Crowded enzyme kinetics using simulation and machine learning

William Cappelletti, *MA*, Charles Dufour, *MA*, Marie Sadler, *CGC*

ABSTRACT

We derive a model to predict the four kinetic rate constants of the reversible Michaelis-Menten enzymatic reaction mechanism in a crowded environment. The model is trained on simulated data obtained from a crowded particle model which has the following features: reactant concentrations, volume exclusion fraction and mass distribution of inert particles. Reproducing the previous work done on the available dataset, we first perform weighted linear regressions for each combination of volume exclusion and mass distribution parameters which is our baseline model. We then implement several machine learning algorithms and develop a generalized model integrating all the parameters. Our final model makes use of the computationally efficient gradient boosting algorithm `XGBoost` and is composed by four boosters, each one trained to predict one of the kinetic rate constants. This model significantly increases the prediction power compared to the previous aforementioned baseline model.

I. INTRODUCTION

Kinetic data of biological organisms is often obtained from *in vitro* experiments in dilute conditions which are not representative of the real cell environment whose volume can be occupied up to 40% by a mixture of proteins, lipids, polysaccharides, RNA and DNA. These crowding agents influence the kinetics of biochemical reactions, and thus there is a necessity to develop computational models capturing the effect of macromolecular interactions. A particle-based simulation studying an enzyme-catalyzed reaction system, namely the activity of phosphoglycerate mutase in Escherichia coli, allowed to predict the kinetical parameters ($k_{1,f}$, $k_{1,b}$, $k_{2,f}$ and $k_{2,b}$) of the reversible Michaelis-Menten mechanism in a crowded environment (Weilandt *et al.* [1]), which is formulated as follows:

$$\mathrm{S} + \mathrm{E} \underset{k_{1,b}}{\overset{k_{1,f}}{\rightleftharpoons}} \mathrm{ES} \underset{k_{2,b}}{\overset{k_{2,f}}{\rightleftharpoons}} \mathrm{E} + \mathrm{P}, \tag{1}$$

where the substrate S binds to the enzyme E to form a complex ES, which then produces the product P liberating again the enzyme E.

This particle-based simulation, entitled GEneralized Elementary Kinetics (GEEK), predicts the enzymatic rate constants for varying volume exclusion conditions (i.e. the space occupied by inert molecules representative for a crowded environment), mass and mass distribution of the inert molecules, and reactive species concentrations (see Fig. 1). Starting from this simulation data ($151'956$ observations), our study aims to develop a model predicting the rate constants as a machine-learning black box.

Our fist step is the reproduction of the regression conducted by Weilandt [1], which is based on the mass-action kinetics mode, and the analysis of the coefficient estimates as described in section II. Then, we derive a regression model which also includes additional parameters such as the volume exclusion fraction and inert particle properties. Afterwards, we use machine learning techniques to implement an highly accurate new model, deviating from the mass-action kinetics law; we start from basic regression techniques, such as ridge regression, to move on to more advanced models, like Support Vector Regression and XGBoost, in section III.

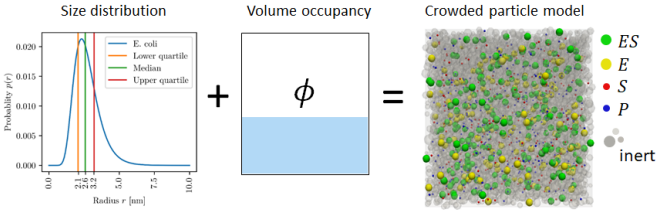

Fig. 1: Scheme of the crowding effect in the enzyme environment (the size distribution is equivalent to the mass distribution of the inert particles)

II. REPRODUCTION

A. Michaelis-Menten mechanism and mass-action kinetics

The rates of the elementary reactions in crowded conditions, simulated by the introduction of inert molecules, can be derived from a generalized mass-action rate law [1] and the effective rate constants $k_{1,f,eff}, k_{1,b,eff}, k_{2,f,eff}$ and $k_{2,b,eff}$ can be expressed as follows:

$$\ln\left(\frac{k_j, eff}{k_{j,0}}\right) = \sum_{n=1}^{N} \alpha_{i,j} \ln\left(\frac{[X_i]}{[X_i]_0}\right) + \beta_j. \tag{2}$$

In the above equation (2) $j \in [(1,f),(1,b),(2,f),(2,b)]$, $k_{j,eff}$ is the effective rate constant in crowded condition and $k_{j,0}$ the rate constant in dilute condition; to avoid the otherwise cumbersome notation, we will from now on refer to the log of the ratio as k_j. $[X_i]$ is the concentration of the reactive species (i $\in$ [S, E, ES, P]), and $[X_i]_0$ is the reference concentration (for simpler notation, we denote $[X]/[X]_0$ by $[X]_{\text{norm}}$).

