Parallel Thompson Sampling for Large-scale Accelerated Exploration of Chemical Space,

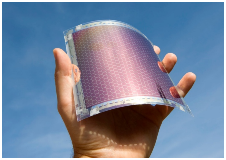
José Miguel Hernández–Lobato
Department of Engineering
University of Cambridge

http://jmhl.org, jmh233@cam.ac.uk

Joint work with James Requeima, Edward O. Pyzer-Knapp and Alan Aspuru-Guzik.

Drug and material design

Goal: find novel molecules that optimally fulfill various metrics.

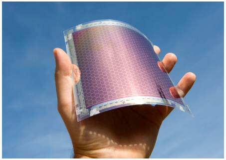
About 10^8 compounds in databases, potential ones: $10^{20} - 10^{60}$.

Challenges:

- Evaluating molecular properties is slow and expensive.
- Chemical space is huge.

Drug and material design

Goal: find novel molecules that optimally fulfill various metrics.

About 10^8 compounds in databases, potential ones: $10^{20} - 10^{60}$.

Challenges:

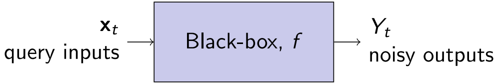
- Evaluating molecular properties is slow and expensive.
- Chemical space is huge.

Bayesian optimization can accelerate the search.

Bayesian optimization aims to efficiently optimize **black-box** functions:

$$\mathbf{x}^\star = \arg\max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$

No gradients, observations may be **corrupted by noise**.

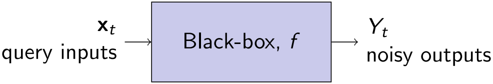


Black-box queries are **very expensive** (time, economic cost, etc...).

Bayesian optimization aims to efficiently optimize **black-box** functions:

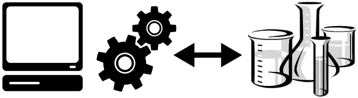
$$\mathbf{x}^\star = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

No gradients, observations may be **corrupted by noise**.

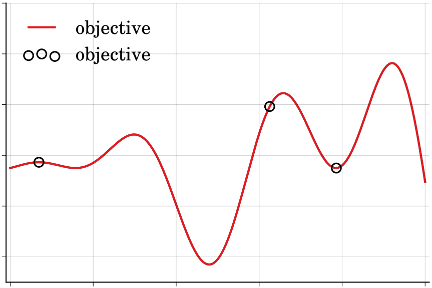


Black-box queries are **very expensive** (time, economic cost, etc...).

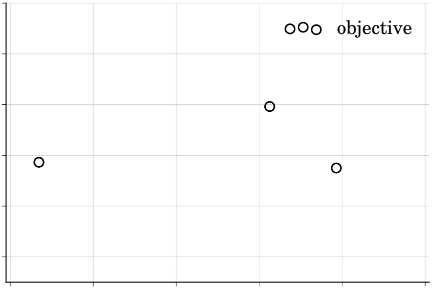
Main idea: replace expensive black-box queries with **cheaper computations** that will save additional queries in the long run.



1. Get initial sample.

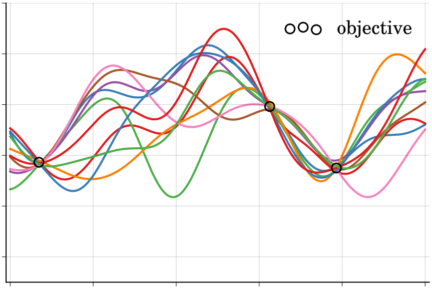


❶ **Get initial sample.**



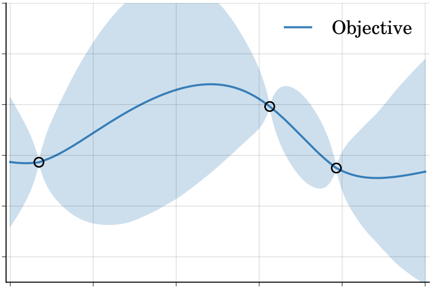
1. **Get initial sample.**
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D})\,.$$



1. Get initial sample.
2. **Fit a model to the data:**

$$p(y|\mathbf{x}, \mathcal{D}) \,.$$

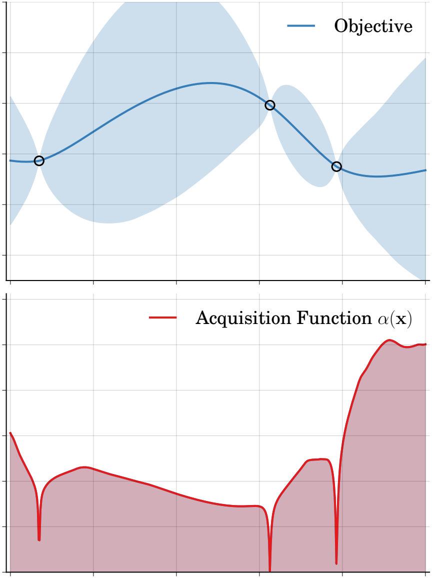


1. Get initial sample.
2. **Fit a model to the data:**

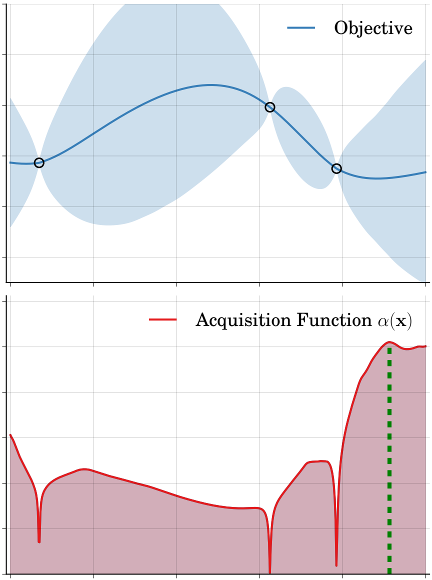
$$p(y|\mathbf{x}, \mathcal{D}) \,.$$

3. Select data collection strategy:

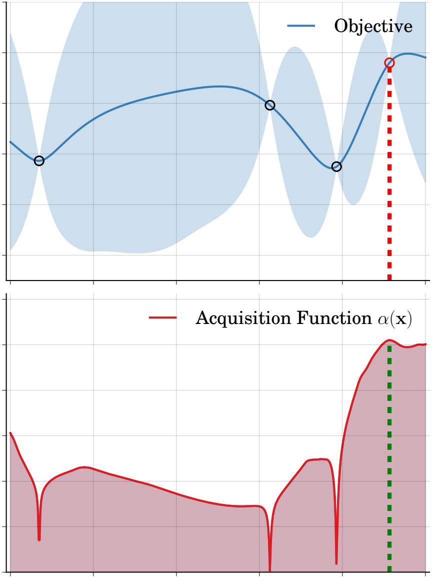
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})] \,.$$



