Parameter inference for enzyme and temperature constrained genome-scale models

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 system level is of importance. A recently developed Bayesian computational framework for enzyme and temperature constrained genome-scale models (etcGEM) 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.
Here, we show that the Bayesian calculation method for inferring parameters for an etcGEM converges under a range of different priors and random seeds. However, the method is unstable, resulting in estimated solutions that differ both in parameter space and on a phenotypic level. To remedy this problem, we developed an alternative evolutionary algorithm which did well in exploring the parameter space.
We quantified the phenotypic consequences on six metabolic network signature reactions of the different parameter solutions resulting from use of the evolutionary algorithm. While two of these reactions showed little phenotypic variation between the solutions, the remainder displayed huge variation in flux-carrying capacity. This result indicates that the model is under-determined given current experimental data and that more data is required to narrow down the model predictions. Finally, we made improvements to the software to reduce the running time of the parameter set evaluations by a factor of 8.5, allowing for obtaining results faster and with less computational resources.

Introduction

Temperature is a key effector of life, which is partially due to the consequence that temperature has on catalytic properties of enzymes. For a long time, it has been known that enzymatic reactions slow down at low temperatures, whereas high temperatures destroy the enzymes, rendering them non-functional. In recent research[1,2], it has also been acknowledged that enzymes 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 them. 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 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 microorganismal phenotypes by propagating the temperature dependence of metabolic enzymes to the entire metabolic network of an organism. However, Gang Li *et al.*[5] came 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 model ecYeast7.6[6] (*Saccharomyces cerevisiae* strain S288C) and contains 764 metabolic enzymes and $2,292$ parameters associated with enzymatic activity. For each of the enzymes, the following parameters must be determined: (1) T_m, the melting temperature; (2) T_{opt}, the temperature optimum; and (3) $\Delta C_p^{\ddagger}$, 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, Li and coworkers[5] used a Bayesian approach to infer the enzymatic parameters mentioned above from sets of training data: (1) Maximal growth rates in aerobic batch cultivations[10]; (2) Maximal growth rates in anaerobic batch cultivations[11]; (3) Chemostat

cultivations which include measurements of exchange fluxes of carbon dioxide, ethanol and glucose[12].

In the training data, the experimentally determined exchange 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 defining the Bayesian model, 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 calculation method, an estimate of the posterior distribution of parameter sets was found[5]. However, Li *et al.* did not systematically investigate the stability of this calculation method, nor whether it suffered from identifiability issues. Thus, it is unclear how metabolic flux results from the etcGEM can be interpreted.

In this paper, we investigated the stability of the Approximate Bayesian calculation method algorithm by choosing multiple different random seeds over different priors. Additionally, we explored parameter sets resulting from a different algorithmic approach, that of an evolutionary algorithm. Finally, we 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 calculation method once for 500 iterations with the chosen hyper-parameters consumed approximately $17,000$ CPU hours (approximately corresponding to two weeks on a 48 core computer) using the implementation from Li *et al.* Profiling showed that the particle evaluation procedure (see Methods) was the performance bottleneck, and excess time consumption was caused by COBRApy's[13] internal routines to modify metabolic models prior to solving them. 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 calculation method could be obtained the day after starting it when running on a compute server. Still, only about 20% of the particle evaluation time was spent on optimization, so improvements within the ReFramed package has the potential for increasing performance even more.

Assessing the stability of the Bayesian calculation method

In order to investigate the stability of the Bayesian calculation method for stochastic effects, we ran the Bayesian calculation method with two different random seeds on the three training datasets. These runs of the Bayesian calculation method are referred to as Bayesian simulation 1 and 2. Also, in addition to using the priors selected by Ref.[5], we created three randomized priors by permutation (see Methods for details) and repeated the process of assessing stability given these priors. Thus in total, we ran the Bayesian calculation method eight times, yielding eight different populations of estimated posterior distributions. 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 result in parameter sets with high fitness, we next investigated whether these solutions were similar. For this, we created a Principal Component Analysis[14] (PCA) plot of the parameter sets obtained under estimation (Figure 1 **C** and **D** with a more complete overview in Supplementary Figure S2). The estimated 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 the Bayesian calculation method is unstable for all four priors and that there are identifiability issues causing the calculation method to converge at different locations in the parameter space.

