Improving N-Glycosylation and Biopharmaceutical Production Predictions Using AutoML-Built Residual Hybrid Models

Pedro Seber **Richard D. Braatz**
{pseber,braatz}@mit.edu
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

N-glycosylation has many essential biological roles, and is important for biotherapeutics as it can affect drug efficacy, duration of effect, and toxicity. The prediction of N-glycosylation and other important biopharmaceutical production values have mostly been limited to mechanistic modeling. We present a residual hybrid modeling approach that integrates mechanistic modeling with machine learning to produce significantly more accurate predictions for N-glycosylation and bioproduction. For the largest dataset, the residual hybrid models have an average 736-fold reduction in testing prediction error. Furthermore, the residual hybrid models have lower prediction errors than the mechanistic models for all of the predicted variables in the datasets. We provide the automatic machine learning software used in this work, allowing reproduction and use of our software for other tasks.

1 INTRODUCTION

Glycosylation is a protein co-translational and post-translational modification that involves adding a glycan or glycans to proteins. N-linked glycosylation, a subtype of glycosylation, occurs when a glycan is added to the nitrogen of an asparagine or arginine. N-glycosylation contributes to many essential functional and structural roles (Imperiali and O'Connor, 1999; Patterson, 2005; Schjoldager et al., 2020). Highlighting the importance of N-glycosylation, improper glycosylation or deglycosylation is associated with multiple diseases, including cancers (Stowell et al., 2015), infections (Bhat et al., 2019), and congenital disorders (Jaeken, 2013).

There is great interest in N-glycosylation from the biomedical and pharmaceutical industry, physicians, and patients because of its high therapeutical and diagnostic relevance. In many types of carcinoma, increases in fucosylation, branching, and sialylation occur (Almeida and Kolarich, 2016). Disialoganglioside is expressed by almost all neuroblastomas, and Phase I–III studies have shown that anti-disialoganglioside monoclonal antibodies can be successful against those (Ho et al., 2016; Ahmed and Cheung, 2014). Poly-α2,8-sialylation, for example, increases the half-lives of antibodies but does not lead to tolerance problems (Van Landuyt et al., 2019). Conversely, the presence of glycans foreign to humans can be detrimental to a therapeutic. N-glycolylneuraminic acid is immunogenic to humans (Padler-Karavani et al., 2008) but is present in some CHO-cell-derived glycoproteins (Hokke et al., 1995).

Despite these critical functions of N-glycosylation in biotherapeutical contexts and multiple developments in this field, such as the genetic engineering of CHO cells to increase glycoprotein sialylation (Bork et al., 2007), some challenges persist. Proteins can be glycosylated in multiple locations, so any investigations need to elucidate not only the glycan compositions but also where each glycan is located (Almeida and Kolarich, 2016). This structural and regional diversity makes it challenging to determine specific functions of N-glycans (Schjoldager et al., 2020). An important goal for modeling is obtaining the complete N-glycan distribution for a given protein or biopharmaceutical. As defined by Seber and Braatz (2025), "the numerical N-glycan distribution is, for a known glycosylation site X, what percentage of proteins have glycan A at-

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

tached to that site, what percentage of proteins have glycan B attached to that site, and so forth for every glycan to obtain the complete glycan distribution for a site, then so forth for every site to obtain the complete glycan distribution for all sites of a protein".

To assist in better understanding and predicting N-glycosylation, many computational models have been created, which may be subdivided into mechanistic and data-driven models. Mechanistic models use physical knowledge, typically in the form of differential equations, to make predictions. They require little-to-no process data and always output physics-constrained answers; however, they also demand significant understanding of the system and can be slow (Willard et al., 2022), a problem alleviated by model simplifications (Shen et al., 2020; Derbalah et al., 2022). Data-driven models directly leverage experimental data to make predictions. They require zero domain-based knowledge (but can benefit from it), are typically fast once trained, and the more complex data-driven models are universal function approximators (Cybenko, 1989; Hornik, 1991; Willard et al., 2022); however, they require high amounts of high-quality data due to their high variance, can produce non-physical outputs, and are typically not interpretable (Willard et al., 2022; Rudin, 2019). Most works on N-glycosylation, particularly those on predicting N-glycan distributions, use mechanistic models due to a lack of high-quality data and data-driven modeling knowledge. The literature on mechanistic models for the prediction of N-glycan distributions is extensive; some examples can be found in the reviews of Štor et al. (2021) and Kontoravdi and Jimenez del Val (2018). Significant works using data-driven models for the same task include Liang et al. (2020) and Seber and Braatz (2025).

An alternative to mechanistic and data-driven models are hybrid models. Hybrid models combine these two types of models to create something with the advantages of both and the disadvantages of neither (Willard et al., 2022). Although of high interest, the construction of hybrid models is not straightforward. First, they require knowledge of both mechanistic and data-driven modeling to be successfully implemented. Second, hybrid learning encompasses many architectures and model integration methods, and there have not been systematic studies on how to determine the best hybrid learning method for a given problem or system. Multiple examples of these architectures can be found in Willard et al. (2022), Aykol et al. (2021), and Liao and Köttig (2014). One of those architectures is the Residual Hybrid Model,[1] in which a data-driven model learns the residuals (prediction errors) of a mechanistic model (Su et al., 1992). Despite the simplicity of this hybrid architecture, it has found success in many scientific problems (Su et al., 1992; Thompson and Kramer, 1994; Forssell and Lindskog, 1997; Aykol et al., 2021; Willard et al., 2022). An illustration of this architecture is available in Fig. 1; other illustrations are also available in Fig. 1-A1 of Aykol et al. (2021) and Fig. 3 of Willard et al. (2022).

Mechanistic models receive the X and y data as inputs during training and generate predictions $\hat{y}_{\text{Mech}}$ that approximate y. These predictions can be compared with the real values y to determine the error $\varepsilon_{\text{Mech}}$ of each prediction, such that $\varepsilon_{\text{Mech}} = y - \hat{y}_{\text{Mech}}$. The data-driven models of residual hybrid models receive the X data and $\varepsilon_{\text{Mech}}$ values as inputs during training and generate predictions $\hat{y}_{\text{Data}}$ that approximate $\varepsilon_{\text{Mech}}$. These predictions $\hat{y}_{\text{Data}}$ are combined with the mechanistic predictions $\hat{y}_{\text{Mech}}$ to form $\hat{y}_{\text{Hybrid}} = \hat{y}_{\text{Mech}} + \hat{y}_{\text{Data}}$. Again, these predictions can be compared with the real values y to determine the error $\varepsilon_{\text{Hybrid}}$ of each prediction, such that $\varepsilon_{\text{Hybrid}} = y - \hat{y}_{\text{Hybrid}} = y - \hat{y}_{\text{Mech}} - \hat{y}_{\text{Data}} = \varepsilon_{\text{Mech}} - \hat{y}_{\text{Data}}$. Because $\hat{y}_{\text{Data}}$ approximates $\varepsilon_{\text{Mech}}$, it is thus argued that $\varepsilon_{\text{Hybrid}} < \varepsilon_{\text{Mech}}$.