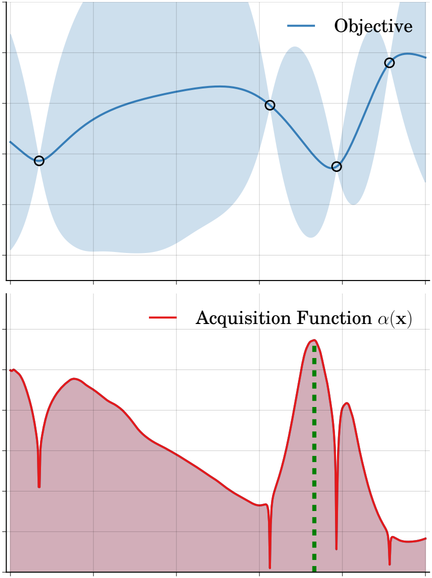
1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D})\,.$$
3. **Select data collection strategy:**
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})]\,.$$
4. Optimize acquisition function $\alpha(\mathbf{x})$.



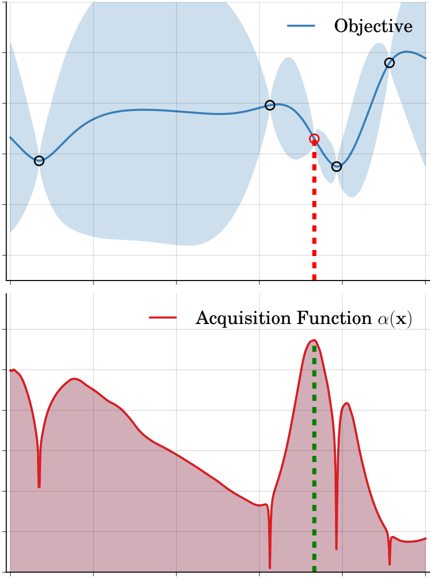
1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D}).$$
3. Select data collection strategy:
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})].$$
4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. Collect data and update model.



1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D}) \,.$$
3. Select data collection strategy:
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})] \,.$$
4. Optimize acquisition function $\alpha(\mathbf{x})$.
5. **Collect data and update model.**
6. Repeat!



1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D}) \,.$$
3. **Select data collection strategy:**
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})] \,.$$
4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



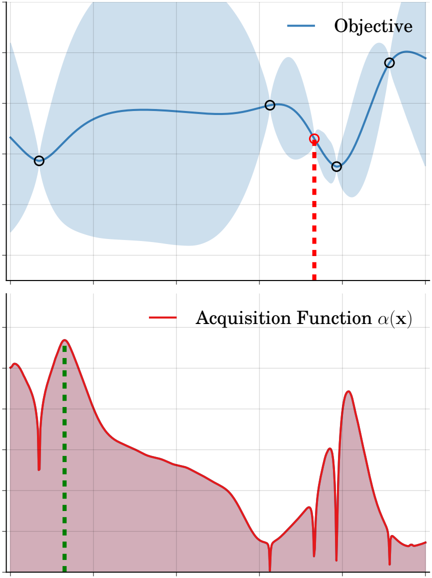
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D})\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



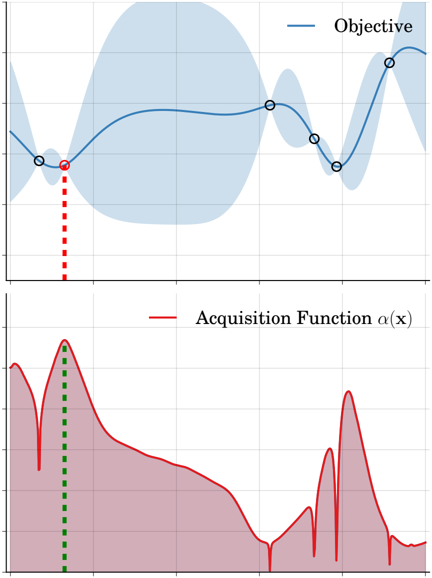
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D})\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



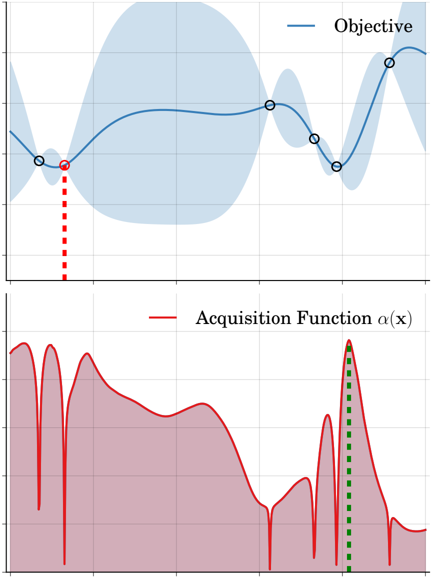
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}) \,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})] \,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



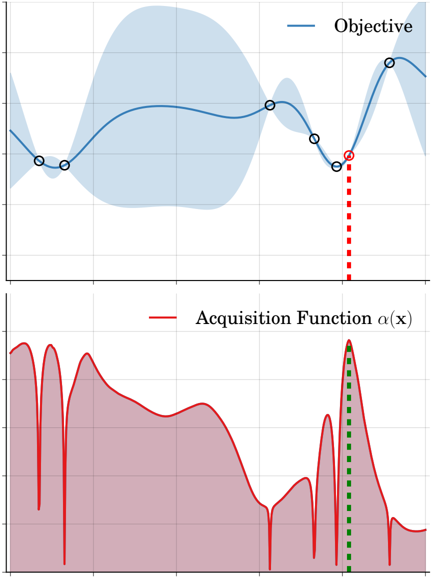
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D})\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



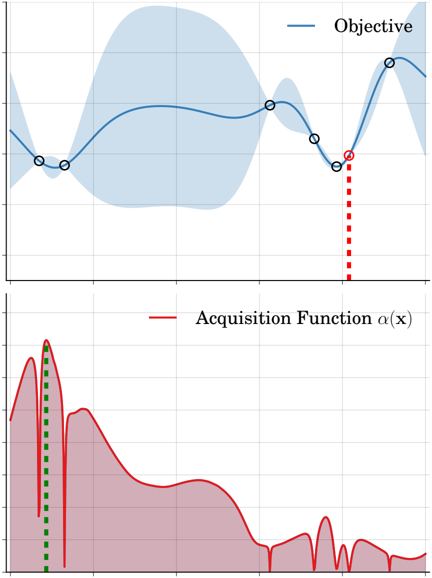
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D})\,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