We also discovered that the calculation of R^2 values for the chemostat dataset suffered from numerical instabilities. For the same particle and software version, the Gurobi solver could sometimes judge the model infeasible given the parameters and sometimes it could find a feasible solution. However, given that a solution was found, the results were consistent up to

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

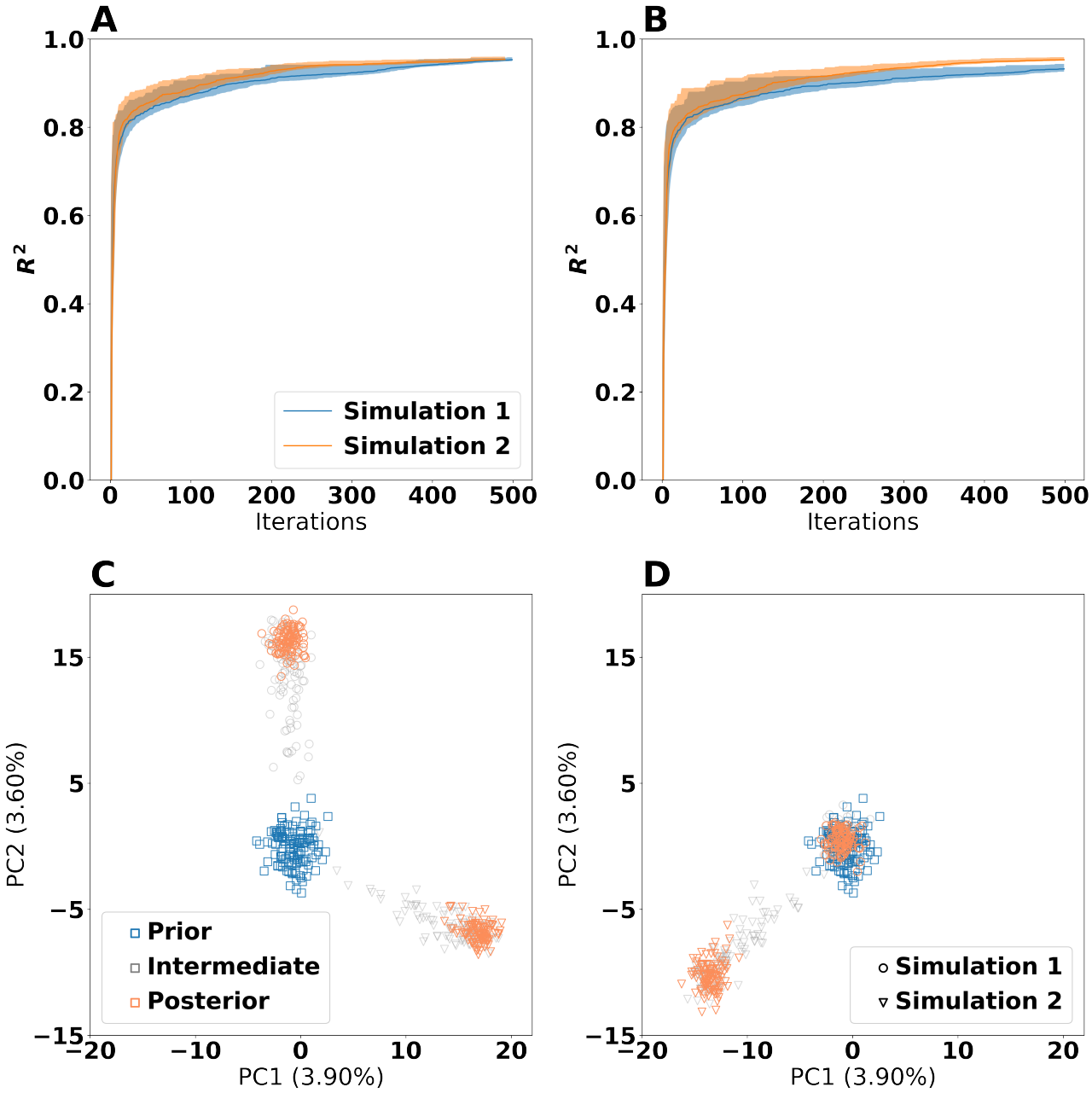

Figure 1. Comparison of the effect of prior and random seed on the Bayesian calculation method. Panel **A** and **B** show the training R^2 values for the unpermuted priors and permuted prior 1, respectively. An R^2 value of 1 corresponds to exact correspondence between the training data and the model 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 calculation method, the estimated 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, making the panels 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.

expected numeric accuracy. Given this problem, we also ran the Bayesian calculation method without the chemostat data. We only used the priors suggested by Li *et al.* and ran four different simulations with differing random seed. The results for this setup (Supplementary Figure S3 and S4) were similar to the simulations including the chemostat dataset. Hence, we find it unlikely that the chemostat dataset alone caused the Bayesian calculation method to become unstable.

Assessing enzyme-level stability of the Bayesian calculation method

To systematically study the phenotypic behaviour of particles in the estimated posterior distributions, we performed Flux Variability Analysis (FVA)[15], 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 capable of supporting optimal metabolic behaviour. 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; Bayesian simulation 1 and 2 with original priors and Bayesian simulation 1 with permuted prior set 1(Figure 2). 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. There are differences between the results of the three different simulations, but the flux ranges and midtpoints within each estimated posterior distribution contain large variations.

