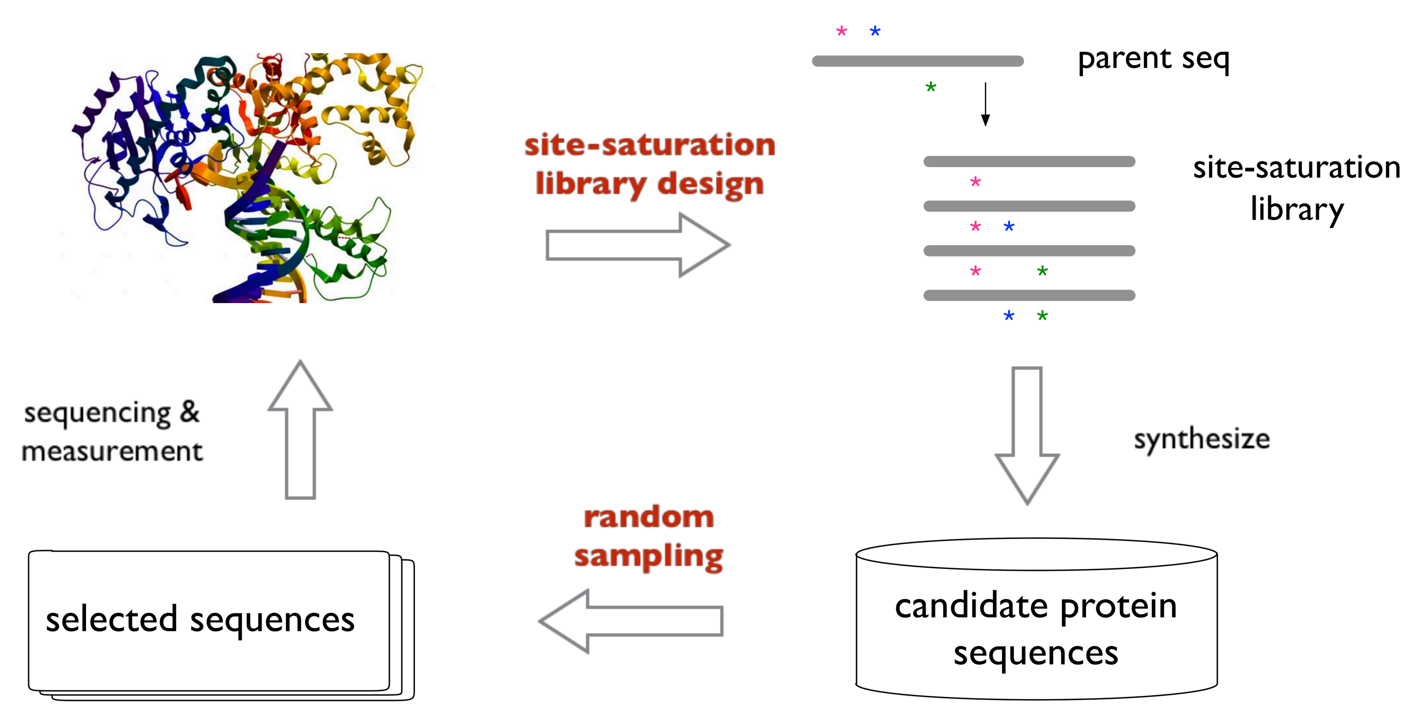
Batched stochastic Bayesian optimization via combinatorial constraints design

Kevin K. Yang,[1] Yuxin Chen,[2] Alycia Lee,[2] Yisong Yue[2]

