is included for comparison with the other reactions. The calculated growth should ideally be identical to the experimental ones, but some deviations occurred because the posterior particles did not in general provide a perfect fit to the data.

We chose to focus on three simulations; a Bayesian simulation 1 with original priors, the evolutionary with original priors and Bayesian simulation 1 with permuted prior 1 (Figure 4). First, we observe that the flux through shikimate kinase had a narrow flux range and was highly coupled with growth. We suspect this is because the primary function of the reaction is provide amino acids for the growth reaction, and no alternative pathways for producing these amino acids exist. The other reactions however, tell a completely different story. For instance, Fructose-bisphosphate aldolase is for some particles used extensively, but in other cases not at all, still giving rise to approximately the same growth rates regardless. This means that the metabolic model uses alternative pathways depending on the choice of enzyme thermodynamic parameters. The same tendency is also visible for Pyruvate dehydrogenase, Ferrocytochrome-c:oxygen oxidoreductase, and Fructose-bisphosphate aldolase. Differences between the evolutionary algorithm and the Bayesian one are also evident: Within the same Bayesian simulation there are often vast differences in flux midpoint of the reactions for the different particles, whereas the results from the evolutionary algorithm show more consistent results across particles.

The results for the anaerobic dataset (Supplementary Figure S4) were similar except for the fact that there was no flux through the Ferrocytochrome-c:oxygen oxidoreductase reaction as there was *no* oxygen available to be consumed.