The results for the anaerobic dataset (Supplementary Figure S6) 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. We also ran FVA on the results omitting the chemostat dataset when running the Bayesian calculation method (Supplementary figure S7 and S8). These results also showed large variability within simulation results and across simulations.

Evolutionary algorithm

Given the instability of Bayesian calculation method, we constructed an evolutionary algorithm[16–19] as an alternative for inferring parameters (see Methods). Our choice of an evolutionary method was motivated by how an evolutionary algorithm searches the parameter space and its ability to combine existing solutions to create improved solutions. We used the same prior as suggested by Li *et al.* and chose two different population replacement strategies, (local) tournament and truncation (see Methods for details). We also chose to discard the chemostat dataset for these simulations due to its associated instability. For the local tournament replacement, we varied the locality of the tournament, ie. how many of the nearest neighbours were considered for the tournament. For truncation replacement, we varied the number of elites (best individuals) to include in the next generation, while the remainder of the next generation was randomly selected. Simulations were conducted in replicate, meaning that for any value of locality or number of elites, two simulations were executed with differing random seeds.

As can be seen from the evolutionary algorithm with truncation replacement (Figure 3), the fitness often manages to exceed $R^2 = 0.9$ by 200 generations. However, there are considerable stochastic effects and even replicate simulations may provide large differences in final fitness. The number of elites does not have a marked effect the convergence properties, as all simulations with 16 or 128 elites exceed $R^2 = 0.9$. However, for each of the 32 and 64 elite simulations, there was one simulation failing to reach $R^2 = 0.8$. In the cases of the evolutionary algorithm with tournament replacement (Supplementary Figure S5), the results are similar and the choice of the locality parameter does not have a large effect on the convergence properties.

