An Improved Systematic Method for Constructing Enzyme-Constrained Genome-Scale Metabolic Models Using a Protein-Chemical Transformer

Anna Schooneveld [*1] **Shafiat Dewan** [*1] **Navot Arad** [2] **Sam Genway** [2] **Sonya Kaur Kalsi** [1] **Kotryna Bloznelyte** [1]
Will Addison [1] **David Berman** [3]

Abstract

Enzyme-constrained genome-scale metabolic models (ecGEMs) have improved Flux Balance Analysis (FBA) by incorporating enzyme turnover numbers (k_{cat}s). Since in-vivo k_{cat} data is costly to obtain and therefore scarce, we present a novel multi-modal transformer-based approach with cross-attention to predict k_{cat} values for *Escherichia coli* using enzyme amino acid sequences and SMILES annotations of reaction substrates. For heteromeric enzymes, we evaluate multiple subunit k_{cat} aggregation strategies. We benchmark ecGEMs constructed with these strategies against current state-of-the-art models using experimental growth rates, ^{13}C fluxes, and enzyme abundances, and prior to any calibration outperform or match existing methods. We also devise a new calibration method using flux control coefficients (derivatives of log flux with respect to log k_{cat}), which we show to be identical to enzyme cost at the FBA optimum. Using these coefficients, we identify 8 key k_{cat} values to recalibrate using experimental data, subsequently achieving superior performance to the current state-of-the-art with 81% fewer calibrations.

1. Introduction

In recent decades, systems biology has progressed significantly with the development of genome-scale metabolic models (GEMs; Feist & Palsson 2008; Oberhardt et al. 2009; Gu et al. 2019). These models describe the interplay between genes, metabolites, and reactions in a mathematical framework that can be solved via Flux Balance Analysis (FBA; Orth et al. 2010). However, this approach is not sophisticated enough to capture all the behaviour of real cells. For example, *E. coli* exhibits a lower growth rate than predicted by simple FBA (Mao et al., 2022). One of the most promising solutions to this challenge has come from enzyme-constrained models, such as GECKO (Sánchez et al., 2017; Domenzain et al., 2022; Chen et al., 2024), sMOMENT/AutoPACMEN (Bekiaris & Klamt, 2020), and ECMpy (Mao et al., 2022; 2024). These models account for the fact that a cell has limited chemical resources for enzyme production, which constrains enzyme abundances and consequently the flux solution space, leading to more accurate predictions (Mao et al., 2022; Massaiu et al., 2019).

The quality of any enzyme-constrained model is strongly dependent on the accuracy of its enzyme turnover rates (Sánchez et al., 2017; Li et al., 2022). The turnover rate, k_{cat}, of an enzyme is the number of substrate molecules catalysed by the enzyme per unit time. The literature can broadly be divided into two methods for obtaining k_{cat} values. Firstly, k_{cat}s can be retrieved from experimental databases such as BRENDA (Chang et al., 2021) and SABIO-RK (Wittig et al., 2018). This is the approach taken by GECKO, AutoPACMEN, and ECMpy v2. The main issue with this method is that the k_{cat} coverage in these databases is incomplete even for well-studied organisms[1], and very sparse or non-existent for less well-studied ones. Furthermore, measurements of k_{cat} values depend on experimental conditions, leading to non-uniformity between data from different studies.

With these issues in mind, a second approach uses machine learning to obtain k_{cat}s. Examples of this approach include Heckmann et al. (2018), used in ECMpy 1.0, and DLKcat (Li et al., 2022). The former employed various machine learning approaches, such as linear regression and deep learning, using features like flux and catalytic site information obtained from protein structures. DLKcat is more general, as it does not require such features, making it suitable

[*] Equal contribution [1] Cambridge Consultants, Cambridge, United Kingdom [2] Hybrid Intelligence, Capgemini Engineering, Stevenage, United Kingdom [3] School of Physics and Astronomy, Queen Mary University of London, London, United Kingdom. Correspondence to: Will Addison <will.addison@cambridgeconsultants.com>, David Berman <d.s.berman@qmul.ac.uk>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

[1] Often there are a large number of reactions for which no k_{cat}s are available and one needs to generate these in an ad-hoc fashion.

for less well-studied organisms. Instead, it requires inputs of SMILES strings for substrates (processed via a CNN), and amino acid sequences for enzymes (processed via a GNN). In the current work, we expand on this approach, presenting a novel transformer-based method (Vaswani et al., 2023) that takes SMILES strings and amino acid sequences as input. We demonstrate that our k_{cat} predictions outperform the state-of-the-art in terms of accuracy.

We apply our k_{cat} predictor to iML1515, the gold standard GEM for E. coli (Monk et al., 2017), to construct an enzyme-constrained GEM (ecGEM). Since existing k_{cat} predictors (including ours) only support monomeric reactions and the literature lacks consensus on how to combine subunit k_{cat} predictions for multimers, we evaluate the effect of multiple aggregation strategies on model performance through various benchmarks. These benchmarks also demonstrate that our model performs competitively with state-of-the-art approaches even before calibration.

Given the strong dependence of ecGEM performance on k_{cat} accuracy, calibration in the form of post-processing is often used to adjust k_{cat} using in vitro data. Generally enzyme cost is used to select reactions for calibration (Mao et al., 2022; 2024). We propose a more general alternative based on perturbative k_{cat} sensitivity analysis, using flux control coefficients (Kacser, 1973) to identify the most influential k_{cat} values. Although first disseminated in 1973, and later republished in 1995 (Kacser et al., 1995), flux control coefficients and metabolic control analysis have not been widely adopted for calibrating ecGEMs with FBA. We demonstrate that flux control coefficients are equivalent to enzyme costs in the case of an optimal FBA solution obeying enzyme constraints, thus providing the link between metabolic control analysis and enzyme costs determined through ecFBA solutions. Using our flux-control-coefficient-based method, we calibrate our ecGEM with significantly fewer ad-hoc k_{cat} corrections than other methods. The resulting ecGEM is on par with, or better than than existing approaches (Mao et al., 2022; 2024; Chen et al., 2024).

2. k_{cat} Prediction Transformer

This section details the data, architecture, and training process used to develop the k_{cat} model. The aim was to create a model that could predict the k_{cat} value given SMILES strings of the substrates and amino acids of the enzyme that catalyses the reaction.

2.1. Model Architecture

At a high level, our model architecture broadly follows the structure of DLKcat. It consists of three sub-models: two generate embeddings from substrates (as SMILES strings) and enzymes (as amino acid sequences), and a third predicts k_{cat} using these embeddings. Figure 1 shows a schematic representation. We introduce two major changes from DLKcat's approach: using pre-trained transformer-based models for embeddings (DLKcat trained a GNN for substrates and CNN for proteins), and introducing a custom 2-way cross-attention mechanism in the third sub-model.

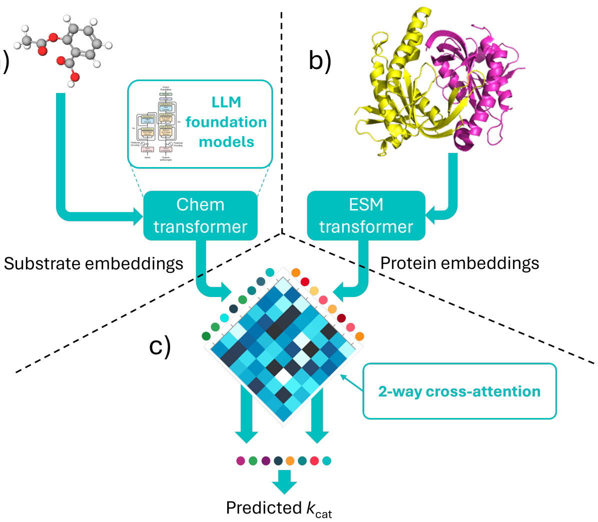

Figure 1. k_{cat} model architecture a) Pre-trained foundation model to generate molecule features. b) Pre-trained foundation model to generate protein features. c) 2-way cross-attention model to predict k_{cat} values

Our model requires two inputs. The first is a set of reaction substrates, each represented as a SMILES string (Weininger, 1988), concatenated into a single string using the non-bond token ".", and passed to the Chem transformer (Chanda (2021); Figure 1a) to generate substrate embeddings. The second input is the enzyme's amino acid sequence, which is passed to the ESM transformer (Lin et al., 2023) to produce protein embeddings. We removed the final projection layer from the Chem and ESM transformers, instead using their last hidden layers as input to the third sub-model, as this model requires embeddings (rather than the SMILES strings or amino acids output by the projection layer). For an input sequence of length N the generated embedding will be a matrix of size (N, E), where E is the fixed sized of the embedding.

The third sub-model implements a cross-attention mechanism between the substrate and protein embeddings produced by the Chem and ESM transformers, respectively. A complication with cross-attention is that it lacks order invariance, because the query (Q) matrix is derived from one input while the key (K) and value (V) matrices are derived from the other, meaning the directionality of the attention (e.g., protein-to-substrate vs. substrate-to-protein) affects the output. Given that there is no natural "order" between the a protein and its substrates, the model implements 2-way cross-attention (see Figure 2) which is invariant to permutations of the substrate and protein embeddings.

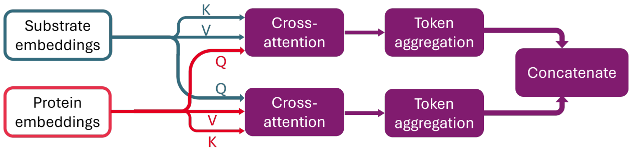

Figure 2. 2-way cross-attention model architecture.

Following the cross-attention block in each branch, a token aggregation step is required to ensure that predictions are made over the complete sequence of the inputs, rather than individual tokens. We implemented three aggregation methods: (1) compute mean aggregation over the token embedding (2) take the embedding vector of the first token in the sequence (3) use a fuzzy membership function of a Gaussian mixture distribution and average memberships across the token dimension. Methods (1) and (2) produce an output vector of length E, (3) produces an output matrix with shape (D, E) where D is the number of 1-D distributions in the mixture. The choice of method is regarded as a hyperparameter.

The aggregated outputs from each branch are then concatenated and passed to the final projection layer to make the k_{cat} prediction (see Figure 2). All models were built using Python 3.9, PyTorch 1.13.1 (Paszke et al., 2019), and Lightning 2.0.3 (Falcon & The PyTorch Lightning team, 2019).