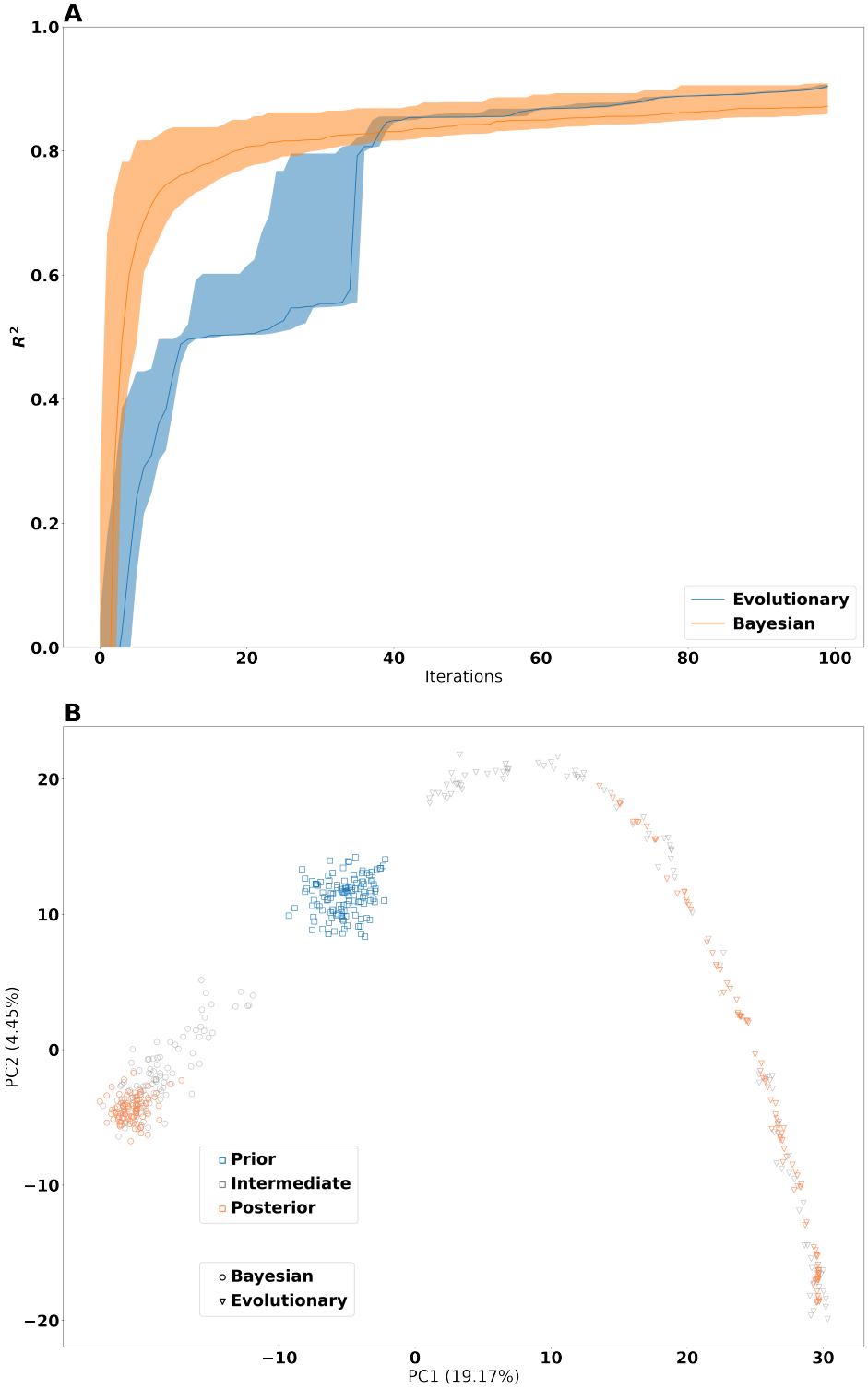

Figure 2. Comparing results from the Bayesian fitting algorithm (unpermuted priors, simulation 1) with the evolutionary algorithm A. Comparison of convergence between the Bayesian and evolutionary approach for the first 100 generations. An R^2 value of 1 corresponds to exact correspondence between the training data and the model predictions. The shaded regions indicate the 5th and 95th percentiles, whereas the solid lines indicate the median (50th percentile). **B.** Principal Component Analysis (PCA) plot of the parameter sets used in the fitting process where each point is a candidate parameter set. The prior points are the ones which served as a starting point for the algorithm, the posterior points are the ones which had $R^2 > 0.9$, whereas all other points are intermediate points stemming from the simulations. The axes are identical for both panels and use the same ordination such that the panels are directly comparable. For clarity, random thinning of the points was applied to the posterior and intermediate points such that only 100 points are shown for each combination of colour and shape.