In this work, we construct residual hybrid models by combining mechanistic models from the literature with Lasso-Clip-EN (LCEN) (Seber and Braatz, 2024) and artificial neural network (ANN) models trained by us. The data-driven models are trained using an efficient automatic machine learning (AutoML) software developed by us that completely eliminates the need for the end-user to have any knowledge in data-driven modeling. This software is free and open-source. These hybrid models are first used to predict the distribution of N-glycans attached to antibodies produced by Chinese hamster ovary (CHO) cells under different culture conditions. Then, the models are used to predict metrics that are relevant to biopharmaceutical production in CHO cells, including the titer, the galactosylation index of the products, and the levels of different chemicals in the culture medium over time. These CHO cultures were done in perfusion, fed-batch, and batch bioreactors depending on the dataset. Our hybrid models reduce the average prediction error by 152-fold on independent test sets when compared to the mechanistic models, and always lead to a reduction in the average test-set prediction error on the datasets investigated in this work. For the largest dataset, residual hybrid models have a test-set error that is 736-fold smaller than that of the mechanistic model. This work differentiates itself from previous works that used residual hybrid models by being the first to apply this method to real biopharmaceutical datasets, by confirming the superior performance of

[1] Not to be confused with the residual connections (also called skip connections) in some deep learning models.

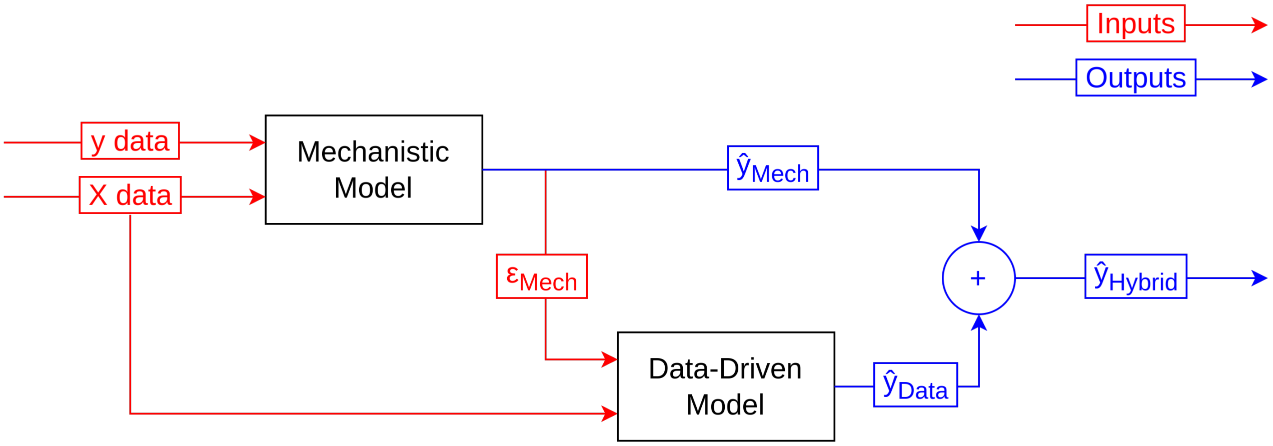

Figure 1: The residual hybrid model architecture. $\hat{y}$ refers to the predictions made by a certain model. $\varepsilon_{\text{Mech}} = y - \hat{y}_{\text{Mech}}$ refers to the residuals (errors) of the predictions made by the mechanistic model. Although not shown in the figure, a similar $\varepsilon_{\text{Hybrid}} = y - \hat{y}_{\text{Hybrid}}$ can be obtained, and it is argued that $\varepsilon_{\text{Hybrid}} < \varepsilon_{\text{Mech}}$.

residual hybrid models over mechanistic model on a variety of datasets, by using multiple data-driven methods as baselines, and by being the first to use residual hybrid modeling in an AutoML context.

2 METHODS

This section describes the datasets and the residual hybrid models training methods. Details on the mechanistic models from previous works are in Section A4. Instructions on running our AutoML software or recreating this article's results are provided in Supplementary Information and in our GitHub repo, available at github.com/PedroSeber/SmartProcessAnalytics.

2.1 Datasets

Four previously published works provide the data used to train the hybrid models in this work. The data are all in numerical/tabular form, and all contain a single output per relevant target (e.g.: the % presence of an N-glycan for every N-glycan or the concentration of a metabolite for every metabolite). The first dataset, an 11×4 dataset for each N-glycan from Karst et al. (2017), comprises the levels of N-glycans on monoclonal antibodies produced in a CHO perfusion culture as a function of viable cell density and the concentration of galactose and manganese ions. The second, a 79×10 dataset for each N-glycan from Villiger et al. (2016), comprises the levels of N-glycans on monoclonal antibodies produced in a CHO fed-batch culture, but as a function of pH, galactose concentration, and manganese ion concentration under different conditions and feeding strategies. This dataset also includes a time-based component, and galactose or manganese supplementation at specific time points was performed in some samples. The third, a 7×9 dataset for each measurement from Kotidis et al. (2019), comprises the titer and galactosylation index of monoclonal antibodies produced in CHO fed-batch culture as a function of feed galactose and uridine. Finally, the fourth, a 216×1 autoregression dataset for each metabolite from Kastelic et al. (2019), comprises the levels of different chemicals (such as amino acids and metabolites) in the culture media over time.

Data are split between cross-validation (CV) and test sets. For the Karst et al. (2017) and Villiger et al. (2016) datasets, the same splits used in these works are used here, and 3-fold CV and 4-fold CV (respectively) are used. For the Kotidis et al. (2019) dataset, points FS2, FS3, and FS7 are used as the test set; the first two because they are outside of the design spec set by Kotidis et al. (2019), and the last because the model of Kotidis et al. (2019) had the highest prediction error on that sample. 4-fold CV is also used. For the Kastelic et al. (2019) dataset, all points in the death phase ($t > 135$ hr) are used as the test set, and 5-fold timeseries CV is used. These choices of test sets avoid test set leakage by ensuring the test set is sufficiently different from the training/cross-validation set. For robustness, the CV procedure is repeated 10 times for each combination of hyperparameters.[2] The (repeated) CV procedure and scaling are performed automatically by our AutoML software (Section A1). Before each training step, all data are automatically scaled based on the training data's mean and standard deviation, such that the scaled training data has mean = 0 and standard deviation = 1.

[2] Except for the procedure for Kastelic et al. (2019), which uses time series CV due to the nature of its dataset.