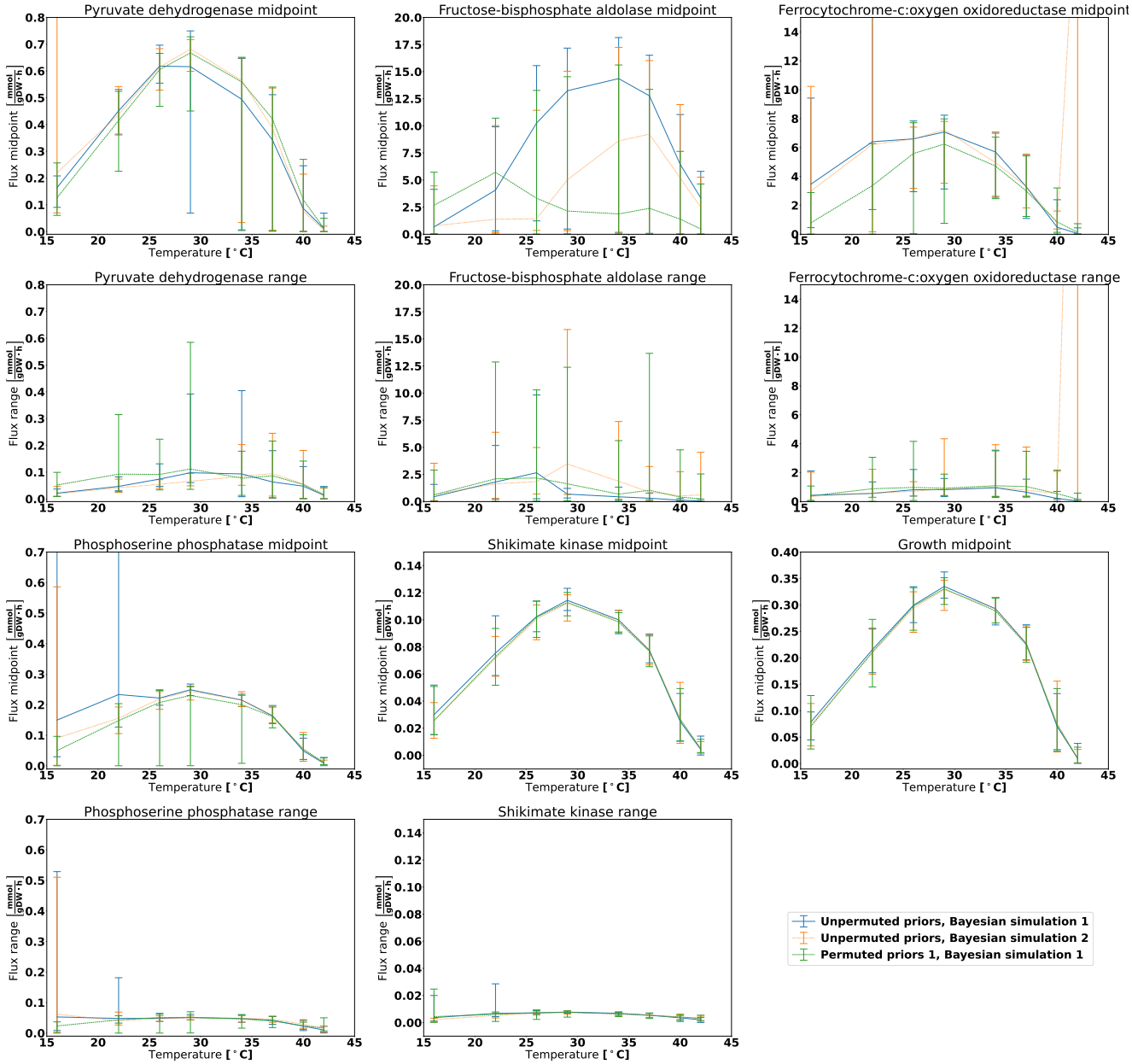

Figure 2. FVA analysis for the aerobic dataset for six reactions and varying temperature when using estimated posterior distributions obtained for the Bayesian calculation method. The midpoint panels show the FVA flux midpoint, this is: The average of the maximum and minimum attainable flux given the optimization objective. The range panels show the absolute difference between the maximum and minimum flux. The lines denote the mean midpoint or 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 an FVA range of zero as it is the optimization target.

PCA ordinations (Figure 4) of the particles showed that the evolutionary algorithm converged to different locations in the parameter space depending on random seed, choice of replacement method, and value of number of elites/locality parameter. With a few exceptions, points from the final generation from the same simulation were stacked atop of each other. This may indicate that the evolutionary algorithm suffers from genetic bottlenecking.

FVA analysis on the results of the evolutionary algorithm (Figure 5 and Supplementary Figure S9) revealed major differences between the simulations. For instance, the two simulations with truncation replacement and number of elites equal to 16 agreed on the fluxes through the selected reactions and reported close to zero variability in flux midpoint and range. However, for other simulations, large variability of flux range and variability within the solution itself was observed for certain reactions in addition to large differences to the other simulations.