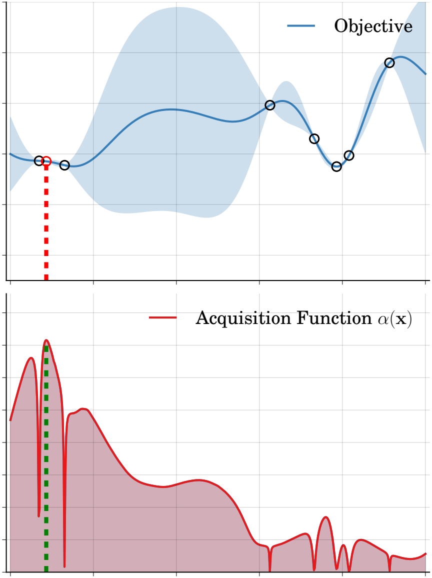
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}) \,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})] \,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



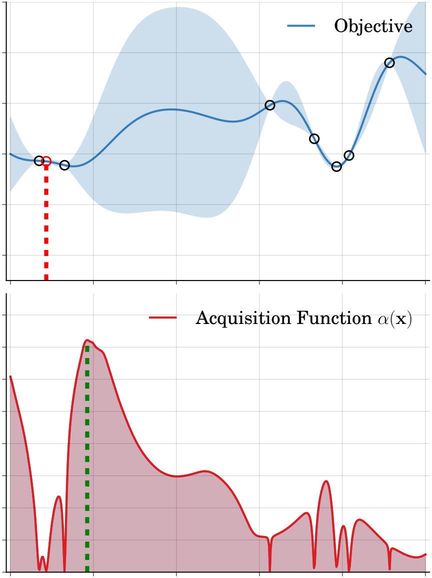
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}) \,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})] \,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



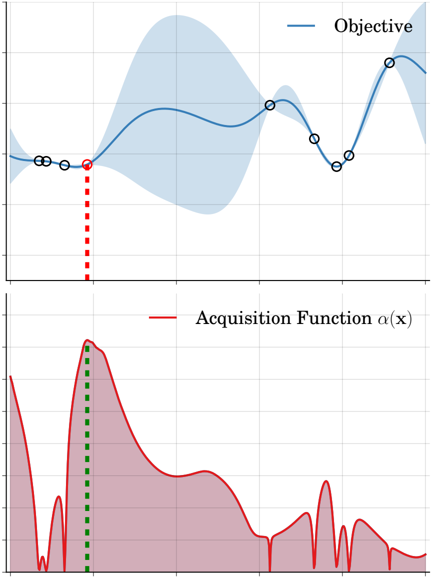
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D}) \,.$$

3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})] \,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



❶ Get initial sample.

❷ Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D})\,.$$

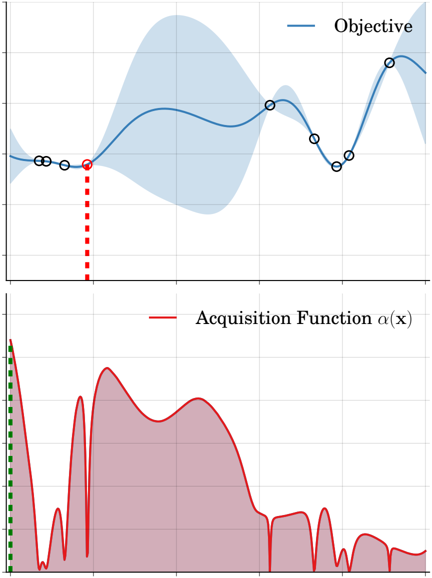
❸ Select data collection strategy:

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})]\,.$$

❹ Optimize acquisition function $\alpha(\mathbf{x})$.

❺ Collect data and update model.

❻ Repeat!



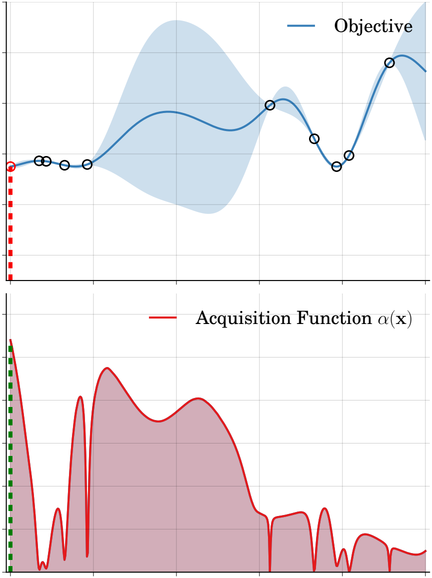
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D})\,.$$

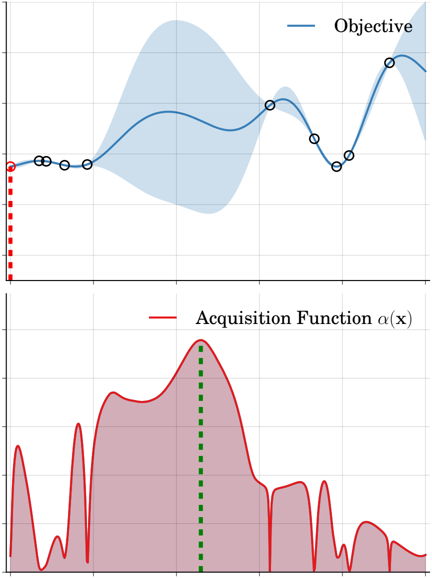
3. **Select data collection strategy:**

$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D}) \,.$$
3. **Select data collection strategy:**
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})] \,.$$
4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**



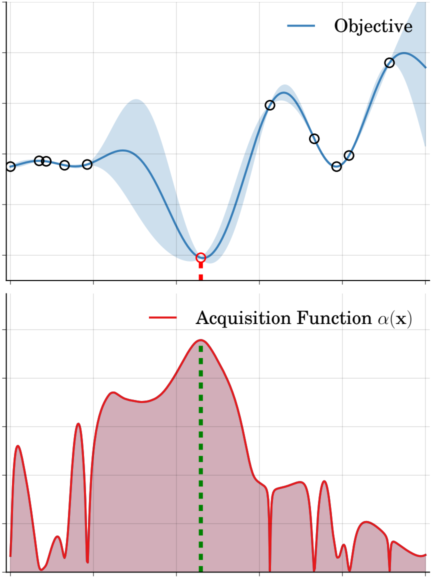
1. Get initial sample.
2. Fit a model to the data:

$$p(y|\mathbf{x}, \mathcal{D})\,.$$

3. **Select data collection strategy:**

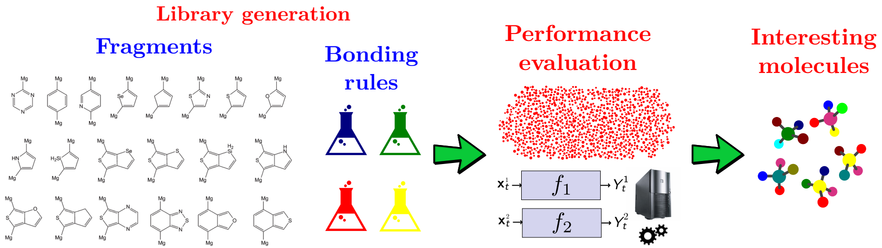
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x},\mathcal{D})]\,.$$