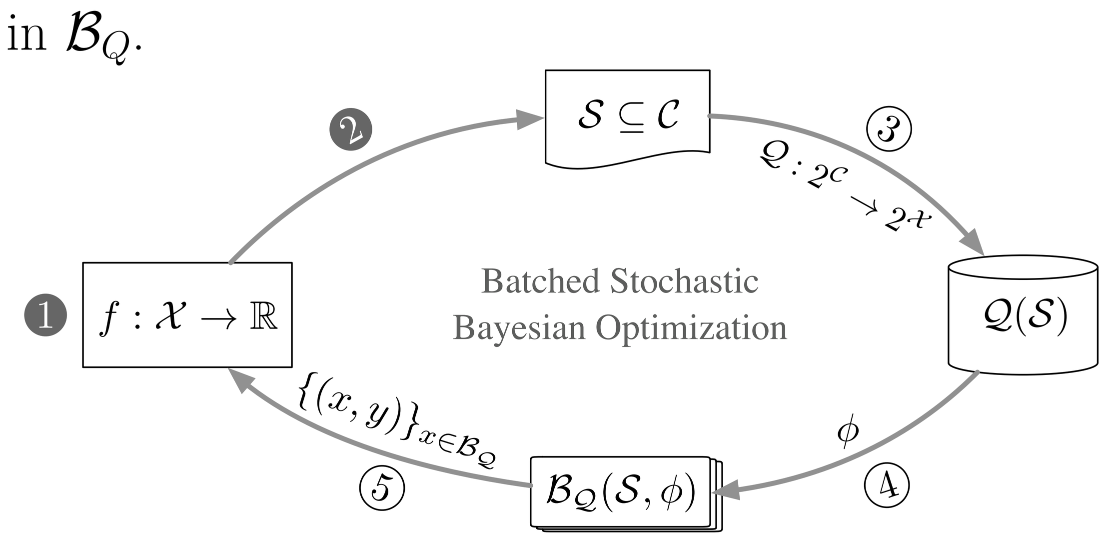
[1]FL57, Inc., [2]Department of Computing and Mathematical Sciences
California Institute of Technology

Problem setting



- User specifies constraints $\mathcal{S} \subseteq \mathcal{C}$
- Constraints generate a library: $Q(\mathcal{S})$, where $Q : 2^{\mathcal{C}} \to 2^{\mathcal{X}}$
- Use makes stochastic batched query $\mathcal{B}_Q$ from Q

At each iteration, want $\mathcal{S}$ that maximizes the (expected) number of improved items observed in $\mathcal{B}_Q$.



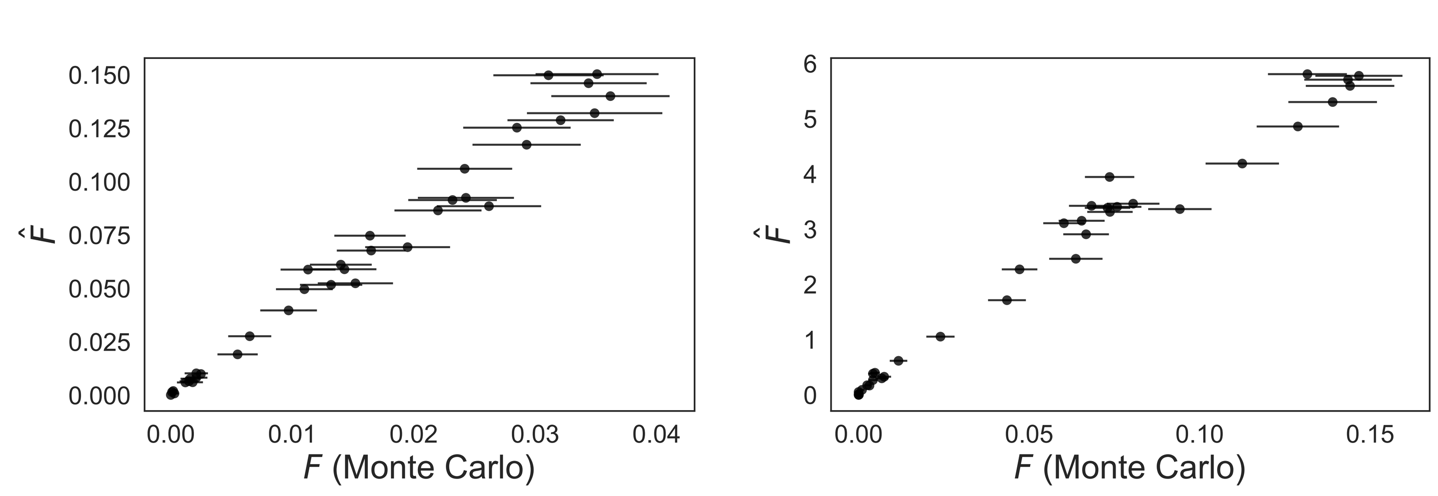
Objective

If the current best item has a value τ, then the objective is

$$F(\mathcal{S}) = \mathbb{E}_{\phi}\left[\sum_{x \in \mathcal{B}_Q(\mathcal{S})} \mathbb{1}(f(x) > \tau)\right] \tag{1}$$