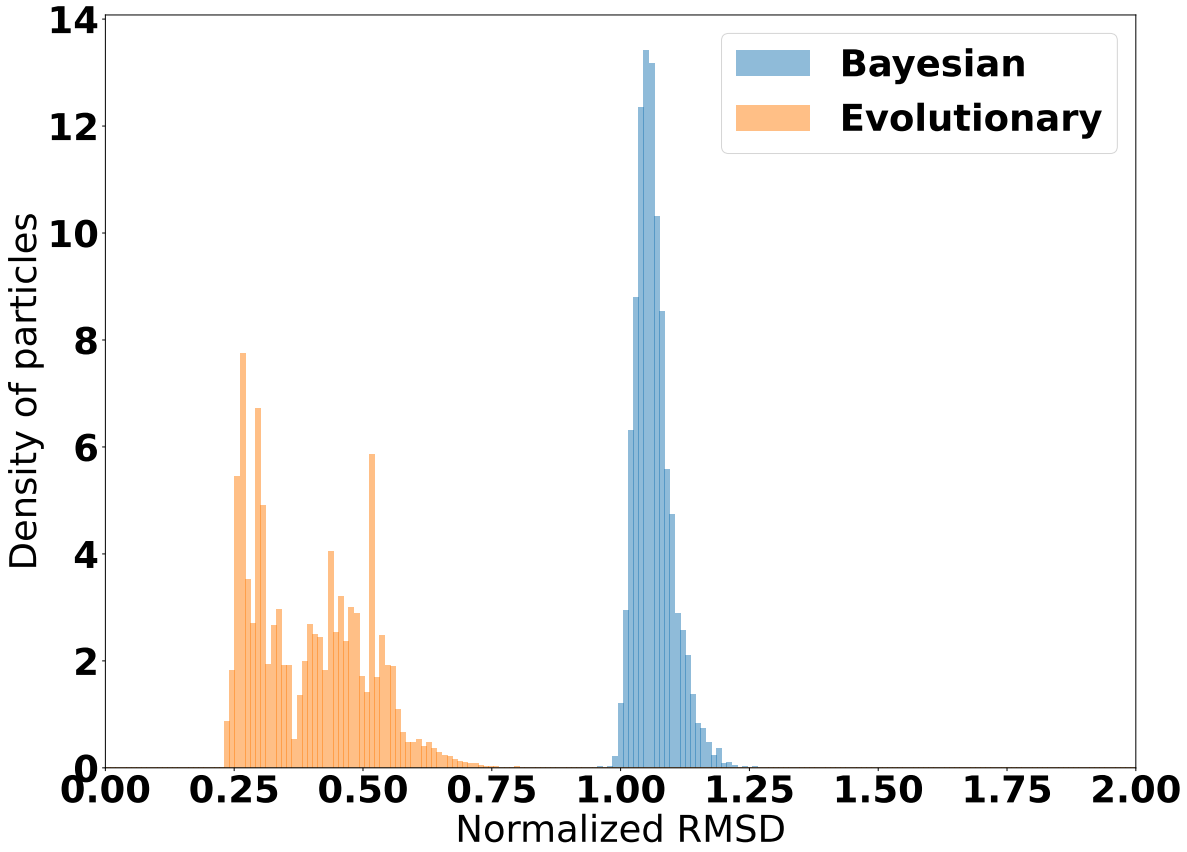

Figure 3. Histogram showing the normalized Root Mean Squared Distance (RMSD) between each particle in the posterior distributions and the centroid of the distribution. The Bayesian algorithm is represented with simulation 1 of the unpermuted priors.