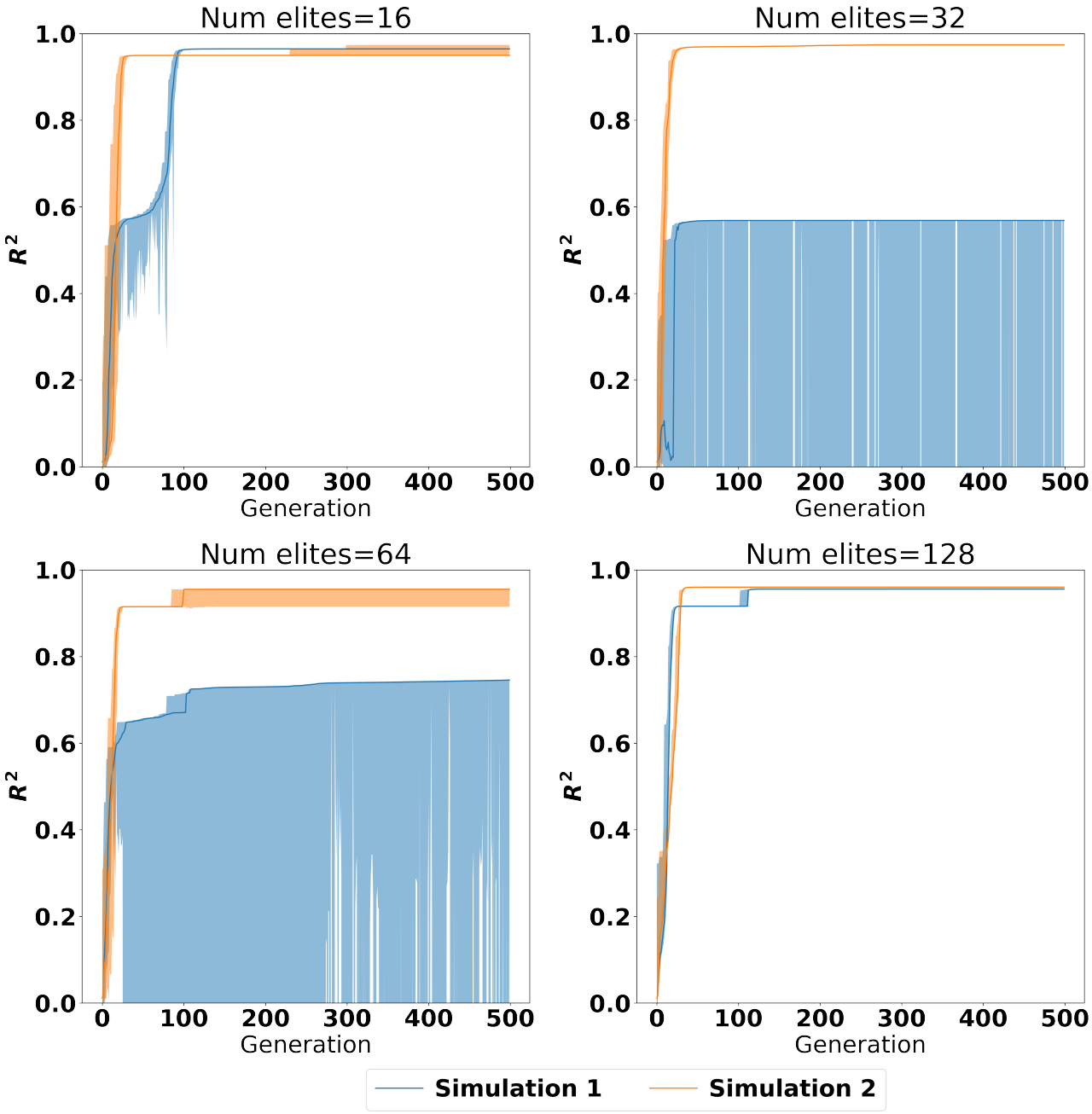

Figure 3. Training R^2 values for the evolutionary algorithm with truncation replacement. Each panel contains the results of both simulation 1 and 2. An R^2 value of 1 corresponds to exact correspondence between the training data and the model predictions. The shaded regions indicate the 5th and 95th percentiles, whereas the solid lines indicate the median (50th percentile).