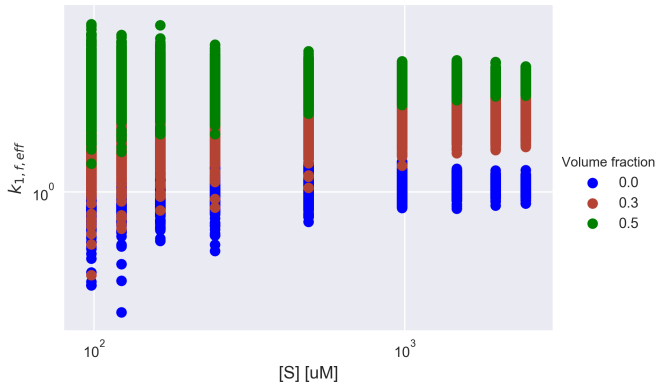

Fig. 2: Rate constant $k_{1,f,eff}$ as a function of the concentration $[S]$ in μM for different volume fractions ϕ

B. Feature description

Following the original article [1], we split the data with respect to the different combinations of values of volume fraction ϕ ($\phi \in [0, 0.1, 0.2, 0.3, 0.4, 0.5]$), median of the mass of the inert particles μ, and of the sigma parameter of the mass distribution σ ($\sigma \in [0, 0.825]$, where 0 means that all the crowding agents have the same mass). Then for each "batch" of our data (without splitting in a train and test set), we fit a weighted linear model as described below depending on the concentrations $[X_i]_{\text{norm}}$ (Fig. 2 shows the rate constant $k_{1,f,eff}$ as a function of the concentration $[S]$ for different values of ϕ; see Fig. 9 in the Appendix for the dependencies of the 4 rate constants on each of the 4 concentrations).

C. Model Reproduction

From the simulation data used in the paper, we reproduce the presented regression model as follows. Using `StatsModels`, we perform a weighted linear regression for

$$k_j = \beta_j + \alpha_{\text{E},j} \ln\left([\text{E}]_{\text{norm}}\right) + \alpha_{\text{ES},j} \ln\left([\text{ES}]_{\text{norm}}\right) \\ \alpha_{\text{P},j} \ln\left([\text{P}]_{\text{norm}}\right) + \alpha_{\text{S},j} \ln\left([\text{S}]_{\text{norm}}\right) \quad (3)$$

The weights of the observations are defined as inversely proportional to the conditional standard deviation $\sqrt{Var(r|X)}$ where r represents the residuals of an ordinary least squares regression. This was done in order to avoid fitting data with high heteroscedasticity [1].

The results we obtain correspond to those presented in the original paper [1] and we report in Table I some indicators of the goodness of fit, namely the *explained variance* (R^2) and the *mean squared error* of the fitted responses. The other results can be found in the supplementary material.

volume fraction	0.1	0.2	0.3	0.4	0.5
R_c^2	0.610	0.457	0.420	0.385	0.289
MSE on train set	0.0128	0.0151	0.0208	0.0283	0.0352

TABLE I: Measures of goodness of fit of the weighted linear regression for $k_{1,b}$ with $\sigma = 0.825$, and $\mu = 31.9$

III. New models

We discuss different models to improve the predictions we obtained in section II. Again, our responses will be the four $k_j = \ln\left(\frac{k_{j,eff}}{k_{j,0}}\right)$, with $j \in \{(1,b),(1,f),(2,b),(2,f)\}$, and our initial training features are the log-scaled concentrations of the reactive species (the previously introduced $[\text{E}], [\text{S}], [\text{ES}], [\text{P}]$, the occupied volume fraction ϕ and the median of the mass of the inert particles μ, now used as continuous variables, along with the two sigma parameters of the mass distribution of the inert particles, considered as categorical and encoded as $0-1$ in two columns of the initial design matrix (dummy variables $\sigma_0 and \sigma_1$). We will consider this as our basic data matrix and treat feature engineering separately for each model.