Ignoring the dependencies, this becomes:

$$\hat{F}(\mathcal{S}) = \sum_{x \in Q(\mathcal{S})} \rho(x) \left[1 - \left(1 - \frac{1}{|Q(\mathcal{S})|}\right)^n\right] \tag{2}$$



(a) $\hat{F}$ vs. F for GB1 (b) $\hat{F}$ vs. F for PhoQ

Algorithm

Input: Constraints set $\mathcal{C} = \cup_{\ell=1}^{L} \mathcal{C}^{(\ell)}$; reward matrix $M = \{f(x)\}$; budget on each batch n
begin
Set up $\hat{F}$ from the inputs (M, n)
$h, g \leftarrow$ DSConstruct-SA$(\hat{F}, \mathcal{C})$ (or $h, g \leftarrow$ DSConstruct-DC$(\hat{F}, \mathcal{C})$)
$\mathcal{S}_{\text{cand}} \leftarrow \emptyset$; $\mathcal{S} \leftarrow$ Init$(\mathcal{C})$
while $\mathcal{S}$ *not converged* **do**
  $\mathcal{S}_{\text{cand}} \leftarrow \mathcal{S}_{\text{cand}} \cup$ LocalSearch$(\hat{F}, \mathcal{S}, \mathcal{C})$
  $\mathcal{S} \leftarrow$ ModMod$(h - g, \mathcal{S}, \mathcal{C})$ (or $\mathcal{S} \leftarrow$ SupSub$(h - g, \mathcal{S}, \mathcal{C})$)
$\mathcal{S}_{\text{cand}} \leftarrow \mathcal{S}_{\text{cand}} \cup \{\mathcal{S}\}$
$\mathcal{S}^* \leftarrow \arg\min_{\mathcal{S} \in \mathcal{S}_{\text{cand}}} \left\{\hat{F}(\mathcal{S})\right\}$
Output: Set of selected constraints $\mathcal{S}^*$

DS *via* submodular augmentation[1]

Any set function q can be written as DS: $h(\mathcal{S}) = q(\mathcal{S}) + \frac{|\beta'|}{\alpha} p(\mathcal{S})$ for any $\beta' < \beta$, $\beta = \min_{\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{C} \setminus j} \Delta_q(j \mid \mathcal{S}) - \Delta_q(j \mid \mathcal{S}')$, $\alpha = \min_{\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{C} \setminus j} \Delta_p(j \mid \mathcal{S}) - \Delta_p(j \mid \mathcal{S}') > 0$.

$$\hat{F}(\mathcal{S}) = \underbrace{\left(\hat{F}(\mathcal{S}) + \frac{|\beta'|}{\alpha} v(|\mathcal{S}|)\right)}_{g_1(\mathcal{S})} - \underbrace{\frac{|\beta'|}{\alpha} v(|\mathcal{S}|)}_{h_1(\mathcal{S})} \tag{3}$$

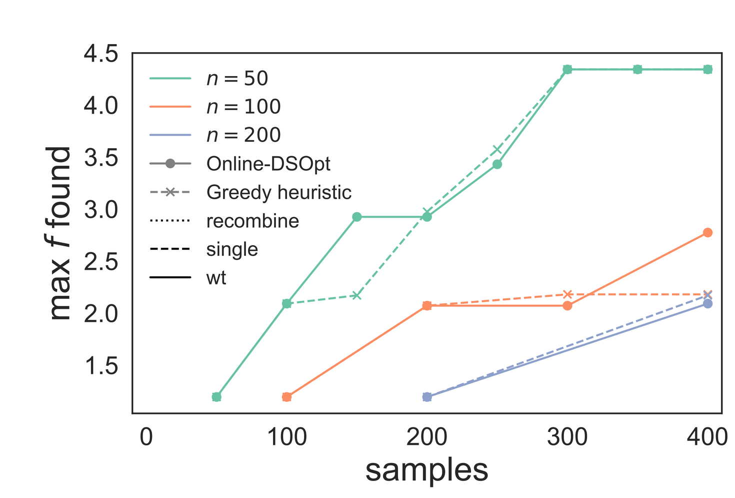
$$\beta' \leq \beta = \min_{\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{C} \setminus j} \Delta_{\hat{F}}(j \mid \mathcal{S}) - \Delta_{\hat{F}}(j \mid \mathcal{S}'). \tag{4}$$