2.2. Data

To train the k_{cat} model, each data point must include a substrate set represented as SMILES strings, an enzyme given as an amino acid sequence, and a corresponding measured k_{cat} value. To obtain training data, measured k_{cat} values were collected from BRENDA (Chang et al., 2021) and SABIO-RK (Wittig et al., 2018). For reactions in BRENDA, substrate SMILES strings were retrieved programmatically from PubChem (Kim et al., 2024) via chemical name, and amino acid sequences from Uniprot by accession ID. For SABIO-RK, substrate SMILES strings were already present, and amino acid sequences were programmatically retrieved identically to BRENDA.

This training data has an issue of degeneracy at multiple levels. The first issue lies in molecular representation: a single molecule can have multiple valid SMILES strings. To address this, we standardised all SMILES strings by sanitizing with RDKit's (Landrum et al., 2022) SanitizeMol function, removing isotopes, neutralizing charges, stripping stereochemistry, and converting to and from InChI to ensure tautomerism consistency. The second issue is that a single reaction can have multiple k_{cat} measurements. To resolve this, only the maximum k_{cat} is kept per unique reaction. Two data points are considered the same reaction if, after SMILES standardisation, they have identical substrates, products, and enzyme. The final degeneracy involves substrate ordering. Since reactions often have multiple substrates, their SMILES strings are concatenated before being input to the model. This introduces ambiguity in how to choose the order of concatenation. To address this, the dataset was augmented by including all possible permutations of substrate SMILES as distinct data points.

Before the final augmentation step, the training data was filtered to exclude entries with combined substrate SMILES over 510 tokens or amino acid sequences over 1000 characters, to accommodate the context limit of the Chem and ESM transformer respectively. To rebalance the data, k_{cat} values above 5000 were also discarded. This exluded only 1% of entries in the unfiltered dataset, which, despite spanning k_{cat} values of $0 - 10^8$, is heavily skewed towards smaller values.

The final dataset contained 35,499 entries, split into 65% train, 15% validation, and 20% test. Splitting was done prior to substrate permutation to avoid information leakage. It is important to note that, both the data and model architecture only support reactions catalysed by monomers; predictions for heteromers or homomultimers are not supported. The issue of how to deal with multimers will be revisited in Sections 4 and 3

2.3. Training and evaluation

During training, k_{cat} values are transformed via $y = \ln(k_{cat} + 1)$ to account for the logarithmic distribution of k_{cat}s, and to prevent large values from dominating the loss. The addition of 1 ensures stability for small or zero values. As a result, the model also predicts in log-space ($\hat{y}$; see Figure 3), and the loss is computed between y and $\hat{y}$. The predicted value $\hat{k}_{cat}$ is retrieved by applying the inverse transformation $\hat{k}_{cat} = e^{\hat{y}} - 1$.

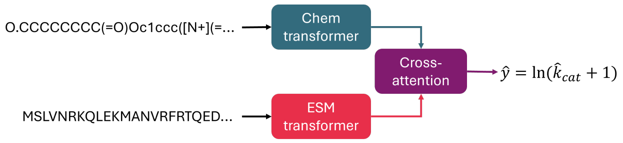
O.CCCCCCCC(=O)Oc1ccc([N+](=...
MSLVNRKQLEKMANVRFRTQED...
Chem transformer
ESM transformer
Cross-attention
$\hat{y} = \ln(\hat{k}_{cat} + 1)$

Figure 3. k_{cat} model inference data flow

Hyperparameter tuning was done using a Kubernetes cluster, with each hyperparameter set (a single model) trained on a single Nvidia V100 GPU. Every model was trained on the same training partition and evaluated on the validation set at the end of each epoch. The models were trained using the esm2_t33_650M_UR50D model variant for the ESM transformer, Mean Squared Error (MSE) loss, and the Adam optimizer (Kingma & Ba, 2017).

The best model, as selected by the minimal validation loss, used mean token aggregation, a dropout of 0.2, 8 attention heads in each of the input branches of the cross-attention sub-model, a fixed learning rate of 10^{-3}, and trained for 26 epochs with early stopping. This model took 24 hours to train. The selected model was evaluated on the held-out test partition, with 90% of predicted values being correct within 1 order of magnitude.

3. The Enzyme-Constrained Genome-Scale Metabolic Model

This section details the process of how our enzyme-constrained models were constructed. We constructed models for use with Python 3.10, COBRApy (Ebrahim et al., 2013).

3.1. Creating an enzyme-constrained model

As a base model, we use iML1515 (Monk et al., 2017), more specifically, the modified version in ECMpy (Mao et al., 2022), which corrects some minor known errors in iML1515. To make this model enzyme constrained, we follow and extend the approach in Mao et al. (2022).

To ensure that each enzyme-reaction pair has a unique k_{cat}, we split reversible reactions into separate forward and backward reactions and split isoenzyme-catalysed reactions into different reactions. Unlike Mao et al. (2022), we add reverse reactions to all exchange reactions with a default lower bound of zero, to allow the intake of metabolites without requiring negative fluxes.

A GEM becomes an enzyme constraint model if, in addition to the standard FBA constraints,

$$\begin{aligned} &\text{maximize} && v_{objective} && (1)\\ &\text{subject to} && S\mathbf{v} = \mathbf{0} && (2)\\ &\text{and} && \mathbf{lowerbound} \leq \mathbf{v} \leq \mathbf{upperbound}. && (3) \end{aligned}$$

The model also obeys a total enzyme constraint

$$\sum_{i=1}^{n} \frac{v_i MW_i}{\sigma_i k_{cat,i}} \leq p_{tot} f. \tag{4}$$

Here, $v_{objective}$ is the objective flux (usually growth rate), $\mathbf{v}$ is a flux vector with elements v_i representing reactions (each with a lower and upper bound). S is the stoichiometric matrix, n is the total number of reactions, and MW_i, $k_{cat,i}$, and σ_i are the molecular weight, turnover number, and saturation coefficient of the enzyme catalysing a reaction. p_{tot} and f denote the total protein fraction and the enzyme mass fraction. Throughout this work, we assume $\sigma_{r,i} = 1$, consistent with (Mao et al., 2022; Bekiaris & Klamt, 2020). We set $p_{tot} = 0.56 \text{ g gDW}^{-1}$ based on experimental data (Bremer & Dennis, 2008; Brunk et al., 2016), and compute f as described in (Mao et al., 2022).[2]

A downside of the ECMpy approach is its incompatibility with major metabolic engineering packages like OptKnock (Burgard et al., 2003). Therefore, we adopt the method from Bekiaris & Klamt (2020), implementing the enzyme constraint via a pseudo-metabolite and pseudo-reaction. An "enzyme pool" pseudo-metabolite is added to each reaction i with a stoichiometric coefficient of $-\frac{MW_i}{k_{cat,i}}$. A pseudo-reaction is then added with this pseudo-metabolite as its only metabolite with an upper bound of $P = p_{tot} f$. This is mathematically equivalent to Equation 4 (Bekiaris & Klamt, 2020). We then solved this system of equations via linear programming using COBRApy.

3.2. Annotations

To obtain the information we needed to implement the enzyme constraint (i.e. MW_i and $k_{cat,i}$), we added additional reaction and metabolite annotations to our model. Amino acid sequences for reactions were obtained from the BiGG database, and SMILES strings for metabolites from MetaNetX. Deprecated MetaNetX annotations were manually updated using PubChem for some metabolites. Molecular weights and subunit information for genes were gathered from UniProt. As transport reactions require special treatment when setting k_{cat} values (see Section 3.3), we also annotated all transport reactions in the model. Appendix A provides more detail about the annotation process.

3.3. Determining k_{cat} values

Using these annotations, we passed amino acid sequences and SMILES strings for each non-transport reaction in our ecGEM to the k_{cat} predictor from Section 2. We did this for each enzyme subunit in the relevant reactions individually, predicting a k_{cat} value for each subunit. As monomers have only one subunit, this will be the final k_{cat} value. For multimeric reactions, i.e. those with multiple subunits, we came up with a strategy detailed below.

Our k_{cat} predictor does not produce meaningful predic-

[2] This treatment of p_{tot} and f assumes constant protein mass for enzyme production across different conditions (e.g., media), a simplification which may not hold true in real-life scenarios.

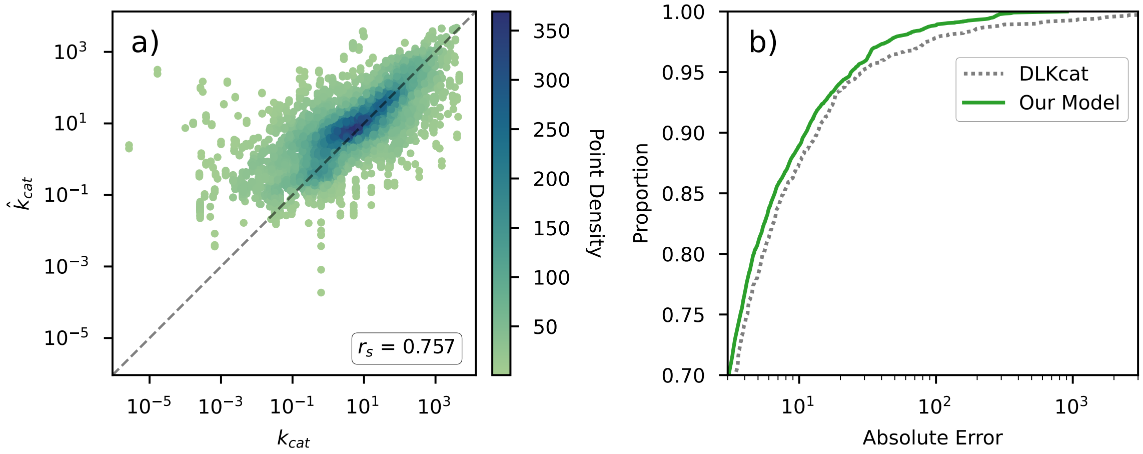

Figure 4. k_{cat} model performance. a) log-log scatter plot showing measured k_{cat} vs predicted $\hat{k}_{cat}$. Spearman's rank correlation coefficient $r_s = 0.757$ for our model, compared to $r_s = 0.720$ for DLKcat. b) Comparison of the error in model predictions between our model and DLKcat. At each point on the x-axis, the curves show the proportion of test set predictions with the same cumulative error or less. Our model makes notably fewer predictions that have large errors.

tions for transport reactions, as these are heavily underrepresented in the training data. The BRENDA database contains k_{cat} values for only 31 out of 97 transport reactions in EC class 7 "Translocases". The literature does not describe a principled method for handling transport k_{cat}s. Therefore, we follow the common convention to use $k_{cat} = 234,000\,h^{-1}$ (Corrao et al., 2024; Heckmann et al., 2018). Appendix B provides further details on the exact heuristics we used.