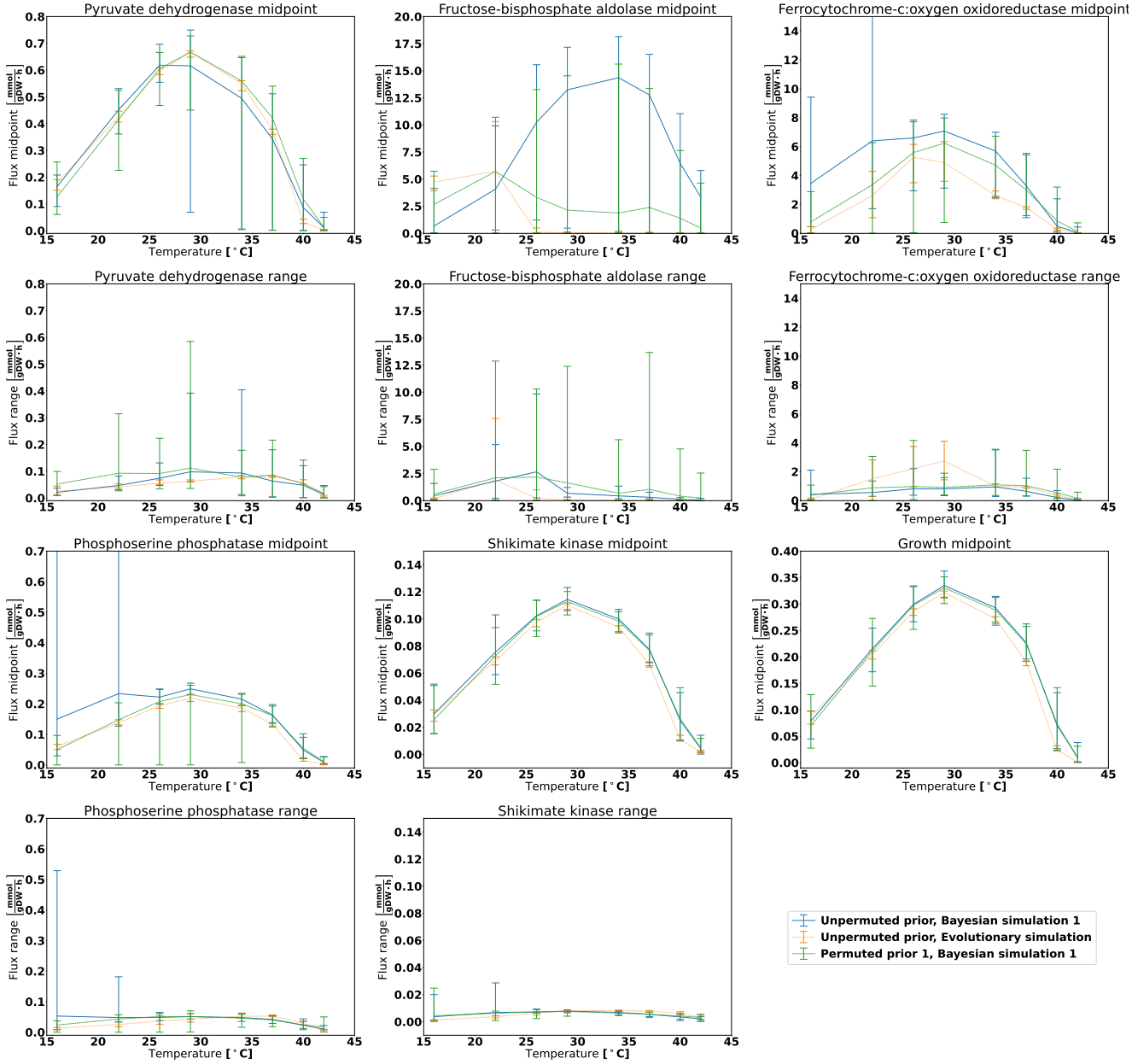

Figure 4. FVA analysis for the aerobic dataset for six reactions and varying temperature. The midpoint panels show the FVA flux midpoint, this is: The average of the maximum and minimum attainable flux given the optimization objective. The range panels show the absolute difference between the maximum and minimum flux. The lines denote the mean midpoint/range value, whereas the error bars span from the lowest to the highest observed value. The growth reaction is included for reference, and it will always display FVA range of zero, as it is the optimization target.

Discussion

From the results seen above, it is clear that the outcome of the Bayesian algorithm indeed depends on priors and random seeds. However, all Bayesian simulations showed similar features: The fitness increased rapidly during the first few iterations before slowly increasing to R^2 values of approximately 0.95. For our study, we used 500 iterations for fitting the parameters. Our rationale for choosing this number was to ensure convergence and acquire a sufficient number of points in the posterior distribution. Spending computational resources on running this many iterations showed diminishing returns, and in our case, 200 iterations would be sufficient to obtain the same major properties.

Each Bayesian simulation converged to a point cloud with no apparent higher-dimensional structure. This observation makes sense considering that the Bayesian algorithm creates new particles by sampling each parameter independently from a normal distribution where the mean and standard deviation is determined by the past generation. Thus, the posterior points are likely to cluster in a high-dimensional cloud where the density of each parameter is normally distributed and the spatial density is the product of the marginal densities of the parameters. Still, because the parameter space had as many as $2,292$ dimensions to average over, the distribution of normalized RMSD values from the cloud centroid was still quite narrow.

With this in mind, it may come as a surprise that particles from the same posterior distribution utilized different metabolic pathways even though the growth rate, metabolic network model topology, and external conditions were identical. This could be due to pathway choice being sensitive to the thermodynamic properties of the enzymes. If so, it would be a possible explanation for the observed cellular metabolic heterogeneity observed in yeast cultures[20,21].

The evolutionary algorithm behaved differently from the Bayesian algorithm even though the final R^2 values were similar. The posterior points did not assemble in a diffuse cloud, by rather seemed to drift as the generations passed on. Still, the phenotypic behaviour was more consistent, and the points closer to the centroid indicating less diversity in the solutions. What might have happened was a genetic bottleneck event upon which subsequent generations became very similar to each other. Using less stringent parameter choices for the selection criteria could potentially alleviate the genetic bottleneck and provide a greater diversity of solutions, some of which could possibly perform better than all the ones found here[22].

Our results have shown that both the Bayesian and evolutionary algorithm were able to produce enzyme parameter sets which equally matched the data. With our software improvements, running the inference algorithms is much faster than with the original implementation, yet the problem is still so computationally expensive that workstation or server-grade hardware is required.

