Supplemental Information for:

Designing Multi-Site Charge-Bifurcation Networks in *De Novo* Proteins: A Kinetic, Statistical, and Machine-Learning Approach

Xiao Huang[1], William F. DeGrado[2], Michael J. Therien[1], and David N. Beratan[1,3,4]

[1]*Department of Chemistry, Duke University, Durham, NC 27708, USA*

[2]*Department of Pharmaceutical Chemistry, University of California, San Francisco, San Francisco, CA 94143, USA*

[3]*Department of Biochemistry, Duke University, Durham, NC 27710, USA*

[4]*Department of Physics, Duke University, Durham, NC 27708, USA*

Email: david.beratan@duke.edu

Contents

S1 Parameters for Hole Transfer Rate Calculations

Table S1 summarizes the physical parameters used in the hole transfer (HT) kinetic analysis. These quantities include reorganization energies, vibrational coupling constants, tunneling decay parameters, and numerical convergence settings that define the non-adiabatic electron transfer rate (Eq. 1) and the electronic coupling model (Eq. 2) in the main text. Together, the parameters define the kinetic framework used in the main text to evaluate hole bifurcation dynamics, ensuring that the simulations are grounded in values consistent with biological charge transfer processes.

Table S1: Parameters used in the HT kinetic analysis.

Parameter	**Value / Description**
Outer sphere reorganization energy (λ_{outer})	0.9 eV[1]
Inner sphere reorganization energy (λ_{inner})	0.075 eV ($\hbar\omega \times D$)
High-frequency vibrational mode energy ($\hbar\omega$)	0.15 eV[2,3]
Electron-vibration coupling strength (D)	0.5[2,3]
Tunneling interaction decay constant (β)	0.6 Å^{-1} [2,3]
Electronic coupling constant (V_0)	0.01 eV[2,3]
Shortest edge-to-edge HT cofactor distance	5 Å
Numerical convergence criteria (n)	100 (sufficient for the sum in eq. 1 in the main text to converge)

S2 HB Network Parameter Optimization with Bayesian Optimization

To identify optimal configurations of hole bifurcation (HB) network parameters that maximize quantum yield, we utilized Bayesian optimization (BOp). This technique was implemented using the Python package `bayes_opt`[4] and is particularly well-suited for optimizing objective functions that are computationally expensive to evaluate, such as the quantum yield derived from our master equation analysis. The objective function, denoted as $f(\mathbf{x})$, is dependent on a vector of design parameters $\mathbf{x}$, which includes inter-cofactor distances (R_{ij}) and the redox potentials ($E_i^{(0)}$) of the cofactors.

BOp operates by constructing a surrogate model, which is a mathematical approximation of the true, computationally intensive objective function (in this case, the HB quantum yield). This surrogate allows for efficient exploration of the parameter space by providing predictions of the objective function's behavior based on previously evaluated points. For our BOp implementation, we selected a Gaussian Process (GP) as the surrogate model. A GP offers a probabilistic framework to model our objective function and its dependence on the HB network's structural and energetic parameters. A GP is characterized by a mean function, $\mu(\mathbf{x})$, and a covariance function (or kernel), $k(\mathbf{x}, \mathbf{x}^*)$. The kernel quantifies the correlation between the objective function's values at different points $\mathbf{x}$ and $\mathbf{x}^*$ in the parameter space, where $\mathbf{x}^*$ typically represents an unobserved point for which predictions are sought. The GP model is expressed as:

$$f(\mathbf{x}) \sim \text{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^*)).$$

The covariance function employed was the radial basis function (RBF) kernel:

$$k(\mathbf{x}, \mathbf{x}^*) = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}^*\|^2}{2l^2}\right),$$

where σ_f^2 represents the variance and l is the length scale parameter. These two hyperparameters dictate how rapidly the correlation diminishes with increasing distance $\|\mathbf{x} - \mathbf{x}^*\|^2$ between points and are typically optimized during the BOp process. The GP thus furnishes a predicted mean value $\mu(\mathbf{x})$ for the objective function at any point $\mathbf{x}$, along with a standard deviation $\sigma(\mathbf{x})$ quantifying the uncertainty of this prediction.