Reaction statistics and a brief discussion of these are provided in Table 3 and Appendix C. Our k_{cat} coverage is extensive - out of 5338 reactions with genes (the required input for our ML model), 5323 have a k_{cat} value.

3.3.1. Calculating k_{cat} for Multimeric Enzymes

The literature on how to handle the choice of combining enzyme turnover rates for homomultimers and heteromers is sparse, and there is no clear consensus. ECMpy uses the k_{cat} with $min(\frac{k_{cat,ij}}{MW_{ij}}, j \in m)$, where m is the number of proteins in the complex (Mao et al., 2022), and DLKcat uses the maximum predicted k_{cat} value (Li et al., 2022). No rigorous derivation is given to explain why these heuristics were chosen.

However, rigorous handling of multimeric enzymes is crucial, as a non-neglicible portion of reactions are multimeric. In eciML1515, 46.5% of the 6191 reactions are homomultimeric, 32.5% are monomeric, 7% are heteromeric, and 14% are non-enzymatic (see Table 3). We address the calculation of k_{cat} for homomultimers and heteromultimers by evaluating models using various methods. For homomultimers comprised of n subunits, we calculate k_{cat} for the multimer as $n \times k_{cat}$ for the monomer[3]. For heteromers, we combine the k_{cat} value for each of the monomer/homomultimer present in the complex:

$$k_{cat} = agg(k_{cat,A}, k_{cat,B}, ...) \tag{5}$$

where agg is the minimum, maximum, average, or average weighted by the number of subunits. We produce one ecGEM for each aggregation method.

3.3.2. k_{cat} Calibration Targets via Flux Control Coefficients

It is inevitable that some k_{cat}s are inaccurate due to machine learning errors or issues in handling multimers and transport reactions. To improve our enzyme-constrained model, we aim to select reactions for experimental calibration.

To understand which k_{cat} values most influence our ecGEM's predictions, we quantified the sensitivity of target flux to individual enzyme turnover rates. Sensitivity is the magnitude of the derivative $\frac{\partial v_T}{\partial k_{cat,i}}$, where v_T is the target flux and i is the enzyme-reaction pair. For each pair, we use FBA to solve for a target flux (e.g., biomass), apply a small perturbation $\delta\bar{k}_{cat}$ to $\bar{k}_{cat,i}$ (where $\bar{k}_{cat,i} = \frac{k_{cat,i}}{MW_i}$), and re-solve the FBA model to compute the derivative as the ratio of flux change to $\delta\bar{k}_{cat}$. Finally, we rescale the derivative to obtain a dimensionless quantity:

[3] There is a question around whether the true turnover rate would be higher than this, since evolutionarily speaking, the mean reason for forming a multimer would be to reduce the reaction's activation energy, but addressing this issue is beyond the scope of our work.

$$C^{v_t}_{k_{cat,i}} = \frac{\delta v_T}{\delta \bar{k}_{cat,i}} \cdot \frac{\bar{k}_{kcat,i}}{v_T} \,. \tag{6}$$

$C^{v_t}_{kcat,i}$ indicates the relative change in target flux when the k_{cat} of the ith reaction changes, known in metabolic control analysis as the flux control coefficient (Kacser, 1973; Kacser et al., 1995). A high flux control coefficient means the target flux is highly sensitive to that reaction. This analysis therefore identifies reactions that bottleneck the target flux. High $C^{v_t}_{kcat,i}$ genes highlight the most efficient improvements to a metabolic model's predictive power from lab measurements. When measuring k_{cat} in vivo, this informs where to invest lab effort to manually calibrate and improve the model. This is useful for metabolic engineering, guiding the choice of proteins/enzymes to improve through protein engineering or identifying genes for upregulation via promoter engineering. Additionally, it benefits general model accuracy. The top N k_{cat} values from this analysis indicate the enzymes that best explain variance in model performance. This reduces the problem space from over 6,000 reactions to tens or hundreds. Combined with tools like metabolic flux visualizations (Figure 6, and 7), this helps identify key pathways that enhance model accuracy or target flux.

To our knowledge, perturbative k_{cat} analysis using flux control coefficients has not been previously applied to ecGEM calibration in the literature. ECMpy v2 (Mao et al., 2024) provide a related approach, but they iteratively calibrate based on enzyme cost ($\frac{v_i MW_i}{\sigma_i k_{cat,i}}$). Notably, for an optimised ecGEM, the flux control coefficient is mathematically equivalent to enzyme cost. A detailed proof and further discussion are provided in Appendix E. A key advantage of using flux control coefficients is their generality: they will inform us of flux bottlenecks even when additional constraints are introduced to the model.

4. Results

4.1. Computing benchmarks

We captured the performance of our ecGEMs using a selection of benchmarks outlined below, and compared our models with reference models from the literature. Optimisation was performed via pFBA (Lewis et al., 2010) with CPLEX.

- **Glucose growth rate.** This benchmark compares the model growth rate on a glucose substrate to the experimental value (0.66 h^{-1} (Adadi et al., 2012)). It is given as a signed percentage difference.

- **Flux comparison to measured ^{13}C fluxes (RMSE).** This benchmark compares model fluxes to measured ^{13}C fluxes (Okahashi et al., 2014) via the Root Mean Squared Error (RMSE). We sum isoenzyme-catalysed reactions and subtract reverse fluxes.

- **Growth rate on different substrates.** This benchmark compares the simulated growth rate on 24 different carbon sources to existing experimental measurements (Adadi et al., 2012), again using the RMSE.

- **Enzyme abundances.** This benchmark compares predicted enzyme abundances in our model to enzyme abundances from Corrao et al. (2024), via the root mean squared logarithmic error (RMSLE). Further detail about this abundance analysis is provided in Appendix F.

4.2. Models to evaluate

We produced four uncalibrated versions of eciML1515 (min, max, avg, wavg), one for each k_{cat} aggregation method from Section 3.3.1. We also create min_4_clbr and min_8_clbr, which are the min model where four or eight reactions are calibrated respectively (see Section 4.5 for more detail).

We benchmark against several state-of-the-art enzyme-constrained models from the literature (Mao et al., 2022; 2024). More detail on the specifics of these models can be found in Appendix G

- **ECMpy_ML** is the ECMpy v1 model without any k_{cat} calibration; it uses machine learning-derived k_{cat} values from (Heckmann et al., 2018).

- **ECMpy_expmnt** is the ECMpy v2 model without calibration, made up of experimental k_{cat} values from BRENDA and SABIO-RK. An average k_{cat} is used for reactions not in the databases.

- **ECMpy_DLKcat** was created via the DLKcat pipeline in ECMpy v2. This pipeline produced NaNs for 15% of k_{cat}s due to SMILES retrieval issues; we left those reactions unconstrained. Note therefore that results for this model reflect the combined ECMpy+DLKcat performance rather than DLKcat alone.

- **ECMpy_ML_clbr** is ECMpy_ML after calibration, where 14 reactions with high enzyme cost were updated using experimental k_{cat}s from BRENDA and SABIO-RK.

- **ECMpy_expmnt_clbr** is ECMpy_expmnt after the v2 calibration process, which consists 50 rounds of iteratively replacing high enzyme cost reactions with the largest available experimental value from BRENDA or SABIO-RK.

4.3. Our k_{cat} predictor creates more accurate predictions of metabolism

We first run the benchmarking on all the uncalibrated models. The results are shown in Table 1. Without any calibration, the performance of our models is generally superior to the benchmark models. Our models are the highest scoring in terms of glucose growth, ^{13}C, and substrate RMSE. The only area where a model from the literature is superior is for the abundance analysis, where ECMpy_ML takes the top spot. The fact that our models are overall the best performing on most benchmarks suggests that our k_{cat} estimates are superior to the current state-of-the-art, and would provide a better basis for an ecGEM prior to any calibration.

Note also that overall, our ecGEMs outperform the ECMpy_DLKcat model. This confirms that the improved performance of our k_{cat} predictor over DLKcat as noted in Figure 4 also translates to measurable improvements in model performance at the ecGEM scale.

4.4. Uncovering sensitivity to modelling of heteromers

Our ecGEM performance also varies substantially depending on the method we use for aggregating k_{cat} predictions for heteromers, and other than the fact that the min model generally performs worst, there is no one method that is clearly superior to the others. This is an important finding, as it shows that the choice of aggregation strategy can have a considerable impact on ecGEM quality. As described in Section 3.3.1, the current literature lacks a cohesive and considered approach to this decision. However, our results imply that it is an important factor for ecGEM performance. We therefore strongly recommend more research into this issue in order for machine learning k_{cat}s to reach their full potential.

4.5. Model improvement with fewer calibrations

As our min model had the worst growth rate error on glucose (57.2%), we selected this model for calibration. Using our method from Section 3.3.2, we computed the flux control coefficients $C^{v_t}_{k_{cat,i}}$ (see Figure 8) to identify the top 10 most sensitive reactions as calibration targets to improve (See Table 4). Ideally, we would have been able to measure k_{cat} for these 10 reactions in vivo, but we did not have the required laboratory resources. Instead, consistent with Mao et al. (2022; 2024), we replaced the k_{cat} value with the highest reported k_{cat} from BRENDA (Chang et al., 2021) and SABIO-RK (Wittig et al., 2018) via the EC number(s).

To quantify how many reactions we would need to adjust to achieve improvements in ecGEM performance, we created ecGEMs with between 1 and 10 k_{cat} values calibrated, and computed their glucose growth rate error values. Figure 5 shows the improvement in glucose growth rate error when successive k_{cat} values (ranked by k_{cat} sensitivity) are adjusted cumulatively. Substantial improvements are made with just 4 reactions adjusted, and the improvement plateaus after just 8. We compare these two models (min_4_clbr and min_8_clbr) to similar models from the literature in Table 2.