2.2 Data-driven and residual hybrid models

Ordinary least-squares (OLS), elastic net (EN), LCEN, support vector machine with radial basis functions (SVM), random forest (RF), AdaBoost, and ANN models are directly trained on the data using our AutoML software to serve as additional baselines. OLS, EN, SVM, RF, and AdaBoost models are constructed with scikit-learn (Pedregosa et al., 2011), LCEN models are constructed as per Seber and Braatz (2024), and ANN models (specifically, multilayer perceptrons [MLPs] and recurrent neural networks [RNNs]) are constructed with PyTorch (Paszke et al., 2019) within our AutoML software. Furthermore, LCEN and ANN models are trained on the residuals of the mechanistic models to create residual hybrid models. A list of the hyperparameters used for each model architecture is available in Section A2. The best combination of hyperparameters for each model and task is determined by grid search, and the combination with the lowest cross-validation average loss (averaged over 10 repeats) is selected. Errors on an independent test dataset are then reported. This overall procedure is repeated 3 times with different cross-validation seeds[3] such that each test error reported is the mean ± standard deviation.

3 RESULTS

3.1 Residual hybrid models improve N-glycan distribution predictions

This section includes the results of the models trained with the datasets of Karst et al. (2017) and Villiger et al. (2016). Karst et al. (2017) featured two forms of models: a mechanistic model that used differential equations (named "Mechanistic" in this work) and a response surface methodology (RSM) model with intercept, linear, and 2nd-degree interaction terms. The mechanistic model provides a better fit to independent test data. Nevertheless, these models still had high errors for some non-minor glycan forms; for example, the mechanistic model had a 47.9% and a 9.31% relative error when predicting the amount of antibodies with high-mannose (Man) and FA2G2 glycosylation respectively (Table 1). As per Section 2.2, we trained data-driven models to expand the models serving as a baseline and an MLP-based residual hybrid model. The OLS and EN models are very similar to the RSM model, but they lack the interactions present in the RSM model. Despite that, they provided similar (slightly worse) performance, indicating that the contribution of the interaction terms is limited yet not significant. LCEN and MLPs had higher prediction accuracy than RSM for all four glycans (Table 1). These results indicate that nonlinear terms can be important if they are not binary interaction terms, which is corroborated by how LCEN frequently selected features of the form $\sqrt{x}$, $\log x$, and $1/(x_j x_k)$. SVM, RF, and AdaBoost models sometimes performed better than the RSM model and sometimes worse. Furthermore, with the exception of the RF and MLP models to predict levels of Man, all of these data-driven models were inferior to the mechanistic model. It is likely that the chief reason for these higher percent relative errors (PREs) is the scarcity of data, as only 8 points are available for model training. Despite this shortage of data and the high accuracy of the mechanistic model for most glycans, a residual hybrid model composed of the mechanistic model followed by an MLP always achieved a lower PRE than the mechanistic model (Table 1). The residual hybrid model led to 2.37-fold [2.25–2.45] average reductions in the relative errors of the mechanistic model. These great results highlight how residual hybrid models are useful in predicting N-glycan distributions even when few data points have been collected and a strong mechanistic model is already in use. They also highlight the effectiveness of our AutoML method for training an MLP to succeed the mechanistic model in this hybrid architecture.

To further validate the ability of residual hybrid models to achieve higher accuracy than mechanistic models, a second work with this type of data was used. Villiger et al. (2016) included only a mechanistic model, again using differential equations and named "Mechanistic" in this work. Villiger et al. (2016)'s mechanistic model had worse fits than that of Karst et al. (2017), as the former had low errors only for FA2G0, average errors for FA2G1, and high errors for the other two glycans (Table S1). Once again, we trained data-driven models to expand the models serving as a baseline and an MLP-based residual hybrid model. Despite the slightly increase in the amount of training data, the OLS and EN models performed poorly — their predictions were worse than predicting with the average of the training set. The only exception was the EN model trained to predict FA2G2 levels, which displayed a good performance, which even surpassed the mechanistic model. LCEN, SVM, RF, and AdaBoost had mixed results, surpassing the training mean and the mechanistic model in a few cases. Once again, these results highlight the importance of nonlinearities to predict the distribution of N-glycans. The final data-driven model, an MLP model, had the best performance out of all models in this task. All of the MLP predictions surpassed the train mean and the mechanistic model, reaching test-set prediction errors 1.5-fold [1.49–1.56] smaller on average than the

[3] Except for the procedure for Kastelic et al. (2019), which uses time series CV due to the nature of its dataset.

Table 1: Mean ± standard deviation test-set percent relative errors (PREs) for different models predicting the levels of major N-glycans on the dataset of Karst et al. (2017). Models "Mechanistic" and "RSM" are from Karst et al. (2017); their PREs are obtained from the published data within. Models "OLS", "EN", "LCEN", "SVM", "RF", "AdaBoost", and "MLP" are data-driven models from this work used as baselines. Model "Mechanistic + MLP" is a residual hybrid model from this work. "Train Mean" is the mean of the training data. The lowest PREs are highlighted in bold.

Model	Man	FA2G0	FA2G1	FA2G2
Mechanistic (From Karst et al. (2017))	47.9	2.18	2.42	9.31
RSM (From Karst et al. (2017))	147	27.7	15.3	28.2
OLS (Baseline)	67.5	38.9	23.3	29.3
EN (Baseline)	71.2±0.0	37.0±0.9	21.7±1.1	28.9±0.3
LCEN (Baseline)	57.6±3.0	23.0±3.0	11.6±0.9	23.6±4.0
SVM (Baseline)	66.0±3.3	29.8±1.5	15.6±0.4	26.7±0.4
RF (Baseline)	**21.7±3.9**	28.7±0.5	18.7±0.9	49.5±1.3
AdaBoost (Baseline)	34.9±0.0	26.3±0.0	16.5±0.0	39.3±0.0
MLP (Baseline)	40.1±1.7	22.8±3.5	15.0±0.9	20.9±0.6
Mechanistic + MLP (This Work)	32.2±0.5	**1.33±0.2**	**1.04±0.2**	**2.44±0.3**
Train Mean (From Karst et al. (2017))	75.7	39.9	26.1	82.3

mechanistic model (Table S1). The Mechanistic + MLP residual hybrid model also consistently made predictions with lower errors than the pure mechanistic model, reducing its errors by 1.2-fold [1.18–1.22] on average. Surprisingly, the residual hybrid model was not as good as a pure MLP model in this dataset, but the difference in prediction errors was statistically insignificant for the FA2G0 and FA2G2 glycans. We hypothesize this difference exists because the mechanistic model for this dataset is not as accurate as the one in Karst et al. (2017), because there are additional data available for training, and potentially because the different features and culture settings are more challenging for models that include mechanistic parts (including the pure mechanistic model). These results again confirm the ability of our AutoML method to train strong-performing MLPs, as both models that included MLPs surpassed the other methods.