The model parameters are supposed to be properties intrinsic to the enzymes within an organism, and as such, a true set of parameters should exist. Heckmann *et al.* inferred apparent k_{cat} values from large-scale proteomics and metabolomics data on a genome-scale level[23]. Doing something similar with the current etcGEM framework and available data to infer *in vivo* thermodynamic enzyme parameters at a genome-scale level is unlikely to give reliable results. Still at the current stage, some applications of the etcGEMs might work fine. One example of this is the original etcGEM article where Li *et al.* discovered that squalene epoxidase ERG1 is a rate-limiting enzyme under high temperatures, a prediction which was verified experimentally[5].

As the current experimental data are insufficient to constrain the models well enough, access to more extensive experimental data is likely to be essential for the further development of etcGEMs. Such experimental data are likely to include measurement for proteomics[24] and fluxomics[25] in addition to covering a broader range of strains and experimental growth conditions.

Methods

Evaluation method for particles

A parameter set, referred to as a particle, is a collection of the three parameters T_m, T_{opt} and $\Delta C_p^{\ddagger}$ for the 764 metabolic enzymes of ecYeast7.6. The goodness of fit for a particle was evaluated by the framework created by Li *et al.*[5]. In brief, this evaluation procedure acted as a black box taking in a particle and returning an overall R^2 value. In the assessment, each particle was matched against experimental data from three experiments with *Saccharomyces cerevisiae*. These were:

- The aerobic dataset[10] measuring growth rates of yeast at 8 temperatures between 16 °C and 42 °C under aerobic batch fermentations.
- The anaerobic dataset[11] measuring growth rates of yeast at 13 temperatures between 5 °C and 40 °C under anaerobic batch fermentation. However, in our particle assessment, only 8 of the temperatures were used.
- The chemostat dataset[12] measuring exchange fluxes of carbon dioxide, ethanol and glucose at 6 temperatures between 30 °C and 38.5 °C in aerobic chemostats.

The parameters were used to adjust the effective catalytic rate $k_{cat}(T)$ (which includes denaturation) for each enzyme in the model and at each temperature T. In addition, the model's non-growth associated ATP maintenance (NGAM) was also adjusted according to the temperature. Please consult Li *et al.*[5] for details.

For the aerobic and anaerobic datasets, the model's temperature-dependent parameters were tuned, and fluxes were predicted by Flux Balance Analysis (FBA) at the specific temperatures setting the biomass function as objective. For each of the aerobic and anaerobic datasets, the growth rates predicted by the model was compared with the experimental growth rates and an R^2 value was reported, yielding R^2_{rae} and R^2_{ran}.

For the chemostat dataset, the procedure was somewhat different. Here, the temperature-dependent parameters were tuned (as with the other datasets) before the growth rate of the model was locked to the dilution rate of the chemostat. Thereafter, the model was optimized for minimum glucose uptake, and this uptake flux value was set as a constraint to be modeled. Finally, the model was optimized for minimum protein pool usage, and exchange fluxes of ethanol, carbon dioxide, and glucose were recorded. Once the three fluxes for all temperatures in the chemostat dataset were recorded, these values were compared to the experimental ones, and R^2_{chemo} was determined.

The overall R^2 for all datasets was the arithmetic mean of R^2_{rae}, R^2_{ran} and R^2_{chemo}. This value was then returned as the final results of the evaluation procedure. The higher the R^2 value, the higher correspondence exists between the modelled solutions and the experimental results, where $R^2 = 1$ corresponds to the highest achievable fitness.

We modified the evaluation procedure to use ReFramed instead of COBRApy[13] in order to reduce overhead related to modifying models. However, the results generated by our modified approach should be identical to the results generated by the original code by Li *et al.* for all possible simulations.

Approximate Bayesian algorithm with permuted priors

The framework and code for the Sequential Monte Carlo based Approximate Bayesian algorithm was taken directly from Li *et al.*[5] and run with almost the same parameters as in the original publication. The computations were run for 500 iterations. The population size at the end of each iteration was 100. We generated 128 new particles for each iteration and evaluated them according to the description in the previous section. New particles were generated by computing the mean and standard deviation for each parameter of the particle population, and sampling new parameters from a normal distribution with the aforementioned mean and standard deviation. Selection of the particles was done through truncation selection, meaning that the 100 best generated particles from the previous iteration were passed to the next iteration.