Table 1. Benchmarking results of uncalibrated ecGEMs against experimental data from the literature for growth rates, ^{13}C fluxes, and enzyme abundances. A description of how these metrics were computed can be found in Section 4.1. Glucose growth is given as a signed percentage difference between simulated and experimental growth rate. These models have not undergone any k_{cat} calibration, see Section 4.2 and Appendix G for details on how these models were obtained. The best performing result for each benchmark is indicated in **bold**.

ecGEM	Glucose growth % error	^{13}C RMSE	Various substrate growth RMSE	Abundance RMSLE glucose $\times 10^{-3}$
min	-57.2	6.18	0.31	6.67
max	29.4	**2.47**	0.31	6.50
avg	**12.3**	2.70	0.22	4.58
wavg	-14.3	9.17	**0.13**	6.16
ECMpy_ML	-44.2	4.12	0.20	**3.41**
ECMpy_exp	-73.7	4.42	0.34	6.93
ECMpy_DLKcat	-45.4	5.13	0.24	6.45

Table 2. Benchmarking results of calibrated ecGEMs against experimental data from the literature for growth rates, ^{13}C fluxes, and enzyme abundances. A description of how these metrics were computed can be found in Section 4.1. Glucose growth is given as a percentage difference between simulated and experimental growth rate. These models have been optimised with k_{cat} calibration, see Section 4.2 for details on how the ECMpy models were obtained, and Section 4.5 for the creation of the min_4_clbr and min_8_clbr models. The best performing result for each benchmark is indicated in **bold**.

ecGEM	Num reactions calibrated	Glucose growth % error	^{13}C RMSE	Various substrate growth RMSE	Abundance RMSLE glucose $\times 10^{-3}$
min_4_clbr	4	-15.2	3.40	0.15	8.06
min_8_clbr	8	**-13.5**	**2.90**	0.16	8.26
ECMpy_exp_clbr	43	-24.3	5.05	0.15	13.21
ECMpy_ML_clbr	14	-15.2	4.47	**0.14**	**3.82**

Calibration substantially improves our min model across most metrics — adjusting just four reactions yields comparable performance to ECMpy_ML_clbr, which calibrated 14 k_{cat}s. Increasing the number of calibrated reactions to eight leads to the lowest growth rate error among all calibrated models, in fewer calibrations than the benchmark models. This model achieves an reduction in glucose growth-rate error from -57.2% to -13.5% (Figure 5) and RMSE on ^{13}C flux data from 6.18 to 2.90 (Tables 1, 2). The significant reduction in growth rate error only occurs when calibrated k_{cat} values are applied cumulatively[4], not separately, as shown in see Figure 5.

We identify ATPS4rpp_num2 as a major biomass flux bottleneck - calibrating its k_{cat} reduces the glucose growth rate error by 40%. Our predicted k_{cat} differs from the experimental value in BRENDA by two orders of magnitude (Table 4). The reaction is catalysed by an eight-subunit enzyme complex (Moore et al., 2024). Our k_{cat} predictor, trained only on monomers, outputs values spanning four orders of magnitude, and for heteromers, our simple aggregation method is highly sensitive to outliers. Notably, 18 of the top 100 reactions ranked by $C^{v_t}kcat, i$ are heteromeric (Figure 9). This is a substantial amount, given that only 7% of all reactions in our ecGEM are heteromeric (Table 3). For all multimers combined (including homomultimers), 81 top 100 reactions are multimeric, compared to 54% of all ecGEM reactions, Together, all these observations highlight the need for a k_{cat} predictor that can directly estimate turnover numbers for heteromeric complexes, and multimers in general.

Our calibration targets for the min model partially overlap with ECMpy v1 (Mao et al., 2022), sharing two round 1 enzyme usage corrections and one round 2 ^{13}C flux correction. However, we need just 4 calibrations (Figure 5) to reduce glucose growth rate error to 15% whereas ECMpy v1 uses 14 and ECMpy v2 uses 43. Given that our method is mathematically equivalent to theirs (as shown in Appendix E), the fact that we need fewer calibrations implies that our initial k_{cat} estimates must be more accurate, highlighting the accuracy of our machine learning predictor.

5. Conclusion

Using the novel transformer-based k_{cat} predictor introduced in this work, we produced an enzyme constrained genome scale model which outperforms the current state-of-the-art for ecGEMs that have not been calibrated using lab data. Furthermore, with our k_{cat} sensitivity analysis, we have devised a rigorous way to identify which k_{cat}s would benefit the most from calibration with lab data. When applied to the min model, our worst performing uncalibrated model, this process makes our model competitive with the best calibrated models in the literature, but by calibrating 81% fewer k_{cat} values. Thus, our findings add to the growing body of evidence (Li et al., 2022; Heckmann et al., 2018) that machine learning for predicting turnover numbers is a core part of future systems biology.

However, to produce optimal machine learning k_{cat}s, we have found that it is important to have a well founded method for estimating k_{cat}s for multimers. This is currently an under-explored area of study, and different research groups adopt different methods without a clear justification. Our findings show that ecGEM performance can vary significantly depending on the method used for aggregating k_{cat} values for heteromeric enzymes, and this choice therefore warrants further research. We emphasise that any machine learning predictor for enzyme turnover rates for heteromers and homomultimers needs to be more advanced than the currently available methods. It is likely that a single uniform aggregation method across all reactions in a given model is too simplistic. The function of subunits and the way in which they combine to form an enzyme complex varies significantly from enzyme to enzyme, and it seems biologically likely that the optimal aggregation method might differ per enzyme and therefore require a more complex method than a simple max, min, avg, or wavg operation. Multimeric reactions make up 81 of the top 100 most senstive reactions (Figure 9), so getting these k_{cat}s right is crucial.

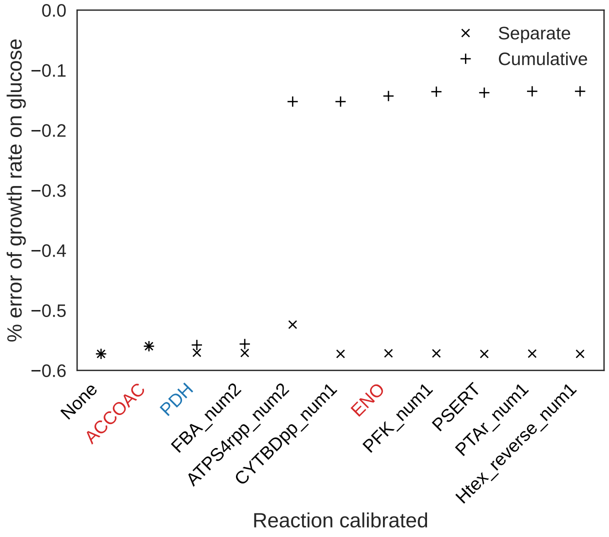

Figure 5. Calibrated min model with reactions calibrated separately and cumulatively (left to right). Red highlighting corresponds to ECMpy v1 round 1 enzyme abundance corrections, Blue highlighting corresponds to ECMpy v1 round 2 ^{13}C corrections.

Therefore, to imbibe as much biological context as possible, an improved future k_{cat} predictor should be trained on a dataset that includes many more examples of reactions catalysed by homomultimers and heteromers and have an architecture that supports multimeric inputs. A second direction of improvement would center around enhanced capabilities for dealing with transport and exchange reactions. To do this, the model would have to encounter a substantial number of these reactions during training.

[4] We only perturb a single k_{cat} at a time, as combinations of perturbations would lead to combinatorial explosions.

However, no matter how advanced the method for estimating k_{cat}, that there will be inaccuracies in estimates/predictions, whether from deep learning or otherwise. So, in order to refine an ecGEM, there needs to be a rigorous calibration method used. We have shown that the method of calibration via per-reaction enzyme cost calculations is mathematically equivalent to calculating the flux control coefficient per reaction. Furthermore, we don't expect this equivalence to hold when additional constraints are added to the optimisation. However our method of calculating the flux control coefficient through perturbing individual k_{cat} will generalise to any ecGEM.

Acknowledgements

We thank the reviewers for their feedback on the submitted version of the manuscript. We are grateful to Jakub Chromý, Luca Rossoni, and Ian Taylor for their expert input on metabolic engineering. We also thank Sadhira Wagiswara for creating the visualisations of our metabolic networks. Finally, we thank Chris Roberts, Frances Metcalfe, Richard Snell, Sally Epstein and those at Capgemini Invent for sponsoring and supporting this project and the subsequent publishing effort.

Impact Statement

Improved metabolic modelling of organisms like *Escherichia coli* requires information about enzyme dynamics. We present a novel, transformer-based machine learning architecture with two-way cross attention to predict enzyme turnover numbers to present a state-of-the-art method for constructing metabolic models. Our insights will provide a deeper understanding of constructing and evaluating metabolic models in the field of Metabolic Engineering.

References

Adadi, R., Volkmer, B., Milo, R., Heinemann, M., and Shlomi, T. Prediction of microbial growth rate versus biomass yield by a metabolic network with kinetic parameters. *PLOS Computational Biology*, 8(7):1–9, 07 2012. doi: 10.1371/journal.pcbi.1002575. URL https://doi.org/10.1371/journal.pcbi.1002575.

Bekiaris, P. S. and Klamt, S. Automatic construction of metabolic models with enzyme constraints. *BMC Bioinformatics*, 21(1):19, January 2020.

Bernard, T., Bridge, A., Morgat, A., Moretti, S., Xenarios, I., and Pagni, M. Reconciliation of metabolites and biochemical reactions for metabolic networks. *Briefings in Bioinformatics*, 15(1):123–135, 11 2012. ISSN 1467-5463. doi: 10.1093/bib/bbs058. URL https://doi.org/10.1093/bib/bbs058.

Bremer, H. and Dennis, P. P. Modulation of chemical composition and other parameters of the cell at different exponential growth rates. *Ecosal plus*, 3(1):10–1128, 2008.

Brunk, E., Mih, N., Monk, J., Zhang, Z., O'Brien, E. J., Bliven, S. E., Chen, K., Chang, R. L., Bourne, P. E., and Palsson, B. O. Systems biology of the structural proteome. *BMC systems biology*, 10:1–16, 2016.

Burgard, A. P., Pharkya, P., and Maranas, C. D. Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol. Bioeng.*, 84(6):647–657, December 2003.

Chanda, P. Pre-trained chemical language model. https://huggingface.co/pchanda/pretrained-smiles-pubchem10m, 2021. Accessed: 2023-12-14.

Chang, A., Jeske, L., Ulbrich, S., Hofmann, J., Koblitz, J., Schomburg, I., Neumann-Schaal, M., Jahn, D., and Schomburg, D. BRENDA, the ELIXIR core data resource in 2021: new developments and updates. *Nucleic Acids Res.*, 49(D1):D498–D508, January 2021.

Chen, Y., Gustafsson, J., Tafur Rangel, A., Anton, M., Domenzain, I., Kittikunapong, C., Li, F., Yuan, L., Nielsen, J., and Kerkhoven, E. J. Reconstruction, simulation and analysis of enzyme-constrained metabolic models using GECKO toolbox 3.0. *Nat. Protoc.*, 19(3): 629–667, March 2024.

Consortium, T. U. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1): D523–D531, 11 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac1052. URL https://doi.org/10.1093/nar/gkac1052.