The BOp methodology iteratively refines the design parameters by updating the GP model with new observations. In each iteration, an acquisition function is evaluated to guide the selection of the subsequent parameter set for evaluation. This function balances the need for exploration (sampling regions of high uncertainty, indicated by $\sigma(\mathbf{x})$) against exploitation (focusing on regions where the predicted mean $\mu(\mathbf{x})$ is high). Within the context of optimizing our HB networks, exploration translates to assessing novel combinations of R_{ij} and $G_i^{(0)}$ to find potentially high-performing configurations, while exploitation involves fine-tuning already promising parameter sets. We employed the Upper Confidence Bound (UCB) acquisition function, a common choice for this balance:

$$\alpha(\mathbf{x}) = \mu(\mathbf{x}) + \kappa\sigma(\mathbf{x}),$$

Here, κ is a tunable parameter that modulates the exploration-exploitation trade-off. Larger κ values encourage sampling in less-certain regions, whereas smaller values prioritize refinement in areas with high predicted performance.

The BOp procedure unfolds as follows: Initially, the GP is constructed using a set of observed data points $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X}$ comprises the evaluated parameter combinations and $\mathbf{y}$ their corresponding quantum yields. Based on these observations, the posterior predictive distribution for the objective function at a new candidate point $\mathbf{x}^*$ is computed. The predicted mean and variance at $\mathbf{x}^*$ are given by:

$$\mu(\mathbf{x}^*) = \mathbf{k}^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y},$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{k}.$$

In these expressions, $\mathbf{K}$ is the covariance matrix derived from the observed input parameters $\mathbf{X}$ and their quantum yields $\mathbf{y}$. The vector $\mathbf{k}$ represents the covariances between the observed points and the new point $\mathbf{x}^*$, while $k(\mathbf{x}^*, \mathbf{x}^*)$ is the prior variance at $\mathbf{x}^*$. The term σ_n^2 accounts for noise in the observations, which was assumed to be zero in our study. The acquisition function $\alpha(\mathbf{x})$ is then maximized over the parameter space to select the next point for evaluation, $\mathbf{x}_{\text{next}}$:

$$\mathbf{x}_{\text{next}} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}).$$

The true objective function $f(\mathbf{x}_{\text{next}})$ is subsequently computed by running the full kinetic simulation. This new data point, $(\mathbf{x}_{\text{next}}, f(\mathbf{x}_{\text{next}}))$, is then used to update the GP. This iterative cycle of selection, evaluation, and updating continues until a convergence criterion is met, which, in this work, involved identifying

parameter configurations that achieve near-optimal quantum yields (approaching 100%).

Through this iterative refinement of the GP model and strategic balance between exploring new regions and exploiting known high-performance areas, BOp facilitates an efficient search of the high-dimensional parameter space, ultimately identifying HB network designs that maximize the quantum yield.

S3 Pearson Correlation Framework for Statistical Analysis of Generated HB Network Datasets

To analyze how structural and energetic parameters influence hole bifurcation (HB) network performance, we employed Pearson's correlation framework.[5] For two random variables X and Y, the Pearson correlation coefficient $r_{X,Y}$ is defined as eq. S1:

$$r_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}, \tag{S1}$$

where $\mathrm{cov}(X,Y)$ is the covariance of X and Y, and σ_X and σ_Y are their standard deviations. The coefficient ranges from -1 to $+1$, with positive values indicating a direct linear relationship, negative values an inverse relationship, and values near zero indicating little or no linear dependence.

This analysis was applied to the 130,000 network configurations generated as described in Sect. 2.2. The design parameters included reduction potentials ($E^{(0)}_{W_2}$, $E^{(0)}_{W_{L1}}$, $E^{(0)}_{W_{L2}}$, $E^{(0)}_{W_{H1}}$, $E^{(0)}_{W_{H2}}$, $E^{(0)}_{PMn}$), electrostatic repulsion ($E_{\text{repulsion}}$), pathway distances (R_{H}, R_{L}, R), the distance difference $\Delta R = R_{\mathrm{H}} - R_{\mathrm{L}}$, and the reorganization energy λ. These were correlated with the two performance metrics: quantum yield and energy efficiency.

The purpose of this framework is to provide a first-order, linearized assessment of how individual parameters track with performance. While the relationships in HB systems are intrinsically nonlinear, Pearson correlation analysis is useful as an initial screening tool to highlight which parameters are likely to exert strong effects, and whether those effects are generally positive or negative. These results establish a baseline for the more detailed regression and machine-learning analyses presented in the main text.