4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

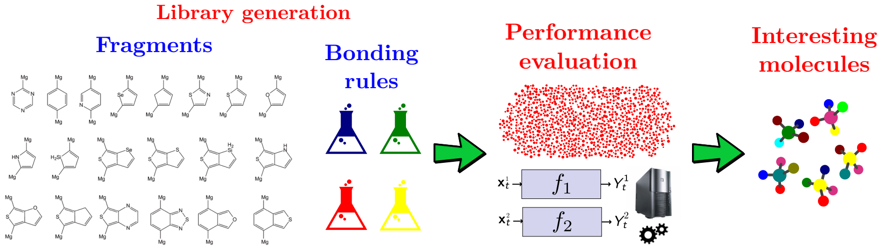


1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D})\,.$$
3. **Select data collection strategy:**
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D})}[U(y|\mathbf{x}, \mathcal{D})]\,.$$
4. **Optimize acquisition function $\alpha(\mathbf{x})$.**
5. **Collect data and update model.**
6. **Repeat!**

Discovering new optimal molecules

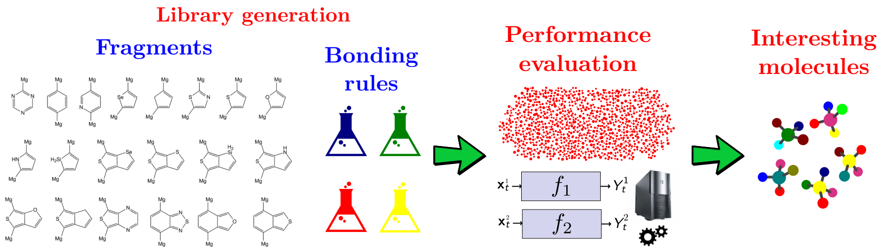


Discovering new optimal molecules



Bayesian optimization can **accelerate** the search!

Discovering new optimal molecules

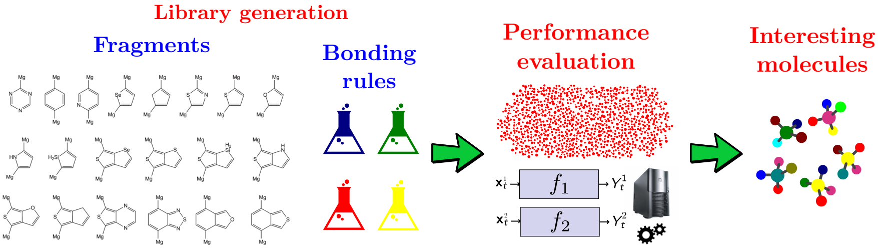


Bayesian optimization can **accelerate** the search!

Challenges:

1. Massive **libraries** with **millions of candidate molecules**.

Discovering new optimal molecules

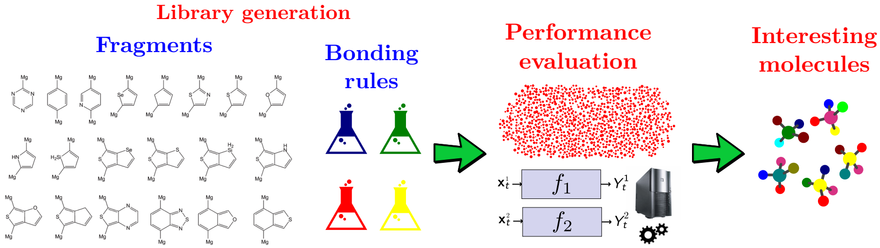


Bayesian optimization can **accelerate** the search!

Challenges:

1. Massive **libraries** with **millions of candidate molecules**.
2. Need to collect **hundreds of thousands of data points**.

Discovering new optimal molecules



Bayesian optimization can **accelerate** the search!

Challenges:

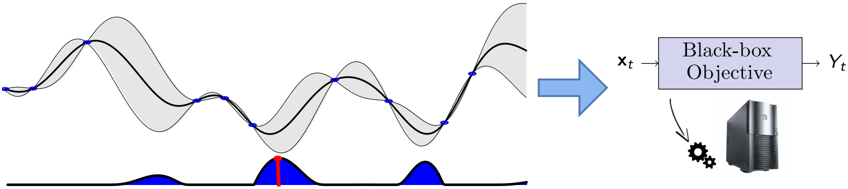
1. Massive **libraries** with **millions of candidate molecules**.
2. Need to collect **hundreds of thousands of data points**.
3. How to collect data in **parallel** efficiently? e.g. with a **computer cluster**.

Parallel Bayesian optimization

Traditional Bayesian optimization is **sequential**!

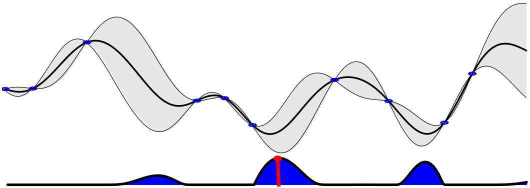
Parallel Bayesian optimization

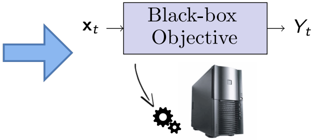
Traditional Bayesian optimization is **sequential**!



Parallel Bayesian optimization

Traditional Bayesian optimization is **sequential**!

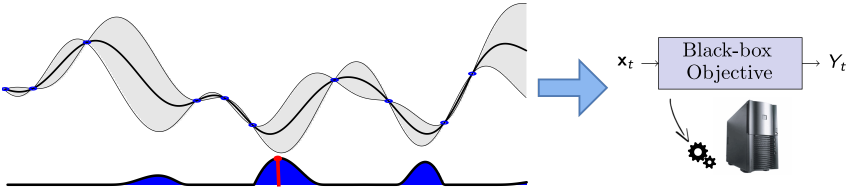




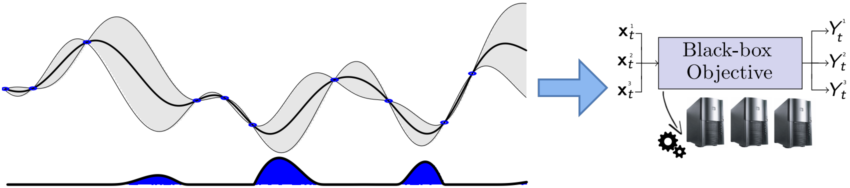
Computing clusters allow us to collect a **batch of data** at once!