Corrao, M., He, H., Liebermeister, W., Noor, E., and Bar-Even, A. A compact model of escherichia coli core and biosynthetic metabolism, 2024. URL https://arxiv.org/abs/2406.16596.

Domenzain, I., Sánchez, B., Anton, M., Kerkhoven, E. J., Millán-Oropeza, A., Henry, C., Siewers, V., Morrissey, J. P., Sonnenschein, N., and Nielsen, J. Reconstruction of a catalogue of genome-scale metabolic models with enzymatic constraints using GECKO 2.0. *Nat. Commun.*, 13(1):3766, June 2022.

Ebrahim, A., Lerman, J. A., Palsson, B. O., and Hyduke, D. R. Cobrapy: Constraints-based reconstruction and analysis for python. *BMC Systems Biology*, 7(1):74, Aug 2013. ISSN 1752-0509. doi: 10.1186/1752-0509-7-74. URL https://doi.org/10.1186/1752-0509-7-74.

Falcon, W. and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL https://github.com/Lightning-AI/lightning.

Feist, A. M. and Palsson, B. Ø. The growing scope of applications of genome-scale metabolic reconstructions using escherichia coli. *Nat. Biotechnol.*, 26(6):659–667, June 2008.

Ganter, M., Bernard, T., Moretti, S., Stelling, J., and Pagni, M. Metanetx.org: a website and repository for accessing, analysing and manipulating metabolic networks. *Bioinformatics*, 29(6):815–816, 01 2013. ISSN 1367-4803. doi: 10.1093/bioinformatics/btt036. URL https://doi.org/10.1093/bioinformatics/btt036.

Gu, C., Kim, G. B., Kim, W. J., Kim, H. U., and Lee, S. Y. Current status and applications of genome-scale metabolic models. *Genome Biol.*, 20(1):121, June 2019.

Heckmann, D., Lloyd, C. J., Mih, N., Ha, Y., Zielinski, D. C., Haiman, Z. B., Desouki, A. A., Lercher, M. J., and Palsson, B. O. Machine learning applied to enzyme turnover numbers reveals protein structural correlates and improves metabolic models. *Nature Communications*, 9 (1):5252, Dec 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-07652-6. URL https://doi.org/10.1038/s41467-018-07652-6.

Heckmann, D., Campeau, A., Lloyd, C. J., Phaneuf, P. V., Hefner, Y., Carrillo-Terrazas, M., Feist, A. M., Gonzalez, D. J., and Palsson, B. O. Kinetic profiling of metabolic specialists demonstrates stability and consistency of in vivo enzyme turnover numbers. *Proc. Natl. Acad. Sci. U. S. A.*, 117(37):23182–23190, September 2020.

Kacser, H. The control of flux. In *Symp Soc Exp Biol*, volume 27, pp. 65, 1973.

Kacser, H., Burns, J. A., Kacser, H., and Fell, D. A. The control of flux. *Biochemical Society Transactions*, 23 (2):341–366, 05 1995. ISSN 0300-5127. doi: 10.1042/bst0230341. URL https://doi.org/10.1042/bst0230341.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B., Thiessen, P., Yu, B., Zaslavsky, L., Zhang, J., and Bolton, E. Pubchem 2025 update. *Nucleic Acids Research*, 53(D1):D1516–D1525, 11 2024. ISSN 1362-4962. doi: 10.1093/nar/gkae1059. URL https://doi.org/10.1093/nar/gkae1059.

King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., Ebrahim, A., Palsson, B. O., and Lewis, N. E. Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research*, 44(D1):D515–D522, 10 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv1049. URL https://doi.org/10.1093/nar/gkv1049.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Landrum, G., Tosco, P., Kelley, B., Ric, sriniker, gedeck, Vianello, R., NadineSchneider, Kawashima, E., Cosgrove, D., Dalke, A., Dan, N., Jones, G., Cole, B., Swain, M., Turk, S., AlexanderSavelyev, Vaucher, A., Wójcikowski, M., Take, I., Probst, D., Ujihara, K., Scalfani, V. F., Godin, G., Pahl, A., Berenger, F., JLVarjo, strets, JP, and DoliathGavid. rdkit/rdkit: 2022_03_5 (q1 2022) release, 2022.

Lewis, N. E., Hixson, K. K., Conrad, T. M., Lerman, J. A., Charusanti, P., Polpitiya, A. D., Adkins, J. N., Schramm, G., Purvine, S. O., Lopez-Ferrer, D., Weitz, K. K., Eils, R., König, R., Smith, R. D., and Palsson, B. O. Omic data from evolved ¡i¿e. coli¡/i¿ are consistent with computed optimal growth from genome&#x2010;scale models. *Molecular Systems Biology*, 6(1):390, 2010. doi: https://doi.org/10.1038/msb.2010.47. URL https://www.embopress.org/doi/abs/10.1038/msb.2010.47.

Li, F., Yuan, L., Lu, H., Li, G., Chen, Y., Engqvist, M. K. M., Kerkhoven, E. J., and Nielsen, J. Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction. *Nature Catalysis*, 5 (8):662–672, Aug 2022. ISSN 2520-1158. doi: 10.1038/s41929-022-00798-z. URL https://doi.org/10.1038/s41929-022-00798-z.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/abs/10.1126/science.ade2574. Earlier versions as preprint: bioRxiv 2022.07.20.500902.

Mao, Z., Zhao, X., Yang, X., Zhang, P., Du, J., Yuan, Q., and Ma, H. Ecmpy, a simplified workflow for constructing enzymatic constrained metabolic network model. *Biomolecules*, 12(1), 2022. ISSN 2218-273X. doi: 10.3390/biom12010065. URL https://www.mdpi.com/2218-273X/12/1/65.

Mao, Z., Niu, J., Zhao, J., Huang, Y., Wu, K., Yun, L., Guan, J., Yuan, Q., Liao, X., Wang, Z., and Ma, H. ECMpy 2.0: A python package for automated construction and analysis of enzyme-constrained models. *Synth. Syst. Biotechnol.*, 9(3):494–502, September 2024. doi:

10.1016/j.synbio.2024.04.005. URL https://doi.org/10.1016/j.synbio.2024.04.005.

Massaiu, I., Pasotti, L., Sonnenschein, N., Rama, E., Cavaletti, M., Magni, P., Calvio, C., and Herrgård, M. J. Integration of enzymatic data in bacillus subtilis genome-scale metabolic model improves phenotype predictions and enables in silico design of poly-γ-glutamic acid production strains. *Microb. Cell Fact.*, 18(1):3, January 2019.

Monk, J. M., Lloyd, C. J., Brunk, E., Mih, N., Sastry, A., King, Z., Takeuchi, R., Nomura, W., Zhang, Z., Mori, H., Feist, A. M., and Palsson, B. O. iml1515, a knowledgebase that computes escherichia coli traits. *Nature Biotechnology*, 35(10):904–908, October 2017. ISSN 1546-1696. doi: 10.1038/nbt.3956. URL http://dx.doi.org/10.1038/nbt.3956.

Moore, L., Caspi, R., Boyd, D., Berkmen, M., Mackie, A., Paley, S., and Karp, P. Revisiting the y-ome of escherichia coli. *Nucleic Acids Research*, 52(20):12201–12207, 10 2024. ISSN 0305-1048. doi: 10.1093/nar/gkae857. URL https://doi.org/10.1093/nar/gkae857.

Moretti, S., Martin, O., Van Du Tran, T., Bridge, A., Morgat, A., and Pagni, M. Metanetx/mnxref – reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks. *Nucleic Acids Research*, 44(D1):D523–D526, 11 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv1117. URL https://doi.org/10.1093/nar/gkv1117.

Moretti, S., Tran, V., Mehl, F., Ibberson, M., and Pagni, M. Metanetx/mnxref: unified namespace for metabolites and biochemical reactions in the context of metabolic models. *Nucleic Acids Research*, 49(D1):D570–D574, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa992. URL https://doi.org/10.1093/nar/gkaa992.

Oberhardt, M. A., Palsson, B. Ø., and Papin, J. A. Applications of genome-scale metabolic reconstructions. *Mol. Syst. Biol.*, 5(1):320, November 2009.

Okahashi, N., Kajihata, S., Furusawa, C., and Shimizu, H. Reliable metabolic flux estimation in escherichia coli central carbon metabolism using intracellular free amino acids. *Metabolites*, 4(2):408–420, 2014.

Orth, J. D., Thiele, I., and Palsson, B. Ø. What is flux balance analysis? *Nat. Biotechnol.*, 28(3):245–248, March 2010.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Sánchez, B. J., Zhang, C., Nilsson, A., Lahtvee, P.-J., Kerkhoven, E. J., and Nielsen, J. Improving the phenotype predictions of a yeast genome-scale metabolic model by incorporating enzymatic constraints. *Mol. Syst. Biol.*, 13(8):935, August 2017.

Schmidt, A., Kochanowski, K., Vedelaar, S., Ahrné, E., Volkmer, B., Callipo, L., Knoops, K., Bauer, M., Aebersold, R., and Heinemann, M. The quantitative and condition-dependent escherichia coli proteome. *Nat. Biotechnol.*, 34(1):104–110, January 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Weininger, D. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, February 1988.

Wittig, U., Rey, M., Weidemann, A., Kania, R., and Müller, W. SABIO-RK: an updated resource for manually curated biochemical reaction kinetics. *Nucleic Acids Res.*, 46(D1): D656–D660, January 2018.

A. ecGEM Annotations

In order to obtain the information we need to implement the enzyme constraint (i.e. MW_i and $k_{cat,i}$), we add additional reaction and metabolite annotations to our model file. We extracted amino acid sequence annotations from the BiGG (King et al., 2015) database to annotate reactions. We annotated metabolites with SMILES strings extracted from the MetaNetX database (Moretti et al., 2020; 2015; Ganter et al., 2013; Bernard et al., 2012). It must be noted that a lot of the pre-existing MetaNetX annotations are deprecated for both the ECMpy eciML1515 model and the BiGG database, such as prbatp_c (MNXM1351). For the majority of metabolites, this wasn't an issue, since the deprecated reference had syntactically correct SMILES annotations associated. However, for other reactions, such as octapb_c (MNXM147531), we had to manually fill these in using the canonical SMILES from PubChem (Kim et al., 2024). We did this for 139 out of 1877 metabolites. Finally, we used UniProt (Consortium, 2022) to obtain molecular weights and subunit information associated with the genes in *E. coli*. Together, these three annotation sets were required to predict k_{cat}s: the first two for the machine learning model input, and the last for the aggregation stage for multimer/heteromer k_{cat} prediction.