Each model is fitted on 60% of the data and then tested on the remaining 40% to have a better estimate of the expected error. Since the original data were generated numerically, multiple observations were gathered during the same simulation and this information is part of the data set.

Therefore, we can split the observations in test and train either in a totally random way (the observation are sampled uniformly among the whole data set), either accounting for the original simulation (sampling uniformly on each simulation to get test and train data well distributed across them). Since no significant difference is observed, we consider in this analysis the uniform split, obtained using `sklearn.model_selection.train_test_split()` (Pedregosa et al., [3]).

A. Ridge regression

Since theory suggests the linear model hypothesis [1], our first guess is to fit a ridge regression for each k_j, using all our features. Looking at scatter plots of the responses against each feature gives little to no insight (see Fig. 9 in the Appendix), therefore, we use the previously cited goodness of fit scores (R^2 and MSE) to choose the best transformations of the features. The λ hyper parameter is tuned at each run using the leave-one-out cross validation, which is highly efficient and, considering the size of our data set, gives a good estimate of the expected error. We tested polynomial expansion of the features up to degree 3, without raising the categorical variables but using them in cross products with the other variables to avoid collinearity (i.e. there is no σ_1^2, neither $\sigma_0 * \sigma_1$ in any design matrix, but there are for instance $[\text{E}]^2 * \sigma_0$ and $[\text{P}] * [\text{ES}] * \sigma_1$ in the degree 3 polynomial expansion), and an interaction model, in which we used all features and the product of each couple (again without $\sigma_0 * \sigma_1$).

For $k_{1,b}$ and $k_{2,f}$ the best model is given by the degree 2 polynomial expansion, while for $k_{1,f}$ and $k_{2,b}$ it is obtained with the basic and the interaction-only design, probably because the L2 regularization is not enough to prevent overfitting. The test scores of these two models are reported in Table II and in Table III respectively. We discard the basic model and the degree-3-polynomial since they give terrible predictions, in terms of the scores.

	R^2 test score	MSE test score
k1_bwd_effective	0.988731	0.012533
k1_fwd_effective	0.867772	0.022036
k2_bwd_effective	0.881837	0.019809
k2_fwd_effective	0.987513	0.013887

TABLE II: Degree 2 polynomial model scores for Ridge

	R^2 test score	MSE test score
k1_bwd_effective	0.929903	0.077959
k1_fwd_effective	-1.120566	0.353405
k2_bwd_effective	0.905372	0.015864
k2_fwd_effective	0.929899	0.077963

TABLE III: Interaction only Ridge regression scores

B. Support vector regression

Our second approach to get an improvement in prediction exploits support vector regression. This method uses kernels to learn a function and, in the same way as support vector machines for classification, it enables implicit feature expansion. We perform a regression using the Gaussian kernel applied to the basic design matrix and then to the interaction design, as explained in III-A, trying to capture nonlinearity in the model. As we can see in Table IV, when applied to the basic design, this method performs worse than the previously cited ridge regressions for $k_{1,b}$ and $k_{2,f}$, while it slightly improves for $k_{1,f}$ and $k_{2,b}$. This suggests that these two last responses are indeed nonlinear with respect to the features, but they are still hard to explain, while $k_{1,b}$ and $k_{2,f}$ are not well fitted by this model.

On the other hand, when applied to the interaction model, SVR gives more accurate predictions for $k_{1,f}$ and $k_{2,b}$, reducing by half the MSE, while producing a slightly worse fit for the other two responses, as reported in Table V. Again this supports the hypothesis of $k_{1,f}$ and $k_{2,b}$ being easily overfitted, and the probable presence of interactions between the features in the underlying process generating $k_{1,b}$ and $k_{2,f}$. In spite of the promising results, we do not insist in trying many feature transformations, because of the high computational cost of SVR regression, and we move to a different, highly performing, model.

	R^2 test score	MSE test score
k1_bwd_effective	0.953209	0.051951
k1_fwd_effective	0.907186	0.015478
k2_bwd_effective	0.910208	0.015009
k2_fwd_effective	0.951694	0.053632

TABLE IV: SVR scores for basic design.