3.2 Residual hybrid models also improve other important predictions for biopharmaceutical production

Although predicting the distribution of N-glycans is an important task, there are also other values and metrics of interest for biopharmaceutical production. The dataset of Kotidis et al. (2019) comprises titer and galactosylation index measurements for CHO-cell produced antibodies under different feed conditions (FS1–7). Kotidis et al. (2019) also trained a mechanistic model based on differential equations, named "Mechanistic" in this work. Their model had medium-low errors for half of the titer and most of the galactosylation index predictions (Table 2, top half). However, these errors are train-set errors, as Kotidis et al. (2019) used all of their data points to train their model, so comparisons with independent test points are not available. We separated points FS2, 3, and 7 to form a test set; the first two because they are the out-of-specification points (Kotidis et al., 2019), and FS7 because it was the point for which the Mechanistic model had the highest prediction errors. As per Section 2.2, we trained data-driven models to expand the models serving as a baseline and an MLP-based residual hybrid model. For all model types, one set of models for the titer prediction task and another for the galactosylation index task were trained. Due to the low number of training samples (4) and higher number of features (8) than samples, the OLS, EN, and LCEN models had overfit results (despite the use of regularization) when predicting antibody titers, with zero error in all training-set predictions, but higher errors than the Mechanistic model for the test-set predictions.[4] The SVM, RF, and AdaBoost models also had low test-set performance despite strong training-set performances. The MLP model performed considerably better and even surpassed the Mechanistic model on all test-set predictions (Table 2, top half). Finally, the residual hybrid model (again, the mechanistic model followed by an MLP) achieved the best overall performance, surpassing both the "Mechanistic" model of Kotidis et al. (2019) and the purely data-driven MLPs. The residual hybrid model reduces the average test-set error of the mechanistic model by 8.2-fold [5.5–11.2] on the antibody titer prediction task (Table 2, top half).

Models for the galactosylation index prediction task had fewer overfitting issues. Nevertheless, this task was more challenging, as all data-driven models had a lower performance than the Mechanistic model (Table 2, bottom half). As before, the MLP model achieved the best purely data-driven results, but its test-set errors were still slightly higher than those of the Mechanistic model. Only the residual hybrid model returns more accurate predictions than the Mechanistic model, and it does so for all data points (Table 2, bottom half). On average, the residual hybrid model reduces the test-set error of the mechanistic model by 12.5-fold [11.4–14.8] on the galactosylation index prediction task. These two results further corroborate the potential of residual hybrid models for tasks beyond predicting N-glycan distributions, and highlight the ability of our AutoML software to train powerful MLPs even in a data-scarce setting.

The final task investigated involves predicting the concentration of nutrients and metabolites in the medium used to cultivate CHO cells. The dataset of Kastelic et al. (2019) comprises the levels of 19 chemicals and the amount of biomass in the medium over a culture period of 215 hours. The chemicals include multiple amino acids, sugars, and ammonium. In addition to the large amounts of data gathered by Kastelic et al. (2019), this dataset is also distinct because each measurement of interest was collected over multiple time points, allowing time series modeling to be done. Kastelic et al. (2019) trained a flux-based kinetic mechanistic model (named "Mechanistic" in this work) consisting of 103 chemical reaction and transport equations (Table 1 of that work). This mechanistic model had varying performance depending on the prediction task. For example, it performed very well when predicting levels of glucose, lactate, valine, or isoleucine; however, it performed poorly when predicting levels of ammonium, alanine, and biomass. We trained data-driven and residual hybrid models based on the LCEN and RNN architectures to predict levels of lactate, ammonium, biomass, glutamate, aspartate, and asparigine. These dynamic models were trained to output next-hour levels[5] based on the levels 1–5 hours prior, with the cutoff depending on the architecture and task. All data-driven and residual hybrid models had more accurate test-set predictions than the mechanistic model (Table 3). In addition, the residual hybrid models trained with a given architecture surpassed the pure data-driven model with the same architecture in all but one case. Overall, the best residual hybrid architectures reduced the test-set prediction error by 736-fold on average (Table 3) and were able to follow the experimental measurements with significant accuracy for both the training and testing periods (Fig. 2). These tests further validate the capabilities of residual hybrid models in yet another context and even despite the fact that the mechanistic model they were based on had limited performance for some metabolites.

4 DISCUSSION

This study constructs residual hybrid models from literature data on the distribution of N-glycans, properties relevant for antibody production, and concentrations of metabolites in CHO cell culture. These datasets not only were built for different tasks (but all relevant for biopharmaceutical production) but also consist of different features, including culture conditions and even a purely autoregressive dataset. As a comparative baseline, purely data-driven models are also trained and tested on the same datasets. The residual hybrid models significantly and consistently had higher prediction accuracy over the mechanistic models, and outperformed the data-driven models in most tasks. Among the four datasets used in this work, residual hybrid models reduce the test-set prediction error of the corresponding mechanistic models by 152-fold on average and always lead to reductions in test-set error for all predicted variables.

The first two datasets (Karst et al., 2017; Villiger et al.,

[4] Keep in mind that the errors for the Mechanistic model are all training-set errors, so they are biased downwards.

[5] Predicting further in the future without major increases in error is simple; see Table 6 of Seber and Braatz (2024) for example.

Table 2: Test-set percent relative errors (PRE) for different models predicting titers (top table) or galactosylation indices (bottom table) for each feed strategy on the dataset of Kotidis et al. (2019). Model "Mechanistic" is from Kotidis et al. (2019); its PREs are obtained from the published data within. All data points were used to train the "Mechanistic" model in Kotidis et al. (2019), so all PREs are train-set values for this model only. Model labels are as in Table 1. The lowest mean test PREs are highlighted in bold.

Model	Train set				Test set			
	FS1	FS4	FS5	FS6	FS2	FS3	FS7	Mean PRE
Mechanistic (Kotidis et al. (2019))	10.1	5.0	11.1	3.1	14.6	1.9	18.7	11.7
OLS (Baseline)	0	0	0	0	12.3	11.0	25.1	16.1
EN (Baseline)	0±0	0±0	0±0	0±0	11.3±0	7.3±0	24.9±0	14.5±0
LCEN (Baseline)	0±0	0±0	0±0	0±0	18.8±0	5.9±0	29.5±0	18.1±0
SVM (Baseline)	0.7±0	0.8±0	18.3±0	2.3±0	26.8±0	15.2±0	23.2±0	21.7±0
RF (Baseline)	2.1±0	0.4±0	7.6±0	2.0±0	20.6±0	9.7±0	23.8±0	18.0±0
AdaBoost (Baseline)	0±0	0±0	17.4±0	0±0	25.7±0	14.2±0	22.6±0	20.8±0
MLP (Baseline)	0.7±0.4	0.3±0.2	0.4±0.2	0.1±0.0	4.1±2.1	1.2±0.9	11.3±6.7	5.5±1.4
Mechanistic + MLP (This Work)	0.4±0.4	0.3±0.1	0.6±0.4	0.5±0.4	1.9±1.4	1.1±0.5	1.6±0.4	**1.5±0.4**
Train Mean (Kotidis et al. (2019))	4.1	2.5	14.5	5.6	22.6	11.4	25.7	19.9