Given the necessity for special treatment of k_{cat} values for transport reactions (see Section 3.3), we also annotated all transport reactions in the model according to their classification in (Heckmann et al., 2020). We discovered that some transport reactions were not in this list, so we decided to add an annotation of "Transport, uncategorised" to all reactions that have either "transport" or "exchange" in the name and that do not have either a gene reaction rule of "s0001" (spontaneous reactions) or "" (empty gene[5]). The latter two types were not annotated as transport reactions in our model.

B. Transport Reactions

Because our k_{cat} predictor cannot predict transport reactions well (due to the fact that they are vastly underrepresented in the training data), and there is no principled method for handling transport turnover numbers, we followed the common convention to use $k_{cat} = 234,000\, h^{-1}$ (Corrao et al., 2024; Heckmann et al., 2018). This choice is somewhat arbitrary, and we recommend future research to find a more rigorous treatment. The precise heuristics we used are the following:

- For all transport reactions that are annotated as a transport reaction but are not porins, e.g. SUCptspp, we set $k_{cat} = 234,000\ \mathrm{h}^{-1}$.
- For transport reactions with one pore, e.g. ACtex gene b1377 (ompN), we also set $k_{cat} = 234,000\ \mathrm{h}^{-1}$.
- For transport reactions with N pores e.g. ACtex gene b0241 (phoE), we set $k_{cat} = N \times 234,000\ \mathrm{h}^{-1}$ ($N = 3$ for phoE).
- For spontaneous reactions with gene s0001 or reactions without genes, we do not set k_{cat}, which means these reactions are unconstrained.

[5] Some reactions are missing an associated gene in the underlying iML1515 model. Some of these reactions are COBRA boundary reactions of the exchange (EX_) type and are essentially spontaneous reactions, but not all of them. To be on the safe side, all such reactions were not annotated, and did not receive a k_{cat} value in our model (see Appendix B.

C. ecGEM metrics

Table 3 shows the reaction statistics for our model. k_{cat} coverage in our ecGEM is extensive; we have k_{cat} values for 5323 out of 5338 reactions for which the required input information (genes) is available. For 7 of the 15 missing reactions, the model output a k_{cat} of 0. The remaining 8 reactions have too long substrate SMILES strings, exceeding the ML model's 512 substrate token limit. These 15 reactions (only 0.2% of all reactions) were left unconstrained. The rest are constrained by either our ML k_{cat}s or the transport values in Appendix B.

In terms of subunits, the largest group consists of homomultimers, which make up 47% of all reactions. 7% of reactions are heteromeric, for which the choice of multimer aggregation described in Section 3.3.1 affects the k_{cat} value, it is important to note that this is a non-negligible number.

Table 3. Reaction statistics for eciML1515

Metric	**Value**	**% of Total**
Total reactions	6190	-
Total reactions with genes	5338	86%
Coverage of k_{cat} incl transport	5323	86%
Missing k_{cat}	15	0.2 %
ML k_{cat}	2360	38%
Exchange reactions (EX_)	662	11%
Spontaneous reactions (s0001)	64	1%
Transport reactions	2963	48%
Porin reactions	2186	35%
Monomers	2020	33%
Homomultimers	2883	47%
Heteromultimers	435	7%

D. Flux visualisations

Figures 6 and 7 show flux visualisations for the min model. Edges correspond to $\ln v_i$. Where $\ln v_i < 0.5$, and inclusion significantly reduces visual clarity, nodes and edges are not shown. Grey nodes and edges in Figure 6 and 7 correspond to flux solutions which fall below the threshold. Key:

Green	Citric Acid Cycle
Purple	Calibration candidates (Section 4.5)
Pink	Glucose
Grey Nodes	Metabolites of reactions falling below flux threshold
Grey Edges	Fluxes clamped at minimum value for visual clarity (0.5)
Dark Blue	Fluxes clamped at maximum value for visual clarity (10.0)
Orange	Other Metabolites
Light Blue	Other Reactions

Figure 6 shows the fluxes in the min model before any k_{cat} calibration was applied. Figure 7 shows the fluxes after calibrating the 8 most sensitive reactions via our flux control analysis. When visually comparing these two, it is clear that k_{cat} calibration of a relatively small number of reactions can have an effect on the whole metabolism of the organism.

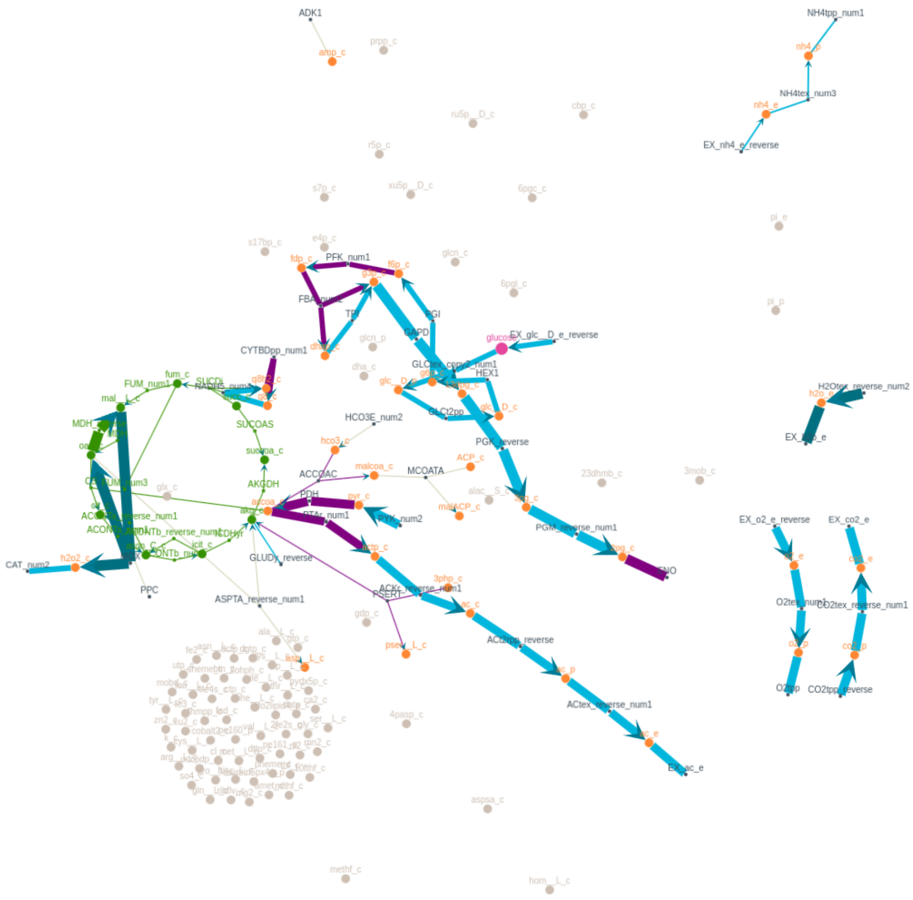

Figure 6. **Metabolic Flux Visualisation**: Enzyme-constrained solution, found with FBA for the min model.

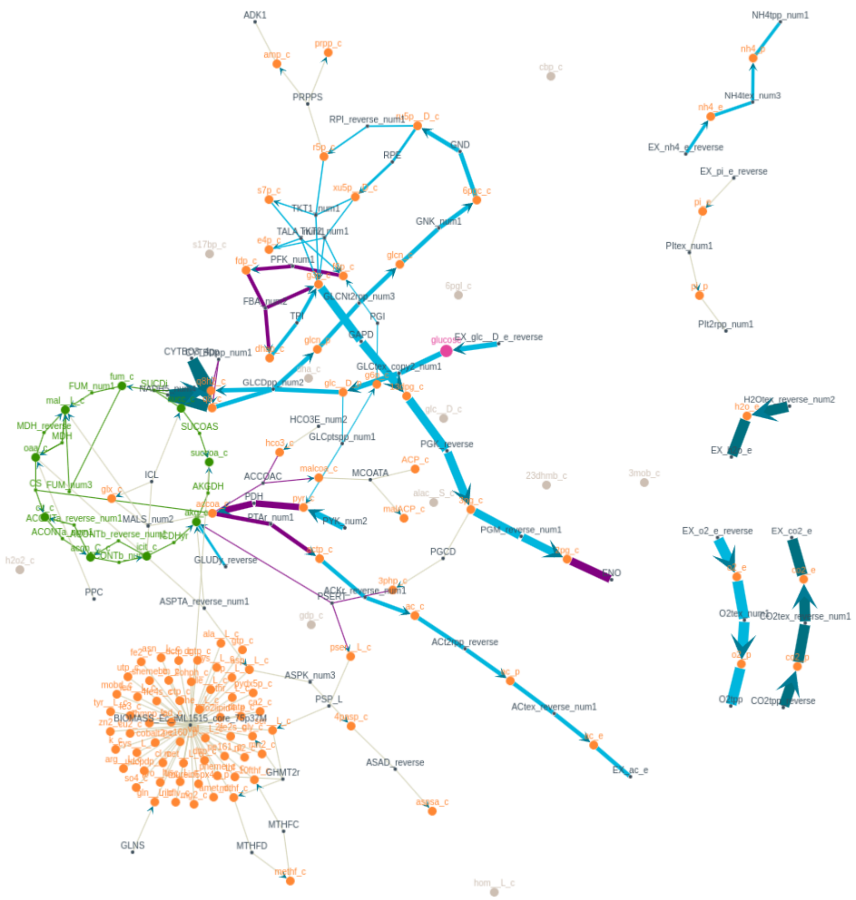

Figure 7. **Metabolic Flux Visualisation**: Enzyme-constrained solution, found with FBA for the min_8_clbr model.

E. Proof of Equivalence of Flux Control Coefficient and Enzyme Cost

E.1. Introduction

In an enzyme-constrained metabolic model, we have:

- A total enzyme budget E_{tot} that is distributed among different enzymes in the cell.
- Each reaction v_j limited by the amount of its enzyme E_j times the enzyme turnover number/catalytic rate k_{cat}, j. That is,
$$v_j \leq k_{cat,j} E_j,$$
often under simplifing assumptions such as saturating conditions.
- An objective function (commonly biomass production) that we seek to maximise subject to stoichiometric and enzyme capacity constraints.

We want to *prove* that under optimal allocation of enzymes (i.e., at the solution maximising biomass flux v_J),

$$\boxed{C^{v_J}_{k_{cat,i}} = \frac{E_j}{E_{tot}},}$$

where $\frac{E_j}{E_{tot}}$ is the *fraction of total enzyme* allocated to enzyme j (the enzyme cost of that enzyme).