DS *via* DC decomposition

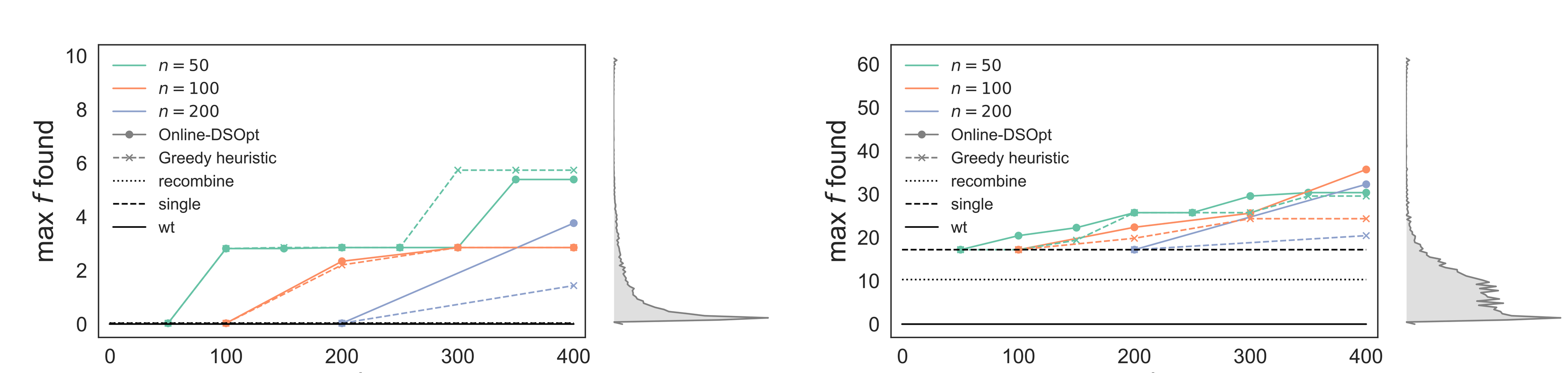
$$\hat{F}(\mathcal{S}) = \underbrace{\sum_{x \in Q(\mathcal{S})} (-\rho(x)) \cdot \left(r(|Q(\mathcal{S})|) + \frac{\beta}{\alpha} u(Q(|\mathcal{S}|))\right)}_{g_2(\mathcal{S})} - \underbrace{\sum_{x \in Q(\mathcal{S})} (-\rho(x)) \left(1 + \frac{\beta}{\alpha} u(Q(|\mathcal{S}|))\right)}_{h_2(\mathcal{S})} \tag{5}$$

where $u(x)$ is a non-negative, monotone convex function (in practice we set $u(x) = \frac{x^2}{2}$ and thus $\alpha = 1$.), $\alpha = \min_x u''(x)$, $r(x) = \left(1 - \frac{1}{x}\right)^n$, and $\beta = |\min_x r''(x)|$.

Synthetic datasets



Experimental protein datasets



(a) GB1[2] (b) PhoQ[3]

Conclusions

- 2 unique challenges: optimizing over the space of constraints instead of directly over items and stochastic sampling.
- Online batch optimization to search combinatorial design space
- 2 efficient DS decompositions
- Finds rare, highly-improved proteins with little screening effort

Acknowledgments

This work was supported in part by the Donna and Benjamin M. Rosen Bioengineering Center, NSF Award #1645832, Northrop Grumman, Bloomberg, and a Swiss NSF Early Mobility Postdoctoral Fellowship.

References

[1] Rishabh Iyer and Jeff Bilmes. "Algorithms for approximate minimization of the difference between submodular functions, with applications". In: *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*. AUAI Press. 2012, pp. 407–417.

[2] Nicholas C Wu et al. "Adaptation in protein fitness landscapes is facilitated by indirect paths". In: *Elife* 5 (2016), e16965.

[3] Anna I Podgornaia and Michael T Laub. "Pervasive degeneracy and epistasis in a protein-protein interface". In: *Science* 347.6222 (2015), pp. 673–677.