Model	Train set				Test set			
	FS1	FS4	FS5	FS6	FS2	FS3	FS7	Mean PRE
Mechanistic (Kotidis et al. (2019))	10.3	14.2	5.2	17.0	10.5	10.8	29.0	16.8
OLS (Baseline)	0	0	0	0	49.1	60.2	30.3	46.5
EN (Baseline)	23.0±0	5.3±0	3.4±0	7.5±0	52.7±0	53.0±0	25.6±0	43.8±0
LCEN (Baseline)	2.1±0	0.7±0	0.1±0	2.0±0	61.4±0	56.5±0	22.7±0	46.9±0
SVM (Baseline)	0.2±0	0.1±0	0.1±0	0.1±0	48.9±0	57.6±0	30.5±0	45.6±0
RF (Baseline)	19.1±0	0.0±0	0.7±0	1.2±0	61.5±0	39.6±0	26.7±0	42.6±0
AdaBoost (Baseline)	0±0	0±0	0±0	0±0	59.9±0	41.0±0	24.5±0	41.8±0
MLP (Baseline)	3.4±0.9	1.2±0.2	0.8±0.8	1.5±0.7	16.6±9.5	8.9±6.0	29.9±2.1	18.5±5.0
Mechanistic + MLP (This Work)	1.5±1.1	0.6±0.5	0.3±0.1	0.3±0.2	0.8±0.3	1.7±0.5	1.6±0.7	**1.4±0.2**
Train Mean (Kotidis et al. (2019))	29.9	7.9	0.3	12.4	47.2	29.9	33.9	37.0

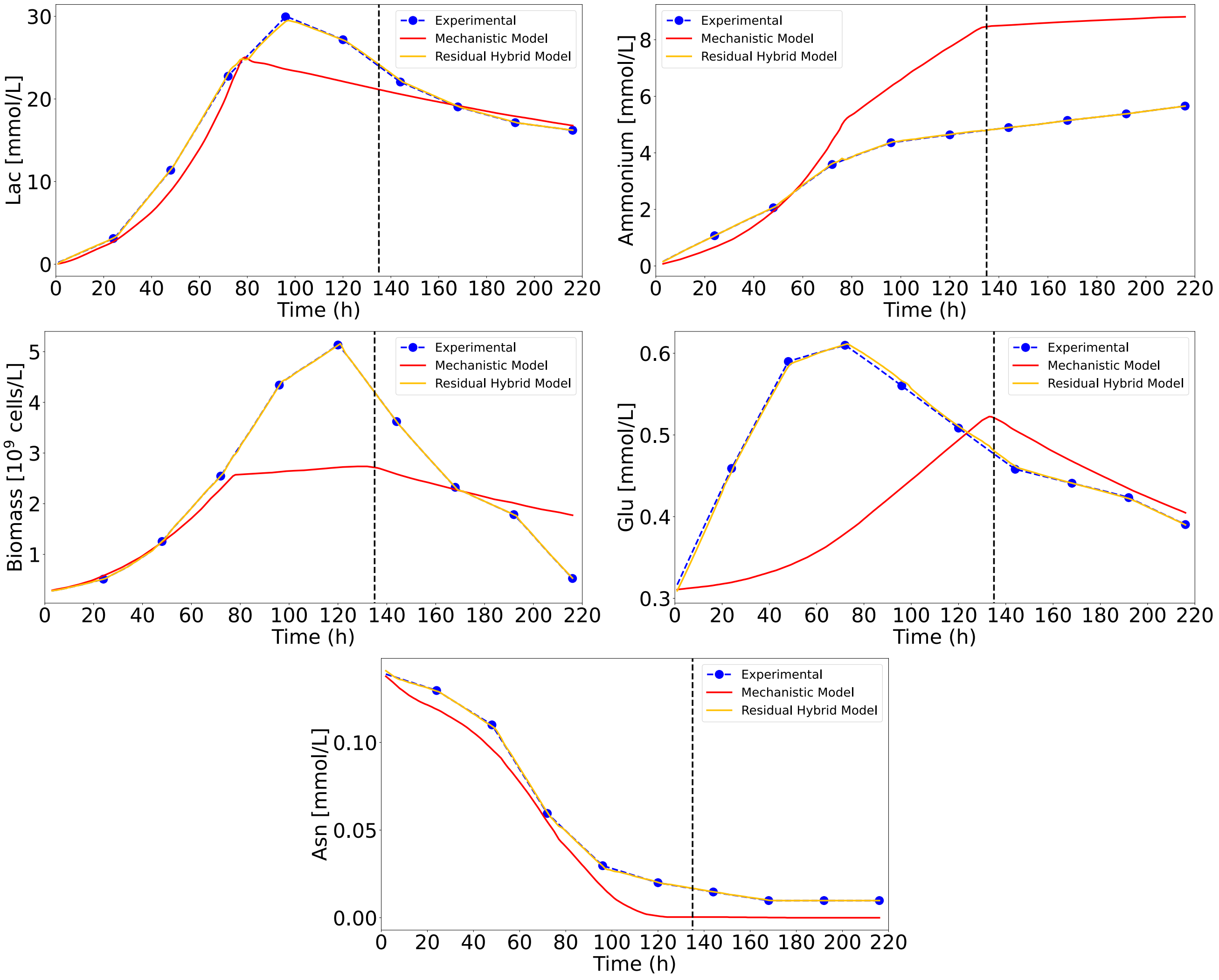

Figure 2: Experimental values and predictions for the mechanistic model (Kastelic et al., 2019) and the best residual hybrid model ("Mechanistic + LCEN" or "Mechanistic + RNN", this work) for selected metabolites from the Kastelic et al. (2019) batch culture dataset. The vertical black dotted line separates the training period (t $\leq$ 135 h) from the test period.

2016) involve predicting N-glycan distributions based on the viable cell density, levels of metabolites (such as Mn and Gal) in the culture, and pH (Section 3.1). Two models were trained by Karst et al. (2017) on their dataset: a mechanistic and an RSM model. The mechanistic model obtained much better results on that dataset. As a baseline, other data-driven models were trained by us. Only an MLP model was able to surpass the mechanistic model, and only for one of the four N-glycans (Table 1). On the other hand, a residual hybrid model consisting of the mechanistic model of Karst et al. (2017) followed by an MLP trained by us was able to reduce the relative errors by 2.37-fold when compared to the mechanistic model, and reductions in prediction errors occurred for all N-glycans (Table 1). Villiger et al. (2016) trained a mechanistic model on another dataset for the same task, which we also use to further validate the ability of residual hybrid models to lower prediction errors. The mechanistic model of Villiger et al. (2016) had higher relative errors (which may be explained by their dataset's being more challenging for prediction than that of Karst et al. (2017)), but it was still relatively accurate. In this task, some data-driven models were able to surpass the mechanistic model: an LCEN model was better than the mechanistic model on 2/4 N-glycans, and an MLP model surpassed the mechanistic model on all N-glycans (Table S1). A residual hybrid model, which was built in the same manner as that used for the Karst et al. (2017) dataset, also surpassed the mechanistic model on all N-glycans, reducing the relative prediction errors by 1.2-fold on average (Table

Table 3: Average test-set percent relative errors (PRE) for different models for predicting metabolite and biomass levels on the dataset of Kastelic et al. (2019). Model "Mechanistic" is from Kastelic et al. (2019); its PREs are obtained from Fig. 6 of that work. Models "LCEN" and "RNN" are data-driven models from this work used as baselines. Models "Mechanistic + LCEN" and "Mechanistic + RNN" are residual hybrid models from this work. "Train Mean" is the mean of the training data. The lowest PREs are highlighted in bold.