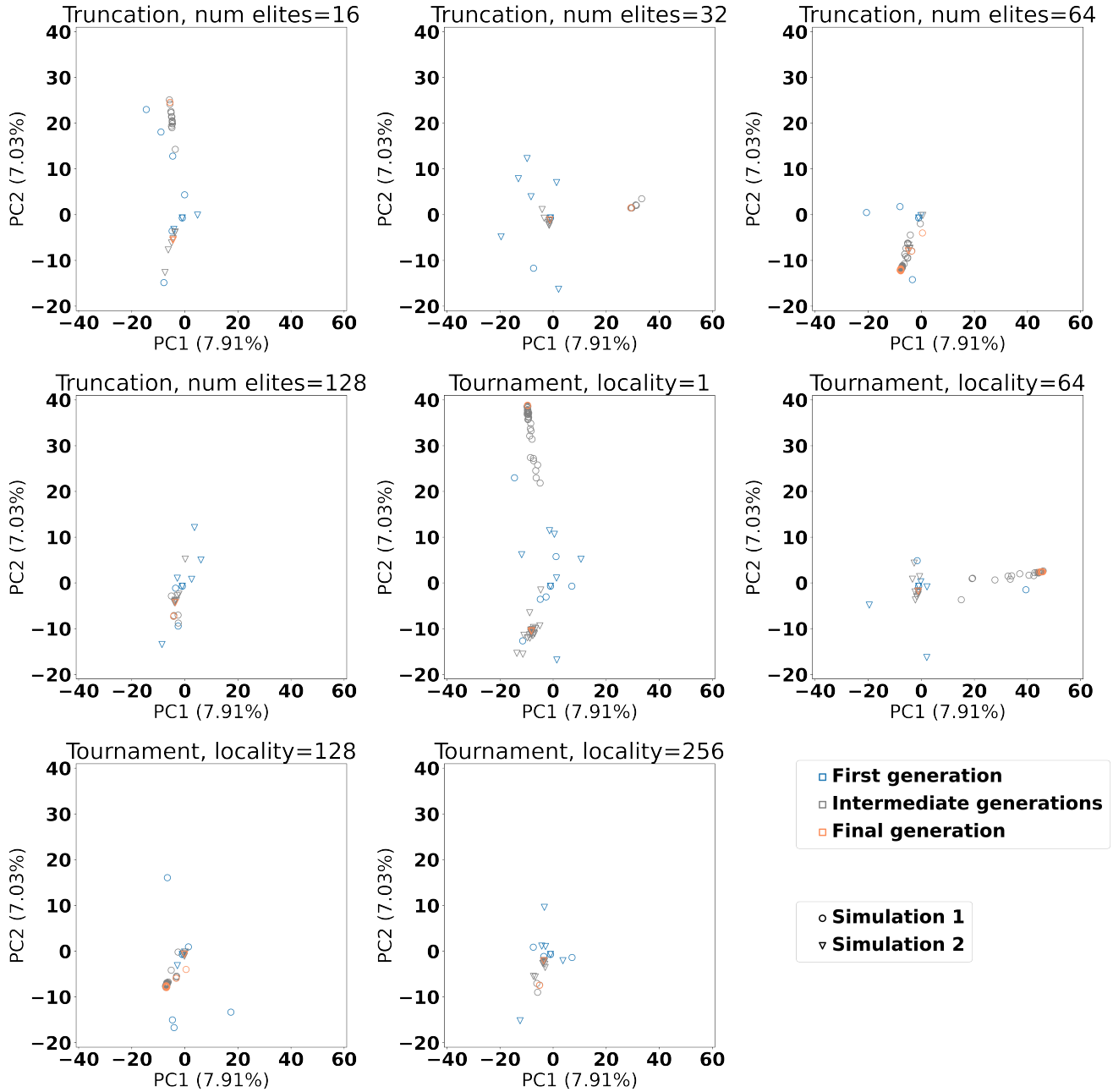

Figure 4. Principal Component Analysis (PCA) plot of the parameter sets used in the evolutionary algorithm where each point is a candidate parameter set. The colours indicate the particles being present in the seeding (first) generation, the final generation and all other (intermediate) generations. The axes are identical for all panels and use the same ordination, making the panels directly comparable. For clarity, random thinning of the points was applied to the particles of the first generation and intermediate generations such that only 10 and 100 points, respectively, are shown for each combination of panel, shape and colour for these points.

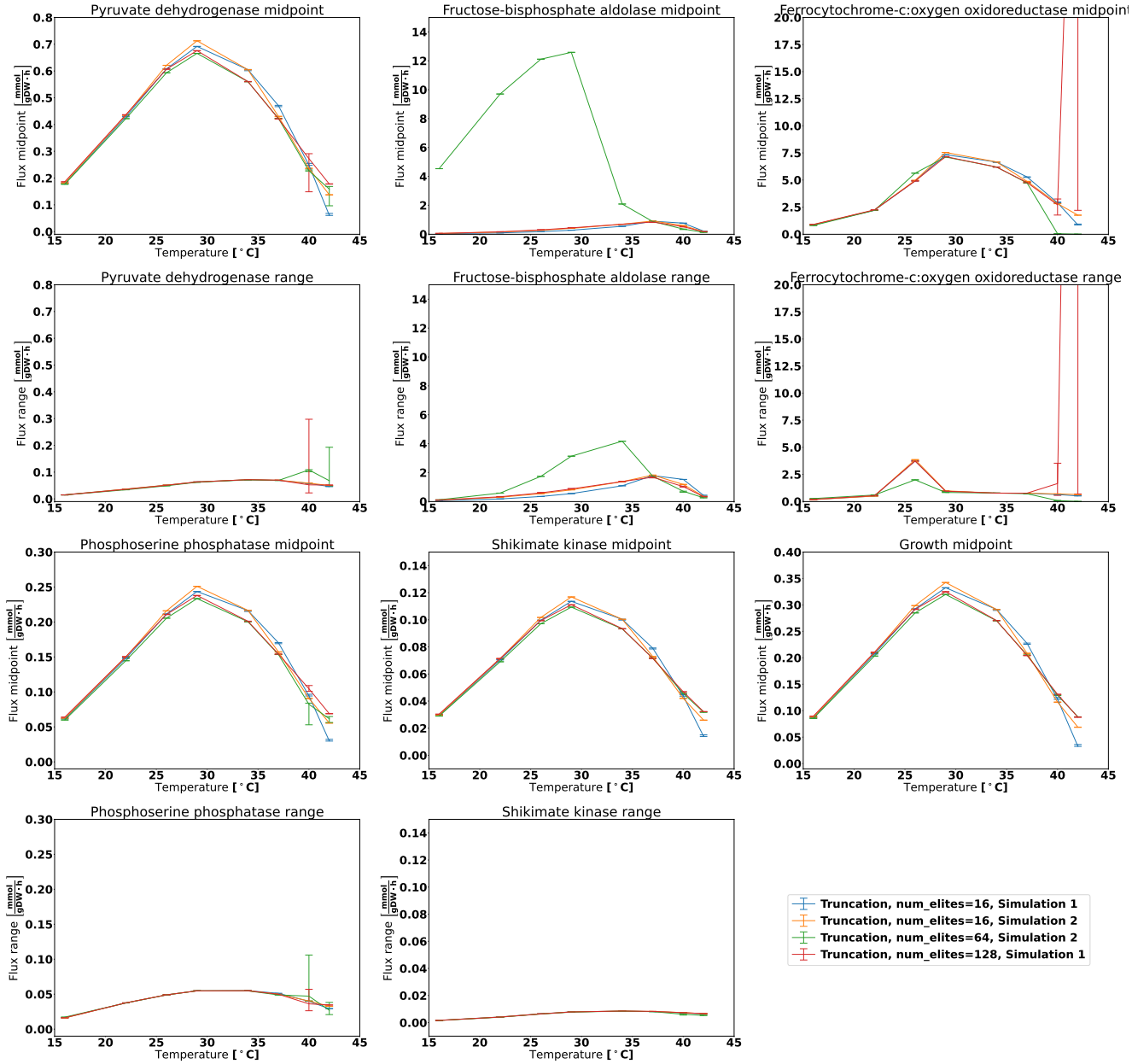

Figure 5. FVA analysis from the evolutionary algorithm with truncation replacement, carried out for the aerobic dataset for six reactions and varying temperature. The midpoint panels show the FVA flux midpoint, ie. the average of the maximum and minimum attainable flux given the optimization objective. The range panels show the absolute difference between the maximum and minimum flux. The lines denote the mean midpoint or 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 an FVA range of zero as it is the optimization target.