	R^2 test score	MSE test score
k1_bwd_effective	0.993300	0.007439
k1_fwd_effective	0.898296	0.016960
k2_bwd_effective	0.902213	0.016346
k2_fwd_effective	0.993300	0.007439

TABLE V: SVR scores for interaction model.

C. XGBoost

Due to the limitations we encountered in trying to identify a model *a priori*, we move now to a different strategy that does not require any hypothesis. We use gradient boosting, whose idea is to approximate the gradient of the true function in an iterative way using weak learners (for instance decision trees) and fit a function that is then a linear combination of these weak learners. In particular, we use the library `xgboost` presented by Cheng and Guestrin in [4], which implements a highly optimized tree boosting system.

We train one model for each response and, for each of them, we carefully tune three hyperparameters, namely the α L1 regularization parameter, the η learning rate, and the maximal depth for each decision tree. Those parameters are chosen based on a 7-fold cross validation, in order to minimize the CV-estimated MSE.

Furthermore, because of the tree component of the algorithm, we get an estimate of features importance, as we can see in Fig. 3. In this figure we present the estimates for the model fitted to $k_{1,b}$ using the basic design matrix and we can see that few parameters significantly contribute to the prediction. Features expansion only reduced all features importance without improving the predictions. Very similar results are yielded for the other responses. We choose, in our final model, to drop the least important features, namely $[\mathrm{E}]$ and σ, and we do not see any change in the model scores. We reported in Fig. 4 the features importance in the fitting of $k_{2,f}$ of the reduced design (the features importance for the other rate constants are found in the supplementary material).

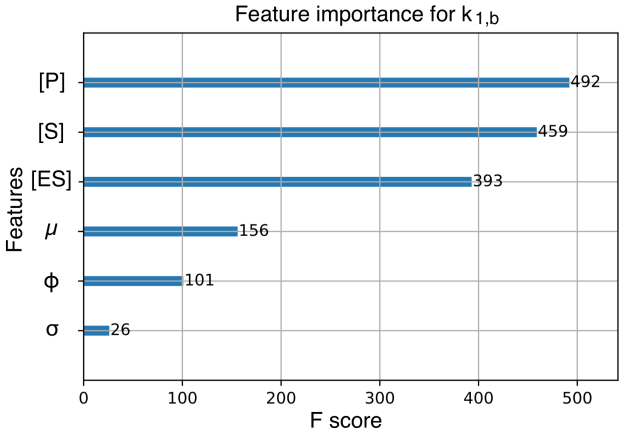

Fig. 3: Features importance of the basic design as estimated by XGBoost before feature elimination.
$[\mathrm{E}]$ is not visible as its F score is close to 0.

Therefore, our final model is, for each response, an XGBoost booster fitted on $[\mathrm{ES}], \mu, [\mathrm{P}], [\mathrm{S}]$ and volume fraction, whose scores are presented in Table VI.

IV. Discussion

In Tables VII and VIII , we see that the initial weighted linear regressions, fitted separately for each ϕ, μ and σ, fit poorly for most of the kinetic constants, as one could expect. Fitting a ridge regression with cross validated λ and then the

	R^2 test score	MSE test score
k1_bwd_effective	0.999838	0.000180
k1_fwd_effective	0.916180	0.013971
k2_bwd_effective	0.918504	0.013651
k2_fwd_effective	0.999839	0.000180

TABLE VI: XGBoost scores

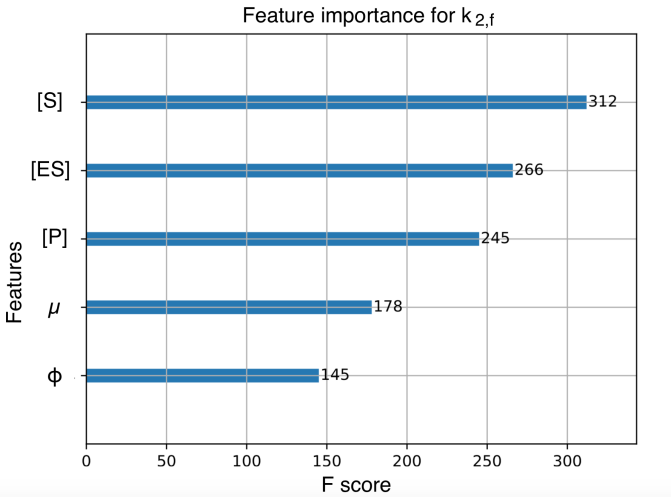

Fig. 4: Features importance as estimated through XGBoost in final model.