Evolutionary algorithm

The evolutionary algorithm for fitting enzyme parameters was written with the Inspyred framework[26] with heavy customizations. Individuals in the evolutionary process were the parameter set particles discussed earlier. The population size for each iteration (generation) was set to 100 in order to provide results comparable the Bayesian algorithm. Selection was implemented through truncation, consistently picking the individuals with the best fitness among the combined pool of the old individuals and the newly created ones. New individuals in the population were created through crossover between existing individuals. In each generation, 32 of the individuals were selected for crossover through tournament selection. These 32 individuals were randomly combined into 16 pairs, and each pair produced 8 children, thus in total giving birth to 128 children per generations. The crossover consisted of averaging the parameter values for all enzymes:

$$\mu_{child} = \frac{\mu_{mom} + \mu_{dad}}{2}. \tag{1}$$

Immediately after birth, each child had a probability of 0.5 of undergoing mutation. If so, the number of parameters to mutate were picked from a Poisson distribution with mean $\lambda = 100$, and the parameters were sampled uniformly with replacement. Mutating a parameter i meant shifting its old value μ_i to a new value μ_i^* by sampling from a normal distribution $\mu_i^* \sim N(\mu_i, \sigma_i)$. The estimated standard deviation of the parameter (σ_i) was initially set to the predefined values reported by Li *et al.*[5]. At the end of each generation, σ_i was updated and set equal to the population standard deviation of μ_i for all individuals in the population. The fitness of the individuals was evaluated as previously mentioned.

Ordinations

We used PCA and t-SNE from scikit-learn (version 1.0)[27] to create ordinations of particles. For both PCA and t-SNE, the values for each parameter were standardized subtracting the mean and dividing by the population standard deviation. These means and standard deviations were computed across all particles present in the ordination. For t-SNE, the learning rate was set to `"auto"` and the ordination was initialized by PCA. Perplexity was varied across a range of 10 values from 5 to 2000. All other t-SNE parameters were set to default values. Initializing t-SNE by PCA eliminated the effect of random state. Choosing high perplexities decreases the fitness gradient for t-SNE, and thus, the results will look more like the initial arrangements of points. For this reason, plots for high t-SNE perplexities looked similar to the corresponding PCA ordinations.

RMSD calculations

Normalized Root Mean Squared Distances (RMSD) of particle posterior distributions were calculated as follows: All posterior particles (having $R^2 > 0.9$) from each of the $S = 9$ simulations were picked. Further, let $x_{i,k,j}$ be the value of parameter i of posterior particle k in simulation j. Further, let n_j denote the number of posterior particles for simulation j and M denote the number of enzyme parameters (which was the same for all particles). Then

$$x_{i,.,.} = \frac{1}{S}\sum_{j=1}^{S}\frac{1}{n_j}\sum_{k=1}^{n_j} x_{i,k,j}, \tag{2}$$

was the overall average of parameter i with the corresponding standard deviation:

$$\sigma_i = \sqrt{\frac{1}{S}\sum_{j=1}^{S}\frac{1}{n_j}\sum_{i=1}^{n_j}\left(x_{i,k,j} - x_{i,.,.}\right)^2}. \tag{3}$$

For each simulation, a centroid was computed by averaging each enzyme parameter over all posterior particles in the simulation:

$$x_{i,.,j} = \frac{1}{n_j}\sum_{k=1}^{n_j} x_{i,k,j}. \tag{4}$$

Finally, the RMSD values were computed for each particle by taking the average standardized Euclidean distances from the centroid for all enzyme parameters:

$$\mathrm{RMSD}_{k,j} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\frac{x_{i,k,j} - x_{i,.,j}}{\sigma_i}\right)^2}. \tag{5}$$

Flux variability analysis

Flux variability analysis (FVA) was conducted on the posterior distributions of all nine simulations. From each simulation, 20 particles were sampled randomly from the posterior particles, sampling a total of 180 particles. For each sampled particle, FVA was run for all three evaluating datasets across the same temperatures used for determining R^2 in the parameter fitting process. For the aerobic and anaerobic datasets, the temperature and the parameters of the particles were first used to fix the effective k_{cat} values. Subsequently, the metabolic model was optimized for maximal growth, and the lower bound of growth reaction was locked to the obtained growth. Thereafter, maximum and minimum fluxes through each reaction in the model were found given the constraints.