Discussion

We observed that computing R^2 values for the chemostat dataset resulted in numerical instability, while this problem was not present for the other two datasets. This is likely due to the the three-stage procedure of first locking the growth rate of the model to the dilution rate, then minimizing glucose uptake and setting it as a constraint for the model, before finally minimizing the protein usage and then reporting the fluxes. Even though the resulting problem is mathematically solvable, the sharp constraints still cause problems for the Gurobi LP solver, which for the same particle sometimes managed to find a feasible solution to the problem, and sometimes not. Potentially, this challenge could be mitigated by reformulating the optimization problem to obtain a growth rate and glucose uptake rate as close as possible (but not necessarily equal) to the desired values. Also, Gurobi supports directly setting lexicographic objectives solved in sequence, an approach which hopefully does not possess the aforementioned problem.

From the results reported above, the outcome of the Bayesian calculation method is unstable and indeed depends on the choice of random seed for all priors tested. Combined with the fact that the fitness of all Bayesian simulations ended at approximately 0.95, it seems that the true theoretical posterior distribution is multimodal for the priors tested. Each Bayesian simulation converged to a point cloud with no apparent higher-dimensional structure. This observation makes sense, considering that the Bayesian calculation method 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. Hence, the Bayesian calculation method under consideration will converge to a unimodal distribution even though the true posterior distribution is not. This is a serious problem, given that the task of the calculation method is to estimate the posterior distribution.

For our case, where the posterior distribution indeed seems to be multimodal, the Bayesian calculation method is therefore of limited value. Additionally, the flux ranges and midpoints determined by FVA display large variations for some reactions, meaning that differences between estimated posterior distributions had considerable phenotypic effects. However, the variability was not only due to being selected from different estimates of the posterior distribution, since particles from the *same* estimated posterior distributions also showed large disagreement on flux ranges and midpoints.

The evolutionary algorithm turned out to produce considerably different results just by the effect of random seed. Sometimes, convergence towards high R^2 values above 0.9 was observed rather quickly, but in some cases the fitness did not reach these levels by the time the simulation ended. Also, there were solutions which were different in the parameter space, but had the same phenotypic behaviour in the selected reactions. On the other hand, there were also solutions which gave considerably different results for some reactions. The choice of replacement method or values of number of elites/locality parameter did not appear to yield any consistent effect, however. In particular, our attempt to use local tournament did not have the intended effect, since the simulation still went through genetic bottlenecking while setting the locality parameter equal to 1.

Unlike the Bayesian calculation method, the great variety of different outcomes of the evolutionary algorithm is not a weakness per se, but rather an indication of the desired feature of exploring the parameter space. Still, the variable results suggest that the identifiability of the model is poor: There is just not enough data to constrain the model so that it is reduced to a narrow phenotype. The results suggest that the choice of metabolic pathways for yeast is sensitive to the thermodynamic properties of the enzymes even if the growth rate, metabolic network model topology, and external conditions were kept the same. If so, it could be a possible explanation for the cellular metabolic heterogeneity observed in yeast cultures[20,21]. However, the fact that a variety of different flux distributions are consistent with the data, has most likely much to do with a lack of constraints from the experimental data rather than heterogeneity within the population of yeast cells.

Given the large variability of internal fluxes, direct or indirect measurements of fluxes would narrow down the solution space. 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 measurements of proteomics[23] and fluxomics[24] in addition to covering a broader range of strains and experimental growth conditions. For instance, Heckmann *et al.* inferred apparent k_{cat} values from large-scale proteomics and metabolomics data on a genome-scale level[22]. Adopting this method to the current etcGEM framework to infer *in vivo* thermodynamic enzyme parameters is thus an approach we recommend.

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. This computational burden is likely to increase when incorporating more experimental data to calibrate the parameters. Therefore, systematic improvements should be implemented in the framework to minimize unnecessary overhead in order to hold the computational burden at a manageable level.

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 the ecYeast7.6 model. 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. Further details are published in Li *et al.*[5].