support vector regression to the respective engineered features improved slightly our predictions. Nevertheless, the greatest prediction improvement is obtained through XGBoosting, which also let us reduce the number of features without loosing any information. With this model, $k_{1,b}$ and $k_{2,f}$ are almost perfectly predicted, while the other two responses stay in the same order of magnitude as before. This suggests that $k_{1,b}$ and $k_{2,f}$ are not only nonlinear in the features, but they most surely have a great random component due to the underlying reaction process.

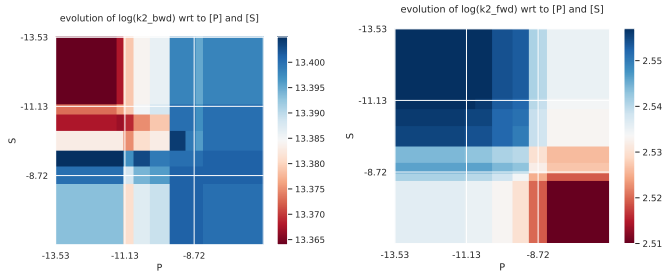

Fig. 5: Predicted values for k_{2b} (on the left) and k_{2f} (on the right) by XGBoost for fixed values of volume fraction 0.3, $[ES]$ is set at the median, while $[S]$ and $[P]$ move along the axis. The values are represented by a color code represented on the right of each plot.

The difficulty we encountered to get accurate predictions for $k_{1,f}$ and $k_{2,b}$ compared to $k_{1,b}$ and $k_{2,f}$ can also be explained from the chemical point of view of the reaction scheme. Kinetically speaking, these latter rate constants only depend on the concentration of [ES], while the other two rate constants depend on [S], [E] and [P], which introduces a higher degree of freedom and results in a larger variation of the rate constants (see Fig. 9). In addition, the feature importance estimated through XGBoost hints that [E] has a F score close to 0, this is explainable through the conservation law of the enzyme E: [E] + [ES] = $[E]_0$, where $[E]_0$ is a constant. Hence, since the concentrations [E] or [ES] are dependent on each other, one could be neglected.

The fact that the kinetic constants and the concentrations are highly non linear is supported by looking at Fig. 5 (similar plots are available in [5]), where we plot the predicted values for two moving features, while fixing the others. Again this supports the hypothesis of the probable presence of interactions between the features in the underlying process generating $k_{1,b}$ and $k_{2,f}$ as discussed in subsection III-B. This gives an explanation to the poor performance of the weighted linear model, which, furthermore, did not account for the exclusion volume and the inert particle properties.

In conclusion, our final model has outstanding performances with respect to the GEEK simulation data, providing simplification and improvement to the baseline linear regression models.

REFERENCES

[1] Weilandt, D. and Hatzimanikatis, V. *Particle-based simulation reveals macromolecular crowding effects on the Michaelis-Menten mechanism* doi: https://doi.org/10.1101/429316

[2] Hastie, T., and Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc., 2001.

[3] Pedregosa, F. and al. *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, , 2011

[4] Chen, Tianqi, and Guestrin. *Xgboost: A scalable tree boosting system.* Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.

[5] Cappelletti, W. and Dufour, C. and Sadler, M. *ML CS433 project2*, (2018), GitHub repository, https://github.com/dufourc1/ML_CS433_project2

APPENDIX

volume fraction	0.1	0.2	0.3	0.4	0.5
R_c^2	0.004	0.004	0.007	0.006	0.003
MSE on train set	0.1235	0.1191	0.1127	0.1290	0.0991

TABLE VII: Measures of goodness of fit of the weighted linear regression for $k_{1,f}$ with $\sigma = 0.825$, and $\mu = 31.9$

volume fraction	0.1	0.2	0.3	0.4	0.5
R_c^2	0.003	0.004	0.007	0.007	0.003
MSE on train set	0.1127	0.1161	0.1134	0.1276	0.0982