E.2. Proof

E.2.1. Defining the Flux Control Coefficient

The flux control coefficient of reaction j with respect to $k_{cat,j}$ is :

$$C^{v_J}_{k_{cat,i}} = \frac{\partial \ln v_J}{\partial \ln k_{cat,j}} = \frac{\partial v_J}{\partial k_{cat,j}} \frac{k_{cat}, j}{v_J}.$$

Here v_J is the biomass flux.

E.2.2. Total Enzyme Budget Constraint

Suppose we have a constraint on total enzyme:

$$\sum_i E_j \leq E_{tot},$$

and we want to maximise v_J. Because $v_j = k_{cat,j} E_j$, we can express

$$v_J = \max\Big\{\text{biomass flux} \mid \text{stoichiometry}, v_j \leq k_{cat,j} E_j, \sum_j E_j \leq E_{tot}\Big\}.$$

At the *optimal* solution, each active reaction satisfies $v_j = k_{cat,j} E_j$.

E.2.3. Lagrange Multipliers

We introduce a Lagrange multiplier λ for the total enzyme constraint. At the optimum:

$$\frac{\partial v_J}{\partial E_j} = \lambda \text{ for each active enzyme } j.$$

This condition states that each actively used enzyme must have the same marginal benefit; otherwise, shifting enzyme to higher-return enzymes could increase v_J.

E.2.4. Relationship Between $\partial v_J/\partial k_{cat,j}$ and $\partial v_J/\partial E_j$

Observe

$$\frac{\partial v_J}{\partial k_{cat,j}} = \frac{\partial v_J}{\partial v_j}\frac{\partial v_j}{\partial k_{cat,j}} = \frac{\partial v_J}{\partial v_j}E_j,$$

Because $v_j = k_{cat,j}E_j \implies \frac{\partial v_j}{\partial k_{cat,j}} = E_j$.

Also,

$$\frac{\partial v_J}{\partial E_j} = \frac{\partial v_J}{\partial v_j}\frac{\partial v_j}{\partial E_j} = \frac{\partial v_J}{\partial v_j}k_{cat,j}.$$

Given $\frac{\partial v_J}{\partial E_j} = \lambda$, we get

$$\frac{\partial v_J}{\partial v_j} = \frac{\lambda}{k_{cat,j}}.$$

Hence,

$$\frac{\partial v_J}{\partial k_{cat,j}} = (\frac{\lambda}{k_{cat,j}})E_j = \lambda\frac{E_j}{k_{cat,j}}.$$

E.2.5. Express $C^{v_J}_{k_{cat,j}}$ in Terms of λ

Recall

$$C^{v_J}_{k_{cat,j}} = \frac{k_{cat,j}}{v_J}\frac{\partial v_J}{\partial k_{cat,j}} = \frac{k_{cat,j}}{v_J}\cdot\lambda\frac{E_j}{k_{cat,j}} = \lambda\frac{E_j}{v_J}.$$

So

$$C^{v_J}_{k_{cat,j}} = \lambda\frac{E_j}{v_J}. \tag{7}$$

E.2.6. Summation Theorem and $\sum_j E_j = E_{tot}$

Under typical control analysis assumptions (each $k_{cat,j}$ is an independent parameter), the sum of the control coefficients $C^{v_J}_{k_{cat,j}}$ for all active j is 1:

$$\sum_j C^{v_J}_{k_{cat,j}} = 1$$

Combined with equation 7:

$$\sum_j C^{v_J}_{k_{cat,j}} = \sum_j(\lambda\frac{E_j}{v_J}) = \frac{\lambda}{v_J}\sum_j E_j = \frac{\lambda}{v_J}e_{tot}.$$

Because this sum equals 1, we obtain

$$1 = \frac{\lambda}{v_J}E_{tot} \implies \lambda = \frac{v_J}{E_{tot}}.$$

E.2.7. Final Step: Fraction of Total Enzyme = Flux Control Coefficient

Substituting $\lambda = \frac{v_J}{E_{tot}}$ back into equation 7,

$$C^{v_J}_{k_{cat,j}} = \lambda \frac{E_j}{v_J},$$

we get

$$C^{v_J}_{k_{cat,j}} = \frac{v_J}{E_{tot}} \frac{E_j}{v_J} = \frac{E_j}{E_{tot}}.$$

Hence,

$$\boxed{C^{v_J}_{k_{cat,i}} = \frac{E_j}{E_{tot}}.}$$

This completes the proof that, under an optimal solution (maximising flux with a fixed enzyme budget), the fraction of total enzyme allocated to each enzyme (the enzyme cost) is numerically equal to its flux control coefficient with respect to $k_{cat,j}$. Equivalently,

$$\text{enzyme cost} \propto \text{control coefficient.}$$

F. Enzyme abundances benchmarking

One of the benchmarks in our benchmarking process traces accuracy in enzyme abundances. Corrao et al. used polypeptide abundances from Schmidt et al. (2016) to compute enzyme abundances for 290 enzymes in *E. coli* (Corrao et al., 2024). We use their data for growth on glucose.

As the enzymes and reactions in their model have different names to ours, we matched our reactions to theirs by 1) finding all the genes for each enzyme in their model by using their knowledge graph and 2) finding all reactions in our model that use this specific set of genes. As some enzymes may catalyse multiple reactions, we first find the ‘abundance’ of each enzyme-reaction pair:

$$A_{r,i} = \frac{v_{r,i} MW_i}{k_{cat,r,i}\sigma_{r,i}}, \tag{8}$$

where the subscripts r and i denote reaction r catalysed by enzyme i. MW_i is the molecular weight of the enzyme. $A_{r,i}$, $v_{r,i}$, $k_{cat,r,i}$, and $\sigma_{r,i}$ are the abundance, flux, turnover number, and average saturation coefficient respectively. As mentioned in Section 3.1, throughout this work we assume a constant value of $\sigma_{r,i} = 1$. If the flux $v_{r,i}$ was lower than the expected floating point accuracy of the solver (generously set to 10^{-14}), the abundance for that reaction was set to zero, as it would be indiscriminately close to zero. To compute the total enzyme abundance for enzyme i, we then simply add the abundances for all reactions that this enzyme catalyses:

$$A_i = \sum_r A_{r,i} = MW_i \sum_r \frac{v_{r,i}}{k_{cat,r,i}} \tag{9}$$

We then compare these abundances to the measured values from (Corrao et al., 2024). To do so, we follow (Corrao et al., 2024) in scaling the predicted abundances by a scale factor such that their sum is equal to the sum of all measured abundances $\sum_i A_i = \sum_i A_{i,measured}$. We then compute the root mean squared logarithmic error (RMSLE; we use a logarithmic scale as the abundance values cover a range of 10^{-12} to 10^{-1}).

G. Reference models

Here we present more detail about the reference models used in the benchmarking process in Section 4.2

- **ECMpy_ML** is the ECMpy v1 model before doing any k_{cat} correction rounds, i.e. iML1515_irr_enz_constraint.json, produced by Mao et al. (2022) as per their GitHub on 25 Dec 2021.[6] This model uses machine learning k_{cat} values from Heckmann et al. (2018).

- **ECMpy_expmnt** is the ECMpy v2 model[7] (Mao et al., 2024) before applying any k_{cat} corrections. Unlike the aforementioned ECMpy_ML, which uses machine learning k_{cat}s, the values in this model are experimental values. They are obtained from the BRENDA (Chang et al., 2021) and SABIO-RK (Wittig et al., 2018) databases using AutoPACMEN (Bekiaris & Klamt, 2020), and an average value of k_{cat} was chosen for reactions that were not in the databases.

- **ECMpy_DLKcat** is a model created via the DLKcat pipeline in ECMpy v2 [8], which computes machine learning k_{cat}s using the DLKcat method (Li et al., 2022). Unfortunately, the code in the ECMpy v2 GitHub produced NaN values for 15% of k_{cat}s. This is almost entirely due to the fact that the code in the notebook does not retrieve the correct SMILES information for these reactions. We changed all NaN values to empty strings, so that these reactions were unconstrained. It must therefore be noted that since DLKcat could not be applied effectively to all reactions, our benchmarking results reflect the performance of the combined ECMpy+DLKcat pipeline rather than DLKcat alone.

- **ECMpy_ML_calibrated** is a model from ECMpy v1. This model, titled iML1515_irr_enz_constraint_adj_round1.json, is produced by subjecting the uncalibrated model ECMpy_ML to the first round of k_{cat} calibration. There is also a second correction round, which uses ^{13}C fluxes, but we do not include this model as it uses additional experimental information and is therefore not comparable to our calibrated models. The first calibration round adjusted k_{cat}s selected by enzyme cost ($\frac{v_i MW_i}{\sigma_i k_{cat,i}}$). k_{cat}s were updated for reactions that used more than 1% of total enzyme. The updated k_{cat}s are pulled from the BRENDA and SABIO-RK databases. See Mao et al. (2022) for more details.

- **ECMpy_expmnt_calibrated** is the calibrated version of the ECMpy v2 model ECMpy_expmt, which has been passed through the v2 k_{cat} calibration process. This process involved 50 rounds of iteratively adjusting k_{cat} for those enzyme-reaction pairs with the highest enzyme cost, and updating its value to the highest datapoint found in BRENDA and SABIO-RK. See Mao et al. (2024) for more details.

[6] `http://github.com/tibbdc/ECMpy/tree/433463a9b22994765351eae1ea1b74d133f7a483`

[7] To be precise, we create this model by running the notebook "02.get_ecModel_using_ECMpy.ipynb" on the saved state of the ECMpy GitHub repository as it was on 15 Feb 2025 (`https://github.com/tibbdc/ECMpy`).

[8] As this model is not saved to their Github we had to generate it ourselves. We did this via the notebook 01.get_reactiion_kcat_using_DLKcat [sic]. We also had to change line 4 in cell 6 to `subbnumdf = pd.read_csv(gene_subnum_path, index_col = 0)`; without this change the pipeline does not compute k_{cat}/MW correctly.