Model	Lac	Ammonium	Biomass	Glu	Asn
Mechanistic (From Kastelic et al. (2019))	3.87	66.1	35.6	5.71	99.0
LCEN (Baseline)	**0.07**	0.17	1.15	3.14	13.6
RNN (Baseline)	1.06	1.68	2.44	0.77	1.77
Mechanistic + LCEN (This Work)	0.36	**0.02**	**0.23**	3.01	10.4
Mechanistic + RNN (This Work)	0.34	0.20	1.98	**0.25**	**0.53**
Train Mean (From Kastelic et al. (2019))	9.88	44.2	60.9	20.5	591

S1). For the dataset of Villiger et al. (2016) only, a pure data-driven MLP led to greater reduction in prediction errors than the Mechanistic + MLP residual hybrid model, but the differences in prediction errors were statistically insignificant for 2/4 N-glycans. This difference in performance may be due to the lower accuracy of the mechanistic model in this task, because there are additional data for training (relative to the dataset of Karst et al. (2017), for example), and because the different features and culture settings in the dataset of Villiger et al. (2016) may be more challenging for models that include mechanistic components.

To highlight how residual hybrid models are widely applicable, models were then trained on datasets containing other relevant metrics for biopharmaceutical production. These include the titer, indices that serve as a proxy for N-glycosylation, and culture metabolite levels. Kotidis et al. (2019) trained mechanistic models on seven different feed conditions to predict antibody titers and galactosylation indices. Their mechanistic model had medium-low errors for about half of these 14 predictions; however, all of the data points were used to train that mechanistic model, so the errors are biased downwards. Again, data-driven models were trained by us as a baseline, but most of these performed poorly primarily due to overfitting issues. A notable exception was the MLP model trained to predict titers, which had a test-set error 2.3-fold lower than that of the mechanistic model (Table 2). Residual hybrid models surpassed all of the mechanistic and data-driven models on these tasks, leading to a 8.2-fold average reduction in test-set errors for titer predictions and 12.5-fold average reduction in test-set errors for galactosylation index predictions, and reducing prediction errors for every sample (Table 2).

A fourth dataset comprised of the levels of important metabolites over time was used. Kastelic et al. (2019) trained a flux-based kinetic mechanistic model on these data. The model achieved mixed success: it was very accurate for some metabolites, but inaccurate for others. As the data of Kastelic et al. (2019) were time series, LCEN models with $lag > 0$ and RNN models were trained by us both as purely data-driven models and as residual hybrid models. For all five metabolites tested, the data-driven and residual hybrid models surpassed the mechanistic model of Kastelic et al. (2019) (Table 3). Furthermore, in all but one case, the residual hybrid model containing a given architecture surpassed the data-driven model of the same architecture. On average, the residual hybrid models led to a 736-fold reduction in test-set prediction error (Table 3) and were able to follow the experimental measurements with significant accuracy for both the training and testing periods (Fig. 2).

Overall, this work attests the high potential of residual hybrid models to substantially reduce the errors of mechanistic models in a variety of tasks, and the high capabilities of our AutoML software to train accurate data-driven and residual hybrid models. The AutoML software used in this work is publicly available, allowing reproduction of this work and its use for other tasks and datasets. The software is simple to install and use, allowing even non-specialists in data-driven or residual hybrid models to train and use powerful models for any predictive task, including tasks not related to N-glycosylation or biopharmaceutical production.

Acknowledgments

This study is supported by the U.S. Food and Drug Administration, Contract No. 75F40121C00090. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the financial sponsor. The authors would like to thank Dr. Thomas Villiger for providing the raw data of Karst et al. (2017) and Villiger et al. (2016), which were used to train and test the models in Section 3.1 of this work.

References

Ahmed, M. and Cheung, N.-K. V. (2014). Engineering anti-GD2 monoclonal antibodies for cancer immunotherapy, *FEBS Letters* **588**(2): 288–297.

Almeida, A. and Kolarich, D. (2016). The promise of protein glycosylation for personalised medicine, *Biochimica et Biophysica Acta (BBA) – General Subjects* **1860**(8): 1583–1595. Glycans in personalised medicine.

Aykol, M., Gopal, C. B., Anapolsky, A., Herring, P. K., van Vlijmen, B., Berliner, M. D., Bazant, M. Z., Braatz, R. D., Chueh, W. C. and Storey, B. D. (2021). Perspective—combining physics and machine learning to predict battery lifetime, *Journal of The Electrochemical Society* **168**(3): 030525.

Bhat, A. H., Maity, S., Giri, K. and Ambatipudi, K. (2019). Protein glycosylation: Sweet or bitter for bacterial pathogens?, *Critical Reviews in Microbiology* **45**(1): 82–102.

Bork, K., Reutter, W., Weidemann, W. and Horstkorte, R. (2007). Enhanced sialylation of EPO by overexpression of UDP-GlcNAc 2-epimerase/ManAc kinase containing a sialuria mutation in CHO cells, *FEBS Letters* **581**(22): 4195–4198.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals, and Systems* **2**: 303–314.

Derbalah, A., Al-Sallami, H., Hasegawa, C., Gulati, A. and Duffull, S. B. (2022). A framework for simplification of quantitative systems pharmacology models in clinical pharmacology, *British Journal of Clinical Pharmacology* **88**(4): 1430–1440.

Forssell, U. and Lindskog, P. (1997). Combining semi-physical and neural network modeling: An example of its usefulness, *IFAC Proceedings Volumes* **30**(11): 767–770.

Ho, W.-L., Hsu, W.-M., Huang, M.-C., Kadomatsu, K. and Nakagawara, A. (2016). Protein glycosylation in cancers and its potential therapeutic applications in neuroblastoma, *Journal of Hematology & Oncology* **9**(1): 100.

Hokke, C. H., Bergwerff, A. A., Dedem, G. W. K., Kamerling, J. P. and Vliegenthart, J. F. G. (1995). Structural analysis of the sialylated N- and O-linked carbohydrate chains of recombinant human erythropoietin expressed in Chinese hamster ovary cells. Sialylation patterns and branch location of dimeric N-acetyllactosamine units, *European Journal of Biochemistry* **228**(3): 981–1008.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks, *Neural Networks* **4**(2): 251–257.

Imperiali, B. and O'Connor, S. E. (1999). Effect of N-linked glycosylation on glycopeptide and glycoprotein structure, *Current Opinion in Chemical Biology* **3**(6): 643–649.

Jaeken, J. (2013). Chapter 179 – congenital disorders of glycosylation, *in* O. Dulac, M. Lassonde and H. B. Sarnat (eds), *Pediatric Neurology Part III*, Vol. 113 of *Handbook of Clinical Neurology*, Elsevier, Amsterdam, pp. 1737–1743.