TABLE VIII: Measures of goodness of fit of the weighted linear regression for $k_{2,b}$ with $\sigma = 0.825$, and $\mu = 31.9$

volume fraction	0.1	0.2	0.3	0.4	0.5
R_c^2	0.610	0.457	0.420	0.3856	0.289
MSE on train set	0.0128	0.0151	0.0208	0.0284	0.0352

TABLE IX: Measures of goodness of fit of the weighted linear regression for $k_{2,f}$ with $\sigma = 0.825$, and $\mu = 31.9$

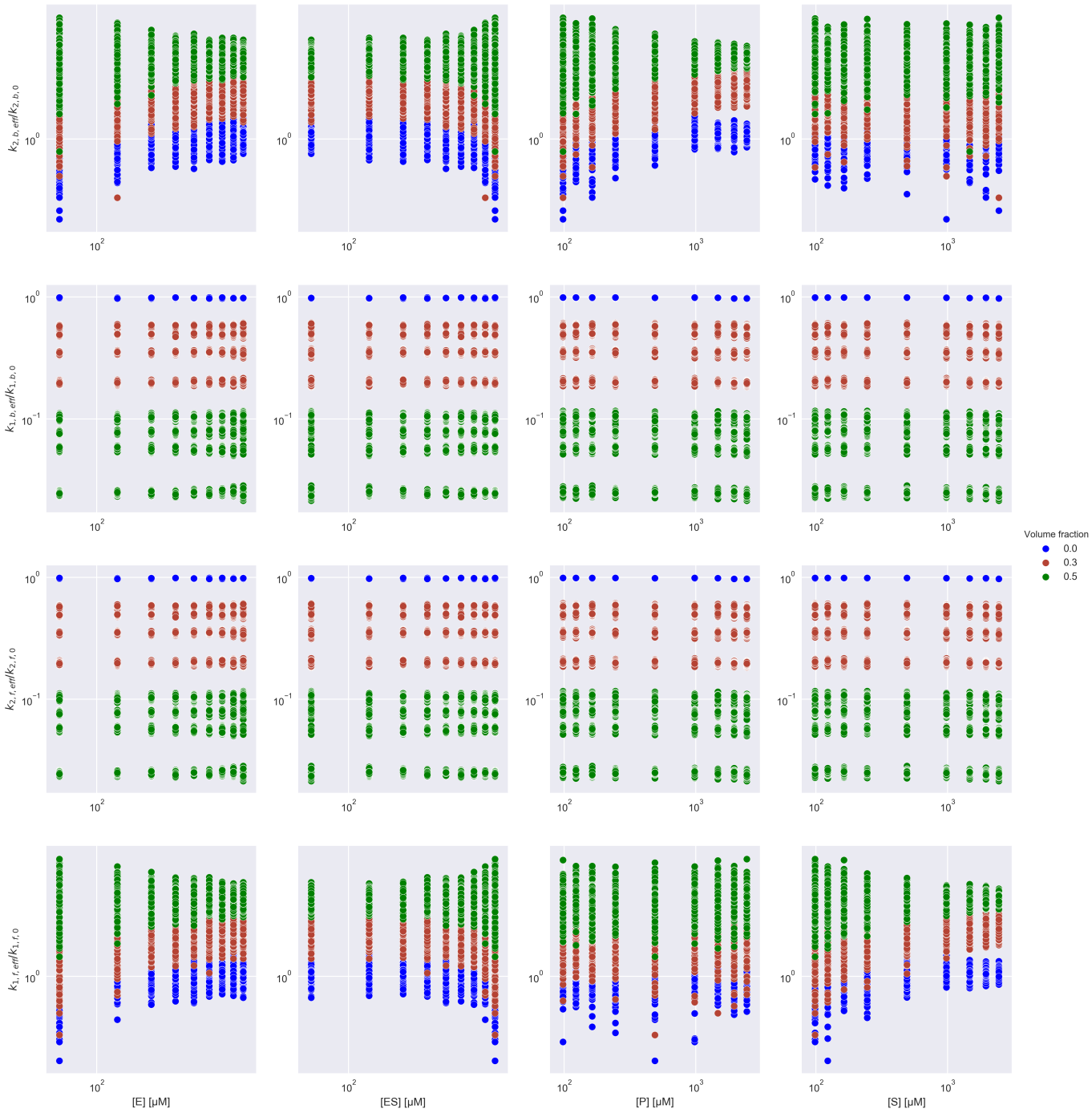

Fig. 6: Relative effective rate constants $k_{j,eff}$ as a function of the concentrations $[X]_i$ for different volume fractions ϕ