For the chemostat dataset, the approach was similar: The effective k_{cat} values were first mapped onto the model before the growth reaction was locked to the experimental dilution rate. Thereafter, the model's glucose uptake was minimized and set as an upper bound. Protein usage was then minimized and set as a constraint, and FVA was used to obtain upper and lower bounds for all reactions. Instead of using the maximum and minimum fluxes directly, we converted them into flux midpoint (average of maximum and minimum) and flux ranges (absolute difference between maximum and minimum). Results with missing or infinite values were removed.

All flux ranges and flux midpoints with the same combination of simulation, reaction, dataset, and temperature were aggregated to give the mean, minimum and maximum values. Usually, there were 20 such values for each combination, as 20 particles were sampled from each posterior distribution. However, this number could be smaller due to removal of missing and infinite values.

References

1. van der Kamp, M. W. *et al.* Dynamical origins of heat capacity changes in enzyme-catalysed reactions. *Nat. Commun.* **9**, 1177, DOI: 10.1038/s41467-018-03597-y.
2. Hobbs, J. K. *et al.* Change in heat capacity for enzyme catalysis determines temperature dependence of enzyme catalyzed rates. *ACS Chem. Biol.* **8**, 2388–2393, DOI: 10.1021/cb4005029.
3. Birch, A. N., Petersen, M. A. & Åse S. Hansen. The aroma profile of wheat bread crumb influenced by yeast concentration and fermentation temperature. *LWT - Food Sci. Technol.* **50**, 480–488, DOI: https://doi.org/10.1016/j.lwt.2012.08.019.
4. Molina, A. M., Swiegers, J. H., Varela, C., Pretorius, I. S. & Agosin, E. Influence of wine fermentation temperature on the synthesis of yeast-derived volatile aroma compounds. *Appl. Microbiol. Biotechnol.* **77**, 675–687, DOI: 10.1007/s00253-007-1194-3.

5. Li, G. *et al.* Bayesian genome scale modelling identifies thermal determinants of yeast metabolism. *Nat Commun* **12**, 190, DOI: 10.1038/s41467-020-20338-2 (2021).

6. Sánchez, B. J. *et al.* Improving the phenotype predictions of a yeast genome-scale metabolic model by incorporating enzymatic constraints. *Mol Syst Biol* **13**, 935, DOI: 10.15252/msb.20167411 (2017).

7. Bekiaris, P. S. & Klamt, S. Automatic construction of metabolic models with enzyme constraints. *BMC bioinformatics* **21**, 19, DOI: 10.1186/s12859-019-3329-9.

8. Fell, D. A. & Small, J. R. Fat synthesis in adipose tissue. an examination of stoichiometric constraints. *The Biochem. journal* **238**, 781–6.

9. Kauffman, K. J., Prakash, P. & Edwards, J. S. Advances in flux balance analysis. *Curr. opinion biotechnology* **14**, 491–6.

10. Caspeta, L. & Nielsen, J. Thermotolerant yeast strains adapted by laboratory evolution show trade-off at ancestral temperatures and preadaptation to other stresses. *mBio* **6**, e00431, DOI: 10.1128/mBio.00431-15 (2015).

11. Zakhartsev, M., Yang, X., Reuss, M. & Pörtner, H. O. Metabolic efficiency in yeast saccharomyces cerevisiae in relation to temperature dependent growth and biomass yield. *J Therm Biol* **52**, 117–29, DOI: 10.1016/j.jtherbio.2015.05.008 (2015).

12. Postmus, J. *et al.* Quantitative analysis of the high temperature-induced glycolytic flux increase in saccharomyces cerevisiae reveals dominant metabolic regulation. *J Biol Chem* **283**, 23524–32, DOI: 10.1074/jbc.M802908200 (2008).