Parallel Bayesian optimization

Traditional Bayesian optimization is **sequential**!

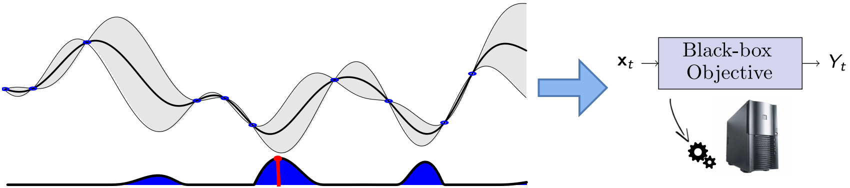


Computing clusters allow us to collect a **batch of data** at once!

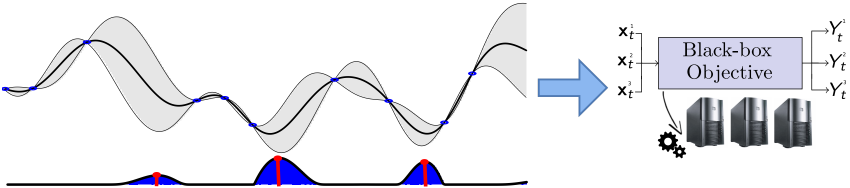


Parallel Bayesian optimization

Traditional Bayesian optimization is **sequential**!

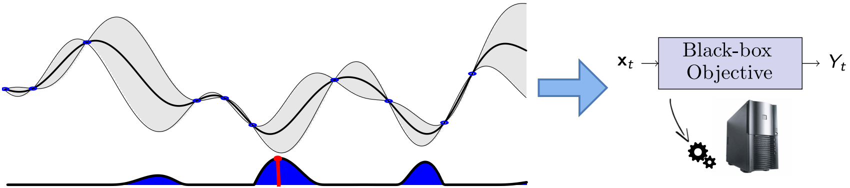


Computing clusters allow us to collect a **batch of data** at once!

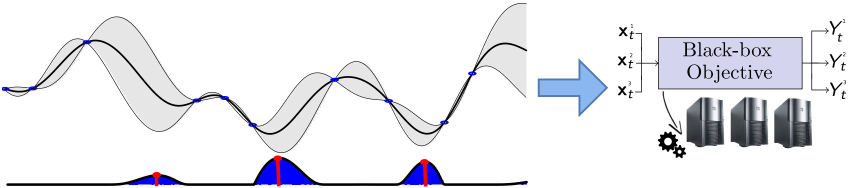


Parallel Bayesian optimization

Traditional Bayesian optimization is **sequential**!



Computing clusters allow us to collect a **batch of data** at once!



Parallel experiments should be highly informative but also diverse!

Traditional parallel BO

Parallel BO can be implemented by averaging the sequential acquisition function across data $\{\mathbf{y}_k\}_{k=1}^K$ **fantasized** at **pending** evaluation locations $\{\mathbf{x}_k\}_{k=1}^K$:

$$\alpha_{\text{parallel}}(\mathbf{x}|\mathcal{D}) = \mathbf{E}_{p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K,\mathcal{D})}\left[\alpha_{\text{sequential}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_k\}_{k=1}^K)\right] .$$

Traditional parallel BO

Parallel BO can be implemented by averaging the sequential acquisition function across data $\{\mathbf{y}_k\}_{k=1}^K$ **fantasized** at **pending** evaluation locations $\{\mathbf{x}_k\}_{k=1}^K$:

$$\alpha_{\text{parallel}}(\mathbf{x}|\mathcal{D}) = \mathbf{E}_{p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K,\mathcal{D})}\left[\alpha_{\text{sequential}}(\mathbf{x}|\mathcal{D}\cup\{\mathbf{x}_k,y_k\}_{k=1}^K)\right] .$$

Approximated by an **empirical average** across fantasies (samples) of $\{y_k\}_{k=1}^K$.

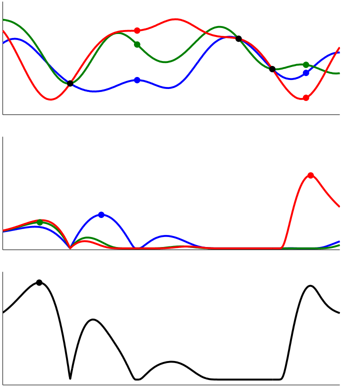
Traditional parallel BO

Figure source: Snoek et al. 2012.

Two pending evaluations, three fantasies.

Three **acquisition functions**, one per fantasy.

Average **acquisition function**.



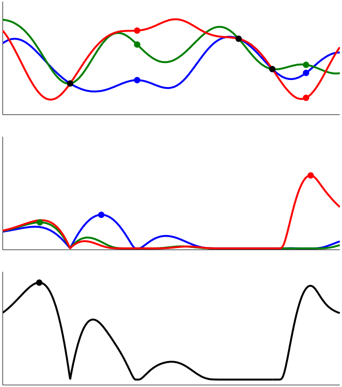
Traditional parallel BO

Figure source: Snoek et al. 2012.

Two pending evaluations, three fantasies.

Three **acquisition functions**, one per fantasy.

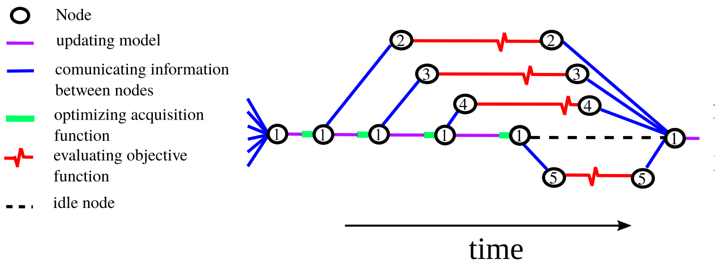
Average **acquisition function**.



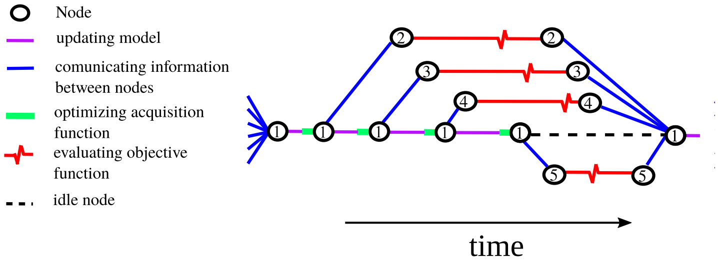
Challenges:

- Lack of scalability with **large batch sizes** and **large library sizes**.

Traditional parallel BO

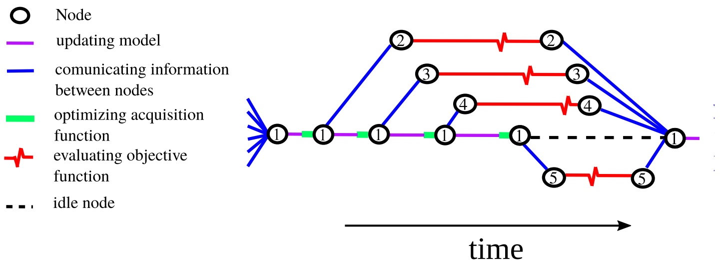


Traditional parallel BO



Updating the model and **optimizing acquisition function** is done sequentially.

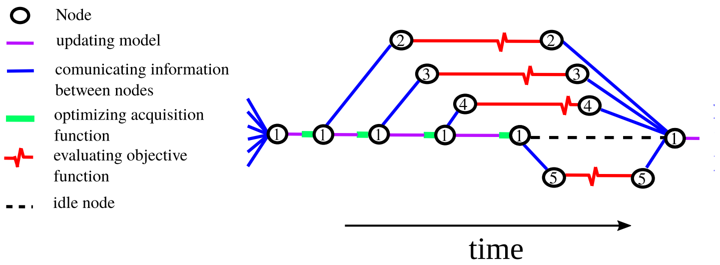
Traditional parallel BO



Updating the model and **optimizing acquisition function** is done sequentially.

Fails to exploit parallelism!

Traditional parallel BO



Updating the model and **optimizing acquisition function** is done sequentially.

Fails to exploit parallelism!

There is a need for methods that fully work in a **parallel and distributed manner**.

Thompson sampling (TS)

Sequential BO method that collects data by evaluating at $\mathbf{x} \sim p(\mathbf{x}_\star|\mathcal{D})$.

Implemented by drawing $f' \sim p(f|\mathcal{D})$ and then evaluating at $\mathbf{x} = \arg\min_{\mathbf{x}'} f'(\mathbf{x}')$.

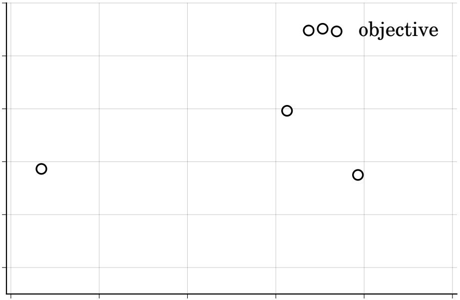
The acquisition function is a **sample** from the posterior over functions!

Thompson sampling (TS)

Sequential BO method that collects data by evaluating at $\mathbf{x} \sim p(\mathbf{x}_\star|\mathcal{D})$.

Implemented by drawing $f' \sim p(f|\mathcal{D})$ and then evaluating at $\mathbf{x} = \arg\min_{\mathbf{x}'} f'(\mathbf{x}')$.

The acquisition function is a **sample** from the posterior over functions!

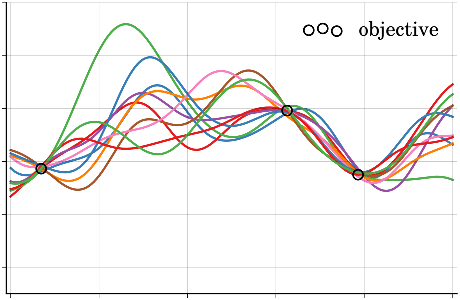


Thompson sampling (TS)

Sequential BO method that collects data by evaluating at $\mathbf{x} \sim p(\mathbf{x}_\star|\mathcal{D})$.

Implemented by drawing $f' \sim p(f|\mathcal{D})$ and then evaluating at $\mathbf{x} = \arg\min_{\mathbf{x}'} f'(\mathbf{x}')$.

The acquisition function is a **sample** from the posterior over functions!

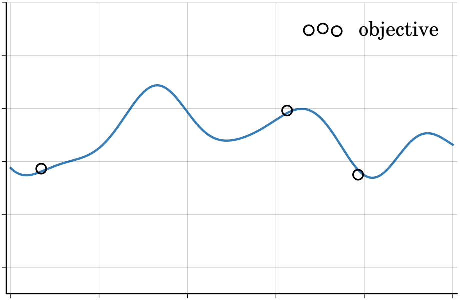


Thompson sampling (TS)

Sequential BO method that collects data by evaluating at $\mathbf{x} \sim p(\mathbf{x}_\star | \mathcal{D})$.

Implemented by drawing $f' \sim p(f | \mathcal{D})$ and then evaluating at $\mathbf{x} = \arg\min_{\mathbf{x}'} f'(\mathbf{x}')$.

The acquisition function is a **sample** from the posterior over functions!

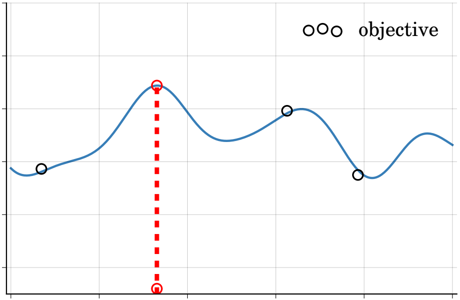


Thompson sampling (TS)

Sequential BO method that collects data by evaluating at $\mathbf{x} \sim p(\mathbf{x}_\star | \mathcal{D})$.

Implemented by drawing $f' \sim p(f | \mathcal{D})$ and then evaluating at $\mathbf{x} = \arg\min_{\mathbf{x}'} f'(\mathbf{x}')$.

The acquisition function is a **sample** from the posterior over functions!

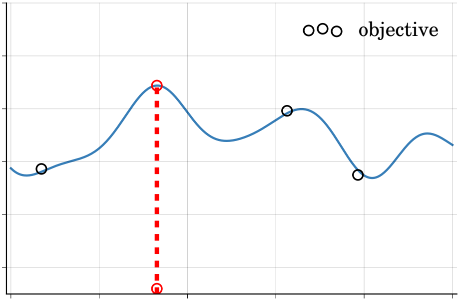


Thompson sampling (TS)

Sequential BO method that collects data by evaluating at $\mathbf{x} \sim p(\mathbf{x}_\star | \mathcal{D})$.

Implemented by drawing $f' \sim p(f|\mathcal{D})$ and then evaluating at $\mathbf{x} = \arg\min_{\mathbf{x}'} f'(\mathbf{x}')$.

The acquisition function is a **sample** from the posterior over functions!



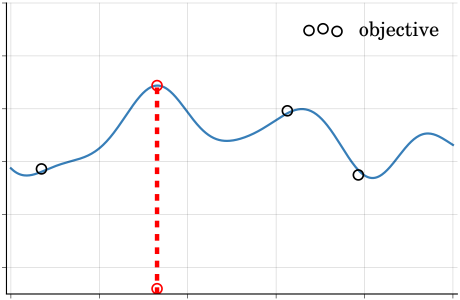
Exploitation: achieved because on average f' minimizes the prediction error.

Thompson sampling (TS)

Sequential BO method that collects data by evaluating at $\mathbf{x} \sim p(\mathbf{x}_\star|\mathcal{D})$.

Implemented by drawing $f' \sim p(f|\mathcal{D})$ and then evaluating at $\mathbf{x} = \arg\min_{\mathbf{x}'} f'(\mathbf{x}')$.

The acquisition function is a **sample** from the posterior over functions!



Exploitation: achieved because on average f' minimizes the prediction error.

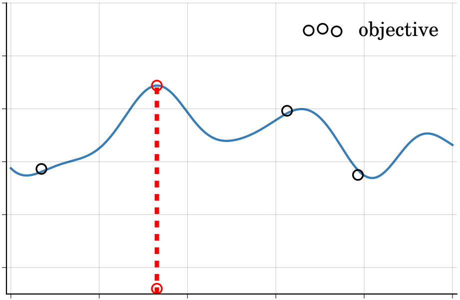
Exploration: achieved because $f' \sim p(f|\mathcal{D})$ is a random sample.

Thompson sampling (TS)

Sequential BO method that collects data by evaluating at $\mathbf{x} \sim p(\mathbf{x}_\star|\mathcal{D})$.

Implemented by drawing $f' \sim p(f|\mathcal{D})$ and then evaluating at $\mathbf{x} = \arg\min_{\mathbf{x}'} f'(\mathbf{x}')$.

The acquisition function is a **sample** from the posterior over functions!



Very simple strategy that often works well in practice.

Exploitation: achieved because on average f' minimizes the prediction error.

Exploration: achieved because $f' \sim p(f|\mathcal{D})$ is a random sample.

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\, \mathcal{D})}[y]$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

.

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\, \mathcal{D})}[y] \qquad\qquad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

.

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\, \mathcal{D})}[y] \approx \frac{1}{M}\sum_{m=1}^{M} \mathbf{E}_{p(y|\mathbf{x},\, \boldsymbol{\theta}_m)}[y]\,, \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

.

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\,\mathcal{D})}[y] \approx \frac{1}{M}\sum_{m=1}^{M} \mathbf{E}_{p(y|\mathbf{x},\,\boldsymbol{\theta}_m)}[y]\,, \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

TS uses $M = 1$ since low values of M increase variance and **exploration**.

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\,\mathcal{D})}[y] \approx \frac{1}{M}\sum_{m=1}^{M} \mathbf{E}_{p(y|\mathbf{x},\,\boldsymbol{\theta}_m)}[y]\,, \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

TS uses $M = 1$ since low values of M increase variance and **exploration**.

We can apply the **traditional parallel BO** approach to TS:

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\mathrm{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\, \mathcal{D})}[y] \approx \frac{1}{M}\sum_{m=1}^{M} \mathbf{E}_{p(y|\mathbf{x},\, \boldsymbol{\theta}_m)}[y]\,, \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

TS uses $M = 1$ since low values of M increase variance and **exploration**.

We can apply the **traditional parallel BO** approach to TS:

$$\alpha_{\text{parallel TS}}(\mathbf{x}|\mathcal{D}) = \mathbf{E}_{p(\{y_k\}_{k=1}^{K}|\{\mathbf{x}_k\}_{k=1}^{K},\mathcal{D})}\left[\alpha_{\mathrm{TS}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_k\}_{k=1}^{K})\right]$$

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\, \mathcal{D})}[y] \approx \frac{1}{M} \sum_{m=1}^{M} \mathbf{E}_{p(y|\mathbf{x},\, \boldsymbol{\theta}_m)}[y]\,, \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

TS uses $M = 1$ since low values of M increase variance and **exploration**.

We can apply the **traditional parallel BO** approach to TS:

$$\alpha_{\text{parallel TS}}(\mathbf{x}|\mathcal{D}) = \mathbf{E}_{p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K, \mathcal{D})}\left[\alpha_{\text{TS}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_k\}_{k=1}^K)\right]$$
$$\approx \frac{1}{M} \sum_{m=1}^{M} \alpha_{\text{TS}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_{k,m}\}_{k=1}^K)$$

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\,\mathcal{D})}[y] \approx \frac{1}{M}\sum_{m=1}^{M} \mathbf{E}_{p(y|\mathbf{x},\,\boldsymbol{\theta}_m)}[y]\,, \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

TS uses $M = 1$ since low values of M increase variance and **exploration**.

We can apply the **traditional parallel BO** approach to TS:

$$\begin{aligned} \alpha_{\text{parallel TS}}(\mathbf{x}|\mathcal{D}) &= \mathbf{E}_{p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K,\mathcal{D})}\left[\alpha_{\text{TS}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_k\}_{k=1}^K)\right] \\ &\approx \frac{1}{M}\sum_{m=1}^{M} \alpha_{\text{TS}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_{k,m}\}_{k=1}^K) \end{aligned}$$

where $\{y_{k,m}\}_{k=1}^K \sim p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K, \mathcal{D})$,

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\,\mathcal{D})}[y] \approx \frac{1}{M}\sum_{m=1}^{M} \mathbf{E}_{p(y|\mathbf{x},\,\boldsymbol{\theta}_m)}[y]\,, \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

TS uses $M = 1$ since low values of M increase variance and **exploration**.

We can apply the **traditional parallel BO** approach to TS:

$$\alpha_{\text{parallel TS}}(\mathbf{x}|\mathcal{D}) = \mathbf{E}_{p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K,\mathcal{D})}\left[\alpha_{\text{TS}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_k\}_{k=1}^K)\right]$$
$$\approx \frac{1}{M}\sum_{m=1}^{M} \alpha_{\text{TS}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_{k,m}\}_{k=1}^K)$$

where $\{y_{k,m}\}_{k=1}^K \sim p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K, \mathcal{D})$, and as before, $M = 1$.

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\, \mathcal{D})}[y] \approx \frac{1}{M}\sum_{m=1}^{M} \mathbf{E}_{p(y|\mathbf{x},\, \boldsymbol{\theta}_m)}[y]\,, \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

TS uses $M = 1$ since low values of M increase variance and **exploration**.

We can apply the **traditional parallel BO** approach to TS:

$$\begin{aligned}\alpha_{\text{parallel TS}}(\mathbf{x}|\mathcal{D}) &= \mathbf{E}_{p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K,\mathcal{D})}\left[\alpha_{\text{TS}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_k\}_{k=1}^K)\right] \\ &\approx \frac{1}{M}\sum_{m=1}^{M} \alpha_{\text{TS}}(\mathbf{x}|\mathcal{D} \cup \{\mathbf{x}_k, y_{k,m}\}_{k=1}^K) = \alpha_{\text{TS}}(\mathbf{x}|\mathcal{D})\,,\end{aligned}$$

where $\{y_{k,m}\}_{k=1}^K \sim p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K, \mathcal{D})$, and as before, $M = 1$.

TS as utility maximization and parallel TS

The utility function used by TS is $U(y|\mathbf{x}, \mathcal{D}) = y$. TS aims to optimize

$$\alpha_{\text{TS}}(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\,\mathcal{D})}[y] \approx \frac{1}{M}\sum_{m=1}^{M}\mathbf{E}_{p(y|\mathbf{x},\,\boldsymbol{\theta}_m)}[y]\,, \quad \boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\mathcal{D})\,,$$

where $p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\,d\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ are the model parameters.

TS uses $M = 1$ since low values of M increase variance and **exploration**.

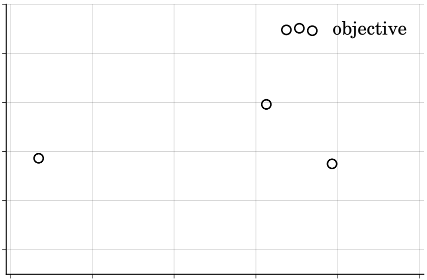
We can apply the **traditional parallel BO** approach to TS:

$$\begin{aligned}\alpha_{\text{parallel TS}}(\mathbf{x}|\mathcal{D}) &= \mathbf{E}_{p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K,\mathcal{D})}\left[\alpha_{\text{TS}}(\mathbf{x}|\mathcal{D}\cup\{\mathbf{x}_k,y_k\}_{k=1}^K)\right]\\ &\approx \frac{1}{M}\sum_{m=1}^{M}\alpha_{\text{TS}}(\mathbf{x}|\mathcal{D}\cup\{\mathbf{x}_k,y_{k,m}\}_{k=1}^K) = \alpha_{\text{TS}}(\mathbf{x}|\mathcal{D})\,,\end{aligned}$$

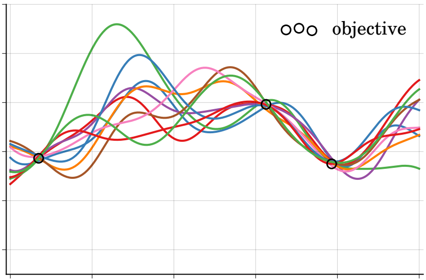
where $\{y_{k,m}\}_{k=1}^K \sim p(\{y_k\}_{k=1}^K|\{\mathbf{x}_k\}_{k=1}^K, \mathcal{D})$, and as before, $M = 1$.

Our parallel TS is equivalent to running sequential TS multiple times!

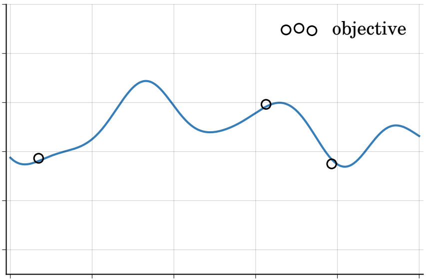
Example



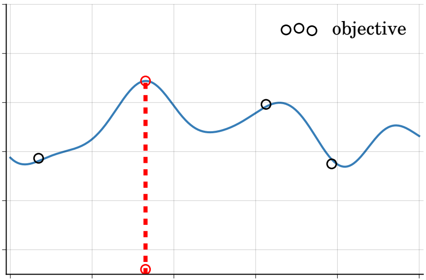
Example



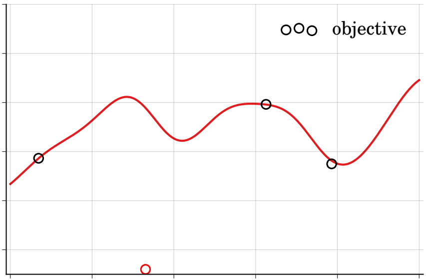
Example



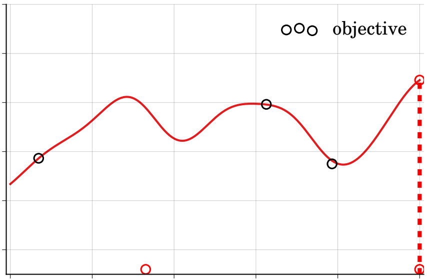
Example



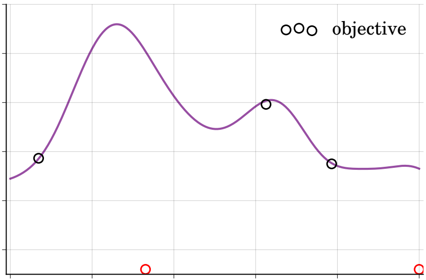
Example



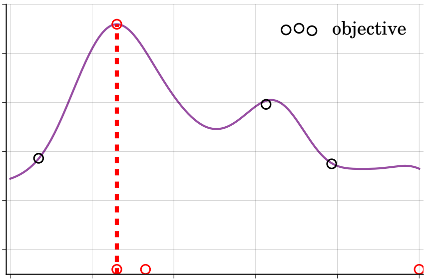
Example



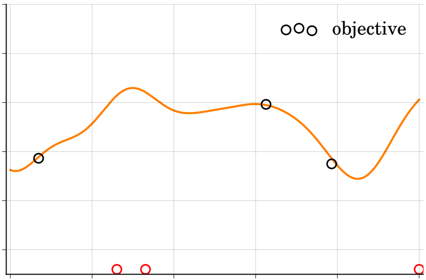
Example



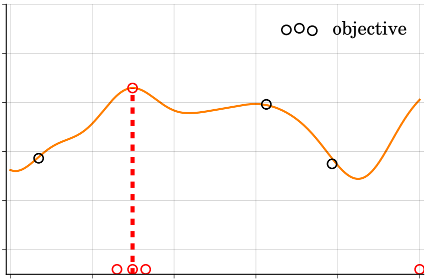
Example



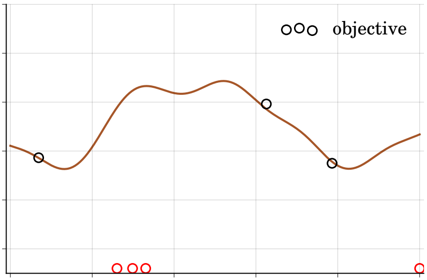
Example



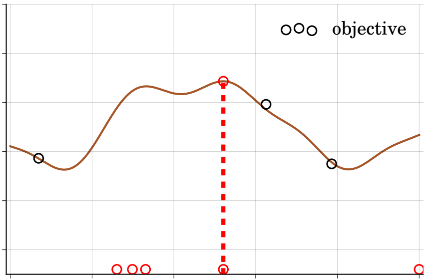
Example