H. k_{cat} sensitivity through flux control coefficients

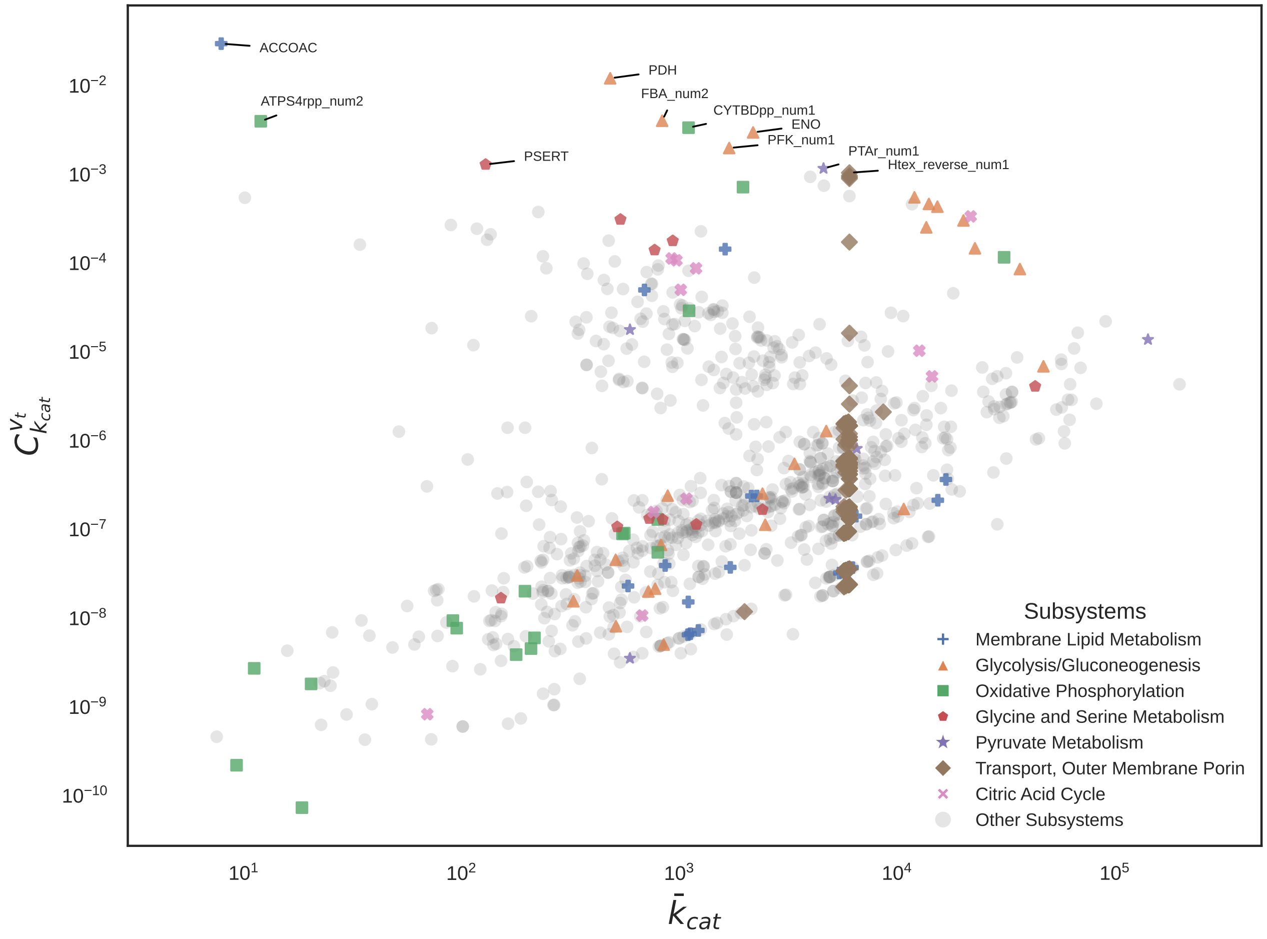

Figure 8. k_{cat} sensitivity plot, categorised and coloured by subsystem for the min model (subsystem annotations from the BiGG database (King et al., 2015)). Flux control coefficient $C^{v_t}_{k_{cat,i}} = \frac{\delta v_T}{\delta \bar{k}_{cat,i}} \cdot \frac{\bar{k}_{kcat,i}}{v_T}$, rescaled $\bar{k}_{cat} = \frac{k_{cat}}{MW}$, perturbation $\delta \bar{k}_{cat,i} = 10^{-4}$, metabolic target T: $BIOMASS\_Ec\_iML1515\_core\_75p37M$. Coloured subsystems are those present in the top 10 most sensitive reactions under flux control coefficient analysis except the Citric Acid Cycle which is highlighted for reference. The band of Transport reactions at $\bar{k}_{cat} = 6012\ \mathrm{h}^{-1}$ is due to the heuristic described in Section 3.3. Not shown are reactions where $C^{v_t}_{k_{cat,i}} \leq 0$ (due to log-scale plotting).

I. Top 10 most sensitive reactions for min model by flux control coefficient

Table 4. Top 10 most sensitive reactions for the min model as measured by flux control coefficient, $C^{vt}_{k_{cat,i}}$. Red highlighting corresponds to ECMpy v1 round 1 enzyme abundance corrections, Blue highlighting corresponds to ECMpy v1 round 2 ^{13}C corrections. EC Numbers for each reaction were queried across BRENDA and SABIO-RK, and we retrieved the largest k_{cat} across each search (including reactions from different species to *E. coli*, and across different environmental conditions).

Reaction (BiGG ID)	$\bar{k}_{cat}$ in min model $[h^{-1}]$	$\bar{k}_{cat}$ after calibrating with BRENDA/ SABIO-RK $[h^{-1}]$	$C^{vt}_{k_{cat,i}}$	Description
ACCOAC	7.8	524.8	0.0294	Acetyl-CoA carboxylase
PDH	479.0	784.1	0.0120	Pyruvate dehydrogenase
FBA_num2	830.0	9437.1	0.0040	Fructose-bisphosphate aldolase
ATPS4rpp_num2	11.9	3661.1	0.0039	ATP synthase (four protons for one ATP) (periplasm)
CYTBDpp_num1	1097.4	1097.4[1]	0.0033	Cytochrome oxidase bd (ubiquinol-8: 2 protons) (periplasm)
ENO	2171.1	9068.0	0.0029	Enolase
PFK_num1	1685.6	113914.2	0.0020	Phosphofructokinase
PSERT	128.3	112.7	0.0013	Phospho-L-serine transport via diffusion (extracellular to periplasm) reverse
PTAr_num1	4571.4	46680.1	0.0012	Phosphotransacetylase
Htex_reverse_num1	6012.0	6012.0[1]	0.0010	Proton transport via diffusion (extracellular to periplasm) reverse

[1] EC numbers for these reactions aren't recorded in BiGG, so we couldn't query BRENDA or SABIO-RK to retrieve in-vivo k_{cat}. These k_{kcat}s remain uncalibrated.

J. Top 100 most sensitive reactions according to flux control coefficient

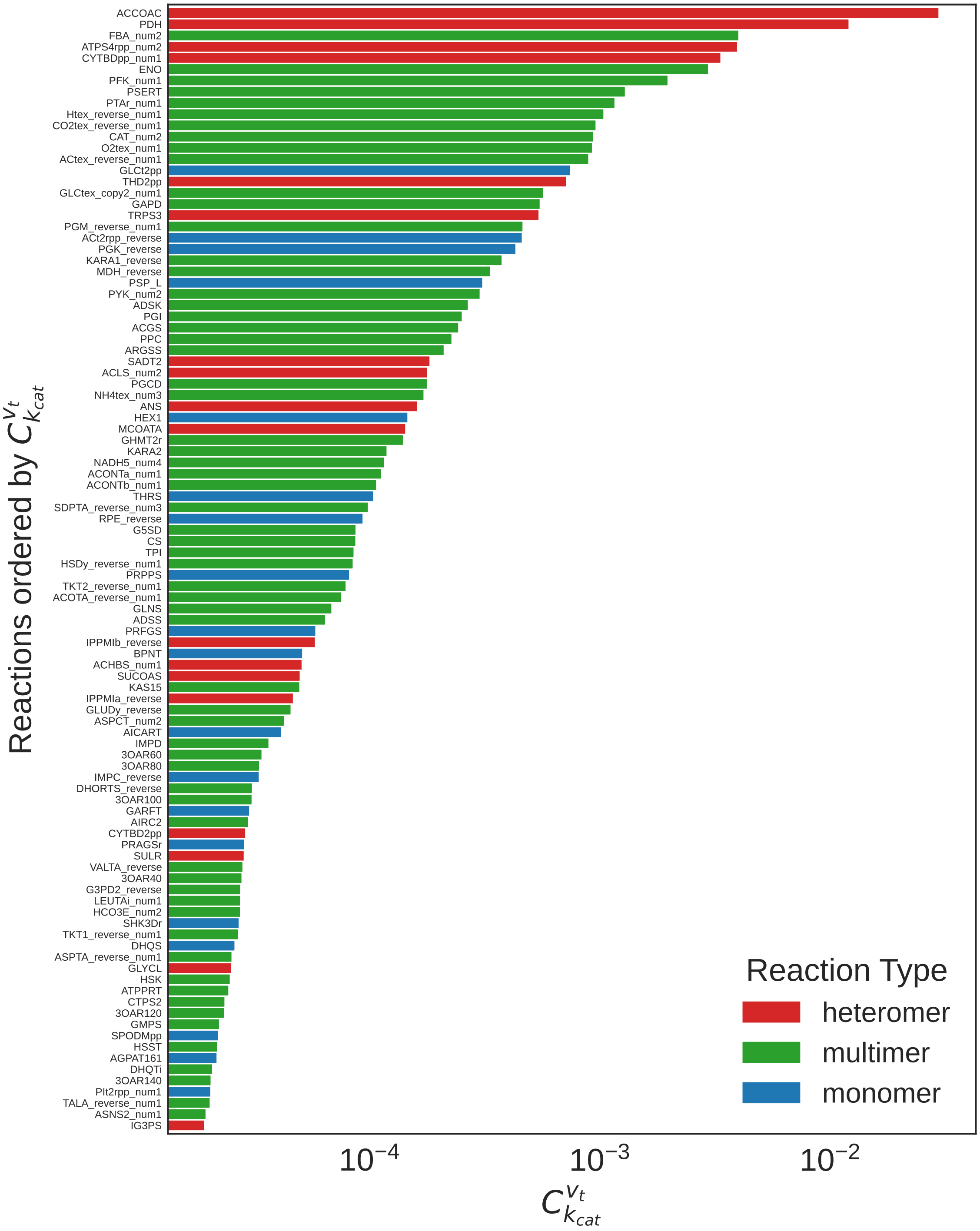

Figure 9. Top 100 reactions in the min model ranked by flux control coefficient $C^{V_t}_{k_{cat,i}} = \frac{\delta V_T}{\delta k_{cat,i}} \cdot \frac{\bar{k}_{kcat,i}}{V_T}$, coloured by enzymatic reaction type (heteromer, homomultimer, monomer). Non-enzymatic reactions are not shown.