Karst, D. J., Scibona, E., Serra, E., Bielser, J.-M., Souquet, J., Stettler, M., Broly, H., Soos, M., Morbidelli, M. and Villiger, T. K. (2017). Modulation and modeling of monoclonal antibody N-linked glycosylation in mammalian cell perfusion reactors, *Biotechnology and Bioengineering* **114**(9): 1978–1990.

Kastelic, M., Kopač, D., Novak, U. and Likozar, B. (2019). Dynamic metabolic network modeling of mammalian Chinese hamster ovary (CHO) cell cultures with continuous phase kinetics transitions, *Biochemical Engineering Journal* **142**: 124–134.

Kontoravdi, C. and Jimenez del Val, I. (2018). Computational tools for predicting and controlling the glycosylation of biopharmaceuticals, *Current Opinion in Chemical Engineering* **22**: 89–97.

Kotidis, P., Demis, P., Goey, C. H., Correa, E., McIntosh, C., Trepekli, S., Shah, N., Klymenko, O. V. and Kontoravdi, C. (2019). Constrained global sensitivity analysis for bioprocess design space identification, *Computers & Chemical Engineering* **125**: 558–568.

Liang, C., Chiang, A. W., Hansen, A. H., Arnsdorf, J., Schoffelen, S., Sorrentino, J. T., Kellman, B. P., Bao, B., Voldborg, B. G. and Lewis, N. E. (2020). A Markov model of glycosylation elucidates isozyme specificity and glycosyltransferase interactions for glycoengineering, *Current Research in Biotechnology* **2**: 22–36.

Liao, L. and Köttig, F. (2014). Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction, *IEEE Transactions on Reliability* **63**(1): 191–207.

Ma, Y., Guo, J. and Braatz, R. (2024). Quasi-steady-state approach for efficient multiscale simulation and optimization of mAb glycosylation in CHO culture.
URL: *https://arxiv.org/abs/2409.00281*

Padler-Karavani, V., Yu, H., Cao, H., Chokhawala, H., Karp, F., Varki, N., Chen, X. and Varki, A. (2008). Diversity in specificity, abundance, and composition of anti-Neu5Gc antibodies in normal humans: Potential implications for disease, *Glycobiology* **18**(10): 818–830.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., pp. 8024–8035.

Patterson, M. C. (2005). Metabolic mimics: The disorders of N-linked glycosylation, *Seminars in Pediatric Neurology* **12**(3): 144–151.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, *Nature Machine Intelligence* **1**: 206–215.

Schjoldager, K. T., Narimatsu, Y., Joshi, H. J. and Clausen, H. (2020). Global view of human protein glycosylation pathways and functions, *Nature Reviews Molecular Cell Biology* **21**(12): 729–749.

Seber, P. and Braatz, R. D. (2024). LCEN: A novel feature selection algorithm for nonlinear, interpretable machine learning models. arXiv:2402.17120.

Seber, P. and Braatz, R. D. (2025). Linear and neural network models for predicting N-glycosylation in Chinese Hamster Ovary cells based on B4GALT levels, *Computers & Chemical Engineering* **194**: 108937.

Shen, L., Jacob, D. J., Santillana, M., Wang, X. and Chen, W. (2020). An adaptive method for speeding up the numerical integration of chemical mechanisms in atmospheric chemistry models: application to GEOS-Chem version 12.0.0, *Geoscientific Model Development* **13**(5): 2475–2486.

Stowell, S. R., Ju, T. and Cummings, R. D. (2015). Protein glycosylation in cancer, *Annual Review of Pathology: Mechanisms of Disease* **10**(1): 473–510.

Su, H.-T., Bhat, N., Minderman, P. and McAvoy, T. (1992). Integrating neural networks with first principles models for dynamic modeling, *IFAC Proceedings Volumes* **25**(5): 327–332.

Thompson, M. L. and Kramer, M. A. (1994). Modeling chemical processes using prior knowledge and neural networks, *AIChE Journal* **40**(8): 1328–1340.

Van Landuyt, L., Lonigro, C., Meuris, L. and Callewaert, N. (2019). Customized protein glycosylation to improve biopharmaceutical function and targeting, *Current Opinion in Biotechnology* **60**: 17–28.

Villiger, T. K., Scibona, E., Stettler, M., Broly, H., Morbidelli, M. and Soos, M. (2016). Controlling the time evolution of mAb N-linked glycosylation - Part II: Model-based predictions, *Biotechnology Progress* **32**(5): 1135–1148.

Willard, J., Jia, X., Xu, S., Steinbach, M. and Kumar, V. (2022). Integrating scientific knowledge with machine learning for engineering and environmental systems, *ACM Comput. Surv.* **55**(4).

Štor, J., Ruckerbauer, D. E., Széliová, D., Zanghellini, J. and Borth, N. (2021). Towards rational glycoengineering in CHO: from data to predictive models, *Current Opinion in Biotechnology* **71**: 9–17.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**. See Sections 2.2 and A2, along with Fig. 1.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Not Applicable**.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
   Yes. Provided in the Supplemental Materials and will be provided in an online repo once the paper is accepted.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results.
   Not Applicable

(b) Complete proofs of all theoretical results.
Not Applicable

(c) Clear explanations of any assumptions.
Not Applicable

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL).
   Yes. See Section A1 and the external repo (to be published after review).

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen).
   Yes. See Sections 2.1, 2.2, and A2.

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times).
   Yes. Standard deviations are used as per Section 2.2.

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).
   Yes. See Section A5.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets.
   Yes. Cited in Section 2.1.

   (b) The license information of the assets, if applicable.
   No.

   (c) New assets either in the supplemental material or as a URL, if applicable.
   Not Applicable

   (d) Information about consent from data providers/curators.
   No. Not included, although consent was obtained.

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content.
   Not Applicable.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots.
   Not Applicable. No crowdsourcing or research with human subjects was done in this work.

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable.
   Not Applicable.

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation.
   Not Applicable.

Improving N-Glycosylation and Biopharmaceutical Production Predictions Using AutoML-Built Residual Hybrid Models Supplementary Materials

A1 USING OUR AutoML SOFTWARE TO TRAIN NEW RESIDUAL HYBRID MODELS

A setup file defining the specific packages and version numbers used in this work is available as `setup.py` on our GitHub (github.com/PedroSeber/SmartProcessAnalytics). Instructions on using the AutoML software are also available in the README file and the Examples folder.

Our AutoML software automates the model setup and building (including hyperparameter optimization), the cross-validation procedures (including data scaling to the mean and standard deviation of the training data), and the reporting of results (including the final model's parameters and hyperparameters, and relevant metrics such as train-set and test-set (R)MSEs, relative errors, and R^2 values for regression tasks). Although not explored in this work, the software can also select the most appropriate methods based on the properties of the input data.

An important design consideration is that our AutoML software does not require any significant programming ability from the end-user. Surveyed hyperparameters and criteria are determined by inputs to Python functions, and all have appropriate default values. For example, to train an MLP model, the user simply needs to pass `model_name = ['MLP']` to the main AutoML function, and can manipulate important hyperparameters with other inputs (including but not limited to `MLP_layers, RNN_layers, batch_size, learning_rate, weight_decay, n_epochs, class_weight, scheduler`). No knowledge of any machine learning frameworks, such as sklearn or PyTorch, is necessary — in contrast to some other open-source tools — as the AutoML software converts human-readable values for these inputs into pre-programmed code.