S4 XGBoost Classifier Training

To model the relationship between the input parameters (Table 1 main text, plus $\Delta R = R_H - R_L$) and the likelihood of achieving high performance, we trained an XGBoost (Extreme Gradient Boosting) classifier.[6] XGBoost is a highly efficient and widely used implementation of gradient boosted decision trees, known for its predictive accuracy and ability to capture complex non-linearities and feature interactions.

The task was formulated as a binary classification problem: predicting whether a given set of input parameters would result in a 'high' quantum yield (> 0.9) or a 'low' quantum yield (≤ 0.9). The combined dataset (130,000 configurations from BOp and random sampling) was split into a training set (80%) and a test set (20%) using stratified sampling to maintain the proportion of high/low yield samples in both sets.

The XGBoost model was trained on the training set using the 'xgboost' Python library. Key hyperparameters (e.g., "n_estimators", "max_depth", "learning_rate", "subsample", "colsample_bytree") were tuned using

randomized search with cross-validation on the training set to optimize performance, although default or commonly effective values often perform well. The objective function was set to "binary:logistic", and the evaluation metric during training was "logloss".

The performance of the final trained model was evaluated on the held-out test set using standard classification metrics: accuracy, precision, recall, and F1-score, reported per class and as macro/weighted averages (Table 3 in the main text). Precision measures the accuracy of positive predictions, while recall measures the proportion of actual positives that were correctly identified. Accuracy provides the overall fraction of correct predictions.

S5 SHAP (SHapley Additive exPlanations) Analysis

To interpret the trained XGBoost model and understand the influence of each input feature on the prediction of high vs. low quantum yield, we employed SHAP (SHapley Additive exPlanations).[7] SHAP is a game-theoretic approach that provides a unified framework for interpreting predictions from complex machine learning models. It calculates the contribution of each feature to the difference between a specific prediction and the baseline or average prediction across the dataset.

For a given prediction $f(\mathbf{x})$ based on input features $\mathbf{x}$, SHAP assigns an attribution value, the SHAP value ϕ_i, to each feature i. These values satisfy several desirable properties, including local accuracy (the sum of SHAP values for all features equals the difference between the prediction and the baseline) and consistency (a feature's contribution increases or stays the same if the model changes such that the feature's marginal contribution increases). For tree-based models like XGBoost, efficient algorithms exist to compute exact SHAP values. (`TreeExplainer` in the `shap` Python library[7])

The SHAP values were calculated for samples in the dataset (typically the test set or a representative subset). We utilized two primary visualizations in the main text (Fig. 4):

1. **SHAP Summary (Beeswarm) Plot (Fig. 4a):** This plot visualizes the SHAP values for every feature for many samples. Each point represents a single feature for a single sample. Its position on the x-axis is the SHAP value (impact on model output; positive pushes towards 'low yield', negative towards 'high yield' in our case, based on the model's output interpretation). The y-axis position corresponds to the feature. The color represents the original value of the feature for that sample (red=high, blue=low). This plot reveals not only the magnitude of impact but also its direction and correlation with the feature's value. Features are typically ranked by the sum of absolute SHAP values across all samples.

2. **Feature Importance (Bar) Plot (Fig. 4b):** This plot provides a global summary by showing the mean absolute SHAP value for each feature ($\frac{1}{M}\sum_{j=1}^{M}|\phi_i^{(j)}|$, where M is the number of samples). This ranks features based on their average impact magnitude on the model's predictions. The "sweet spot" annotations on this plot indicate the 20th to 80th percentile range of feature values observed within the subset of samples classified as having high yield.

These plots allow us to identify the most influential parameters and understand how their values drive the model towards predicting high or low quantum yield configurations.

References

[1] Winkler, J. R.; Gray, H. B. Long-Range Electron Tunneling. *J. Am. Chem. Soc.* **2014**, *136*, 2930–2939.

[2] Matyushov, D. V. Reorganization energy of electron transfer. *Phys. Chem. Chem. Phys.* **2023**, *25*, 7589–7610.

[3] Hopfield, J. J. Electron transfer between biological molecules by thermally activated tunneling. *Proc. Natl. Acad. Sci. U.S.A.* **1974**, *71*, 3640–3644.

[4] Nogueira, F. Bayesian Optimization: Open source constrained global optimization tool for Python. 2014–; https://github.com/bayesian-optimization/BayesianOptimization.

[5] James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An introduction to statistical learning: with applications in R*; Springer, 2013; Vol. 103.

[6] Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016; pp 785–794.

[7] Lundberg, S. M.; Lee, S.-I. A unified approach to interpreting model predictions. *NeurIPS* **2017**, *30*.