13. Ebrahim, A., Lerman, J. A., Palsson, B. O. & Hyduke, D. R. Cobrapy: Constraints-based reconstruction and analysis for python. *BMC Sys Bio* **7**, 74, DOI: 10.1186/1752-0509-7-74.

14. Pearson, K. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, Dublin Philos. Mag. J. Sci.* **2**, 559–572, DOI: 10.1080/14786440109462720. https://doi.org/10.1080/14786440109462720.

15. Maia, P., Rocha, I. & Rocha, M. Identification of robust strain designs via tandem pfba/lmoma phenotype prediction. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, 1661–1668, DOI: 10.1145/3067695.3082542 (Association for Computing Machinery, New York, NY, USA).

16. Patil, K. R., Rocha, I., Förster, J. & Nielsen, J. Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinforma.* **6**, 308, DOI: 10.1186/1471-2105-6-308.

17. Rocha, M. *et al.* Natural computation meta-heuristics for the in silico optimization of microbial strains. *BMC Bioinforma.* **9**, 499, DOI: 10.1186/1471-2105-9-499.

18. van der Maaten, L. & Hinton, G. Visualizing data using t-sne. *JOURNAL OF MACHINE LEARNING RESEARCH* **9**, 2579–2605.

19. Mahadevan, R. & Schilling, C. H. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metab. engineering* **5**, 264–276, DOI: 10.1016/j.ymben.2003.09.002.

20. Pinheiro, S., Pandey, S. & Pelet, S. Cellular heterogeneity: yeast-side story. *Fungal Biol. Rev.* **39**, 34–45, DOI: https://doi.org/10.1016/j.fbr.2021.11.005.

21. MacGillivray, M. *et al.* Robust analysis of fluxes in genome-scale metabolic pathways. *Sci. reports* **7**, 268, DOI: 10.1038/s41598-017-00170-3.

22. Alter, T. B., Blank, L. M. & Ebert, B. E. Genetic optimization algorithm for metabolic engineering revisited. *Metabolites* **8**, 33, DOI: 10.3390/metabo8020033.

23. Heckmann, D. *et al.* Kinetic profiling of metabolic specialists demonstrates stability and consistency of in vivo enzyme turnover numbers. *Proc Natl Acad Sci U S A* **117**, 23182–23190, DOI: 10.1073/pnas.2001562117 (2020).

24. Großeholz, R. *et al.* Integrating highly quantitative proteomics and genome-scale metabolic modeling to study ph adaptation in the human pathogen enterococcus faecalis. *NPJ systems biology applications* **2**, 16017–16017, DOI: 10.1038/npjsba.2016.17.

25. Winter, G. & Krömer, J. O. Fluxomics – connecting 'omics analysis and phenotypes. *Environ. Microbiol.* **15**, 1901–1916, DOI: https://doi.org/10.1111/1462-2920.12064. https://sfamjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/1462-2920.12064.

26. Garret, A. inspyred (version 1.0.1) [software]. inspired intelligence. (2012).

27. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830.

Acknowledgements

Thanks to Gang Li for troubleshooting technical problems with the simulations. J.P.P. and E.A. are grateful to ERA CoBioTech project CoolWine and Norwegian Research Council grant 283862 for funding.

Author contributions statement

E.A. and J.P.P. conceived the project. J.P.P did the analysis, created visualisations and wrote the first draft of the paper. All authors contributed to and accepted the final version of the paper.

Additional information

Availability of source code and data

The source code for the analysis and visualisation is available at GitHub (https://github.com/AlmaasLab/BayesianGEM). Also, an archived copy of the repository is available at Figshare (https://doi.org/10.6084/m9.figshare.20180261). Simulation of genome-scale models has carried out with ReFramed (https://github.com/cdanielmachado/reframed) using Gurobi (Gurobi Optimization, LLC) as solver. The data required for reproducing the figures in the main article and in the Supplementary material is available as a Figshare repository (https://doi.org/10.6084/m9.figshare.20180216).

Competing interests

The authors declare that there are no competing interests.