A2 LIST OF HYPERPARAMETERS USED IN THIS WORK

All possible combinations of the hyperparameters below were cross-validated.

1. For the elastic net (EN) models: $\alpha = 0$ and 20 log-spaced values between -4.3 and 0 (as per `np.logspace(-4.3, 0, 20)`) and L_1 ratios = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.97, 0.99].
2. For the LCEN models: α and L_1 ratios as above. *degree* values = [1, 2, 3], *cutoff* values between 0 and 4×10^{-1}, and *lag* = 0 except for the Kastelic et al. (2019) dataset, which used *lag* between 1 and 5.
3. For the SVM models: with the number of features in a dataset as F, gamma values = [1/50, 1/10, 1/5, 1/2, 1, 2, 5, 10, 50]$/F$, C values = [0.01, 0.1, 1, 10, 50, 100], and epsilon values = [0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.3].
4. For the RF models: number of trees = [10, 25, 50, 100, 200, 300], maximum depth = [2, 3, 5, 10, 15, 20, 40], minimum samples per leaf (as a fraction of total samples) = [0.001, 0.01, 0.02, 0.05, 0.1], and a number of features (as a fraction of the total number) = [0.1, 0.25, 0.333, 0.5, 0.667, 0.75, 1.0].
5. For the AdaBoost models: number of trees as above and learning rates = [0.01, 0.05, 0.1, 0.2].
6. For the ANN models: The MLP hidden layer sizes varied for each dataset; typical sizes were 20–120 neurons per layer. One to three hidden layers were used. For the Kastelic et al. (2019) dataset, an LSTM cell with 30–60 neurons before the MLP and a lag between 1 and 4 were also used. Learning rates = [0.05, 0.1, 0.5], batch size = 32, the ReLU activation function, weight decays = [0.1, 0.5, 1], 40 epochs, and a cosine scheduler with a minimum learning rate equal to 1/16 of the original learning rate with 10 epochs of warm-up.

A3 ADDITIONAL TABLES

Table S1: Test-set percent relative errors (PRE) for different models predicting the levels of each major N-glycan on the dataset of Villiger et al. (2016). Model "Mechanistic" is from Villiger et al. (2016); its PREs are obtained from the published data within. Models "OLS", "EN", "LCEN", "SVM", "RF", "AdaBoost", and "MLP" are data-driven models from this work used as baselines. Model "Mechanistic + MLP" is a residual hybrid model from this work. "Train Mean" is the mean of the training data. The lowest PREs are highlighted in bold.

Model	Man	FA2G0	FA2G1	FA2G2
Mechanistic (From Villiger et al. (2016))	41.4	9.47	17.0	39.1
OLS (Baseline)	135	83.7	126	258
EN (Baseline)	129±5	39.4±3.0	55.7±4.6	101±47
LCEN (Baseline)	31.6±1.5	13.0±0.3	19.7±0.9	35.2±1.7
SVM (Baseline)	27.6±0.3	25.6±1.9	41.2±3.0	65.2±0.7
RF (Baseline)	21.4±0.0	13.5±0.2	19.2±0.5	37.1±0.7
AdaBoost (Baseline)	24.7±0.5	14.4±0.0	20.1±0.9	35.4±0.4
MLP (Baseline)	**19.3±0.8**	**7.72±0.43**	**11.6±0.2**	**30.7±0.9**
Mechanistic + MLP (This Work)	31.7±0.7	8.77±0.02	14.1±0.8	33.0±1.2
Train Mean (From Villiger et al. (2016))	25.1	11.1	19.1	39.0

A4 DESCRIPTION OF THE MECHANISTIC MODELS

The mechanistic models used in Karst et al. (2017), Villiger et al. (2016), and Kotidis et al. (2019) are very similar – Karst et al. (2017) states that their mechanistic model was adapted from Villiger et al. (2016)'s work. These mechanistic models consist of two parts: a cell culture model and a glycosylation model. The cell culture model is a continuous stirred-tank reactor model used to estimate cell growth rate, ammonia levels, specific productivities, and process-related values (Villiger et al., 2016; Karst et al., 2017; Kotidis et al., 2019). The equations for the cell culture model are available in Table II of Karst et al. (2017). The glycosylation model is a dynamic plug flow reactor model for the N-glycosylation reactions that occur in the Golgi apparatus. The glycosylation model consists of 43 chemical reactions for 33 different glycan structures as shown in Figure 1c of Karst et al. (2017) and Figure 2 of Villiger et al. (2016). Limitations of this approach are primarily related to limitations in the knowledge of this biological process, including a lack of transport parameters for specific nucleotide sugar donors, which required the assumption of a single parameter for all donors (Villiger et al., 2016; Karst et al., 2017); differences in productivities between cell lines, which required the assumption that enzyme concentrations are linearly correlated to that productivity (Villiger et al., 2016); the assumption of specific kinetics (such as Michaelis-Menten kinetics) for specific enzymes (Villiger et al., 2016; Karst et al., 2017; Kotidis et al., 2019); and the assumption that enzymes are distributed in a Gaussian manner in the Golgi apparatus and that this distribution is unchanged by pH levels (Villiger et al., 2016; Kotidis et al., 2019). Štor et al. (2021), which was published years after Karst et al. (2017), Villiger et al. (2016), and Kotidis et al. (2019), also mentions that "kinetic modelling is not yet fully optimized to provide precise predictions", indicating the modeling process itself has limitations.

The mechanistic model used in Kastelic et al. (2019) consist of two parts: a micro kinetic model for the interior of CHO cells and a macro reactions model. The micro kinetic model models a CHO cell metabolic network with stoichiometric and other biological constraints (Kastelic et al., 2019). Overall, 103 chemical reactions and 118 metabolites are included in this micro kinetic model, as described in Table 1 of Kastelic et al. (2019). The macro reactions model enforces flux constraints for the reaction network — specifically, the pseudo-steady-state assumption and flux balances with measured extracellular concentrations of metabolites (Table 2 of Kastelic et al. (2019)), then perform macro-scale material balances for the entire bioreactor (Kastelic et al., 2019). Limitations of this approach again involve limitations in the knowledge of this biological process, including the assumption that all metabolites are distributed uniformly within the cell (as opposed to having different concentrations due to organelles, for example) (Kastelic et al., 2019); and the use of the pseudo-steady-state approximation, which may be inaccurate. A recent paper has found that the pseudo-steady-state approximation is typically accurate for most cell culture conditions (Ma et al., 2024), so that approximation potentially is not the main source of any inaccuracies in Kastelic et al. (2019).

A5 COMPUTATIONAL RESOURCES USED

All experiments were done in a personal computer equipped with a 13th Gen Intel® Core™ i5-13600K CPU, 64 GB of DDR4 RAM, and an NVIDIA GeForce RTX 4090 GPU.