For the aerobic and anaerobic datasets, the model's temperature-dependent parameters were tuned, and fluxes were predicted by Flux Balance Analysis (FBA) calculations at the specific temperature values, and the biomass function was set as the objective. For each of the aerobic and anaerobic datasets, the growth rates predicted by the model were 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 for the model. 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 calculated as 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 the module 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 calculation method with permuted priors

The framework and code for the Sequential Monte Carlo based Approximate Bayesian calculation method was taken directly from Li *et al.*[5], and we used the same hyper-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 implemented 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 from the ground up. Individuals in the evolutionary process were the parameter set particles discussed earlier. The population size for each iteration (generation) was set to 128. Replacement of individuals was implemented through one of the two different algorithms:

- **Local tournament algorithm:** Individuals from the combined pool of the current generation and the newly generated individuals (offspring) were replaced one by one until the population again reached the predetermined size of 128. For each tournament, the first individual a was randomly sampled from the population. Then, the normalized parameter-space distance from this individual to the other individuals was calculated:

$$D_{a,i} = \sum_{k=1}^{M} \left(\frac{\mu_{a,k} - \mu_{i,k}}{\sigma_k} \right), \tag{1}$$

where $\mu_{a,k}$ and $\mu_{i,k}$ are values for the enzyme parameter k for the two individuals, M is the total number of enzyme parameters and σ_k is the empirical standard deviation of enzyme parameter k in the previous population.

Defining L as the locality parameter, the set B containing the L nearest neighbours of a was determined:

$$B = \{b_1, b_2, b_3, \ldots, b_L\}, \quad \text{such that } a \notin B \text{ and for any } i \in B \text{ and } j \notin B \cup \{a\} \quad D_{a,i} < D_{a,j}. \tag{2}$$

Thus, the second individual b to participate in the tournament was randomly chosen along the elements of B.

With a and b chosen, the individual with the lowest fitness was removed from the population. Our rationale for exploring the local tournament algorithm for replacement was to facilitate niche exploration. This is a process in which the population diverges into different "geographical" (regions in parameter space) subpopulations and thus becomes more efficient in exploring a multimodal fitness landscapes[25].

The locality parameter L was varied through the values 1, 64, 128 and 256, where 1 corresponds to consistently picking the nearest neighbour and 256 turns the algorithm into an ordinary (global) tournament algorithm.

- **Truncation algorithm:** Individuals from the combined pool of the current generation and the newly generated offspring were ranked according to their fitness. Then, the E individuals having the highest fitness (elite) were picked for the next generation, while the remaining $128 - E$ individuals for the next generation were sampled randomly among the non-elites. The parameter E determining the number of elites was varied through the values 16, 32, 64, and 128. Choosing $E = 128$ has the consequence of consistently selecting only the individuals with highest fitness. On the other hand, choosing $E = 0$ would be equivalent to not applying selection pressure during replacement.

In order to select parents to generate offspring, 64 out of the 128 individuals in the current generation were selected through rank selection. Here, the probability of an individual being chosen was proportional to its fitness rank in the population. These 64 individuals were randomly combined into 32 pairs, and each pair produced 1 child, thus in total giving birth to 32 children per generation. The crossover consisted of averaging the parameter values for all enzymes:

$$\mu_{child} = \frac{\mu_{mom} + \mu_{dad}}{2}. \tag{3}$$

Immediately after birth, each child had a probability of 0.5 of undergoing mutation. If so, the number of parameters to mutate was sampled 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 except that we did not use the chemostat dataset for evaluating the R^2 value.

Ordinations

We used PCA from scikit-learn (version 1.0)[26] to create ordinations of particles. The values for each parameter were standardized subtracting the mean and dividing by the population standard deviation before ordination. These means and standard deviations were computed across all particles present in the ordination. As a result, the presented ordinations are comparable across panels.

Flux variability analysis

Flux variability analysis (FVA)[15] was conducted on particles having $R^2 > 0.9$ in order to ensure that these particles had high fitness. From each simulation, 20 particles were sampled randomly from the particles with $R^2 > 0.9$. For each sampled particle, FVA was run for the aerobic and anaerobic datasets across the same temperatures used for determining R^2 in the parameter fitting process. For the chemostat dataset, the numeric instability was too large to produce reliable results. 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 the growth reaction was locked to the obtained growth. Thereafter, maximum and minimum fluxes through each reaction in the model were found given the constraints. 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. 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.
16. 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).
17. 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.
18. 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.
19. Alter, T. B., Blank, L. M. & Ebert, B. E. Genetic optimization algorithm for metabolic engineering revisited. *Metabolites* **8**, 33, DOI: 10.3390/metabo8020033.
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. 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).

23. 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.

24. 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.

25. Mengshoel, O. J. & Goldberg, D. E. The Crowding Approach to Niching in Genetic Algorithms. *Evol. Comput.* **16**, 315–354, DOI: 10.1162/evco.2008.16.3.315 (2008). https://direct.mit.edu/evco/article-pdf/16/3/315/1493810/evco.2008.16.3.315.pdf.

26. 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. Thanks to Ingelin Steinsland for consulting on Bayesian statistics. 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.21436623`). 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.21436668`).

Competing interests

The authors declare that there are no competing interests.