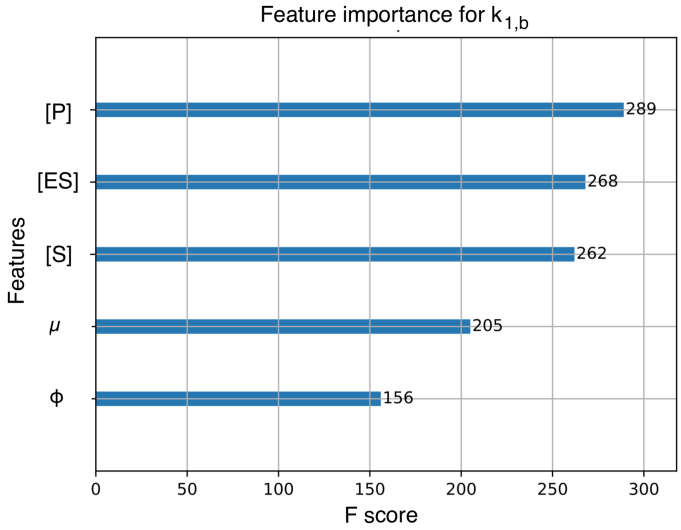

Fig. 7: Features importance as estimated through XGBoost in the final model for $k_{1,b}$.

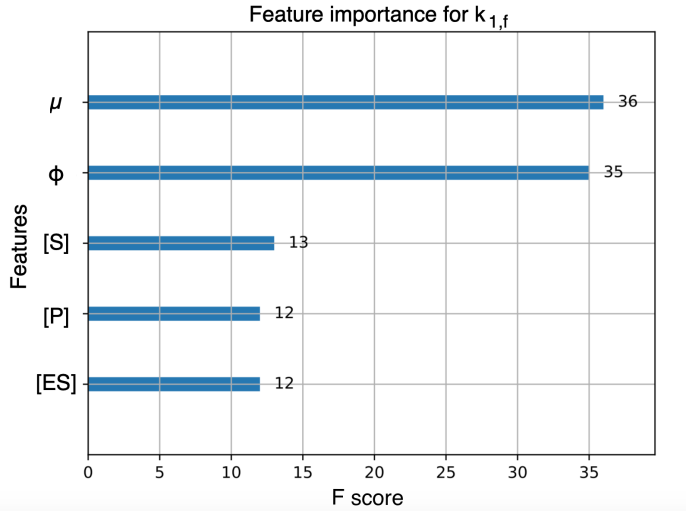

Fig. 8: Features importance as estimated through XGBoost in the final model for $k_{1,f}$.

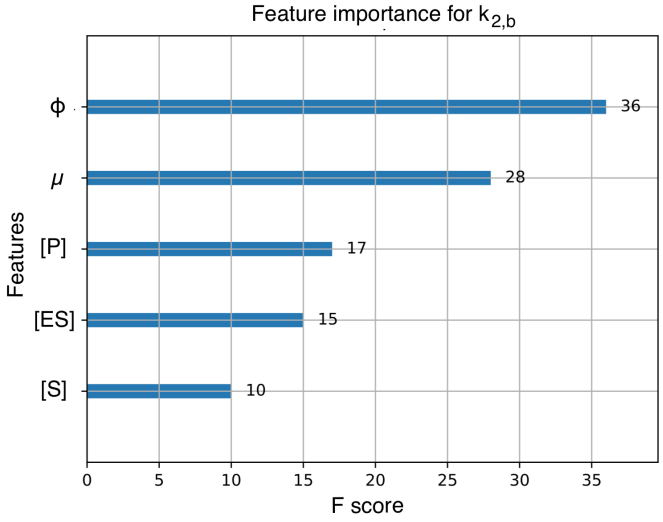

Fig. 9: Features importance as estimated through XGBoost in the final model for $k_{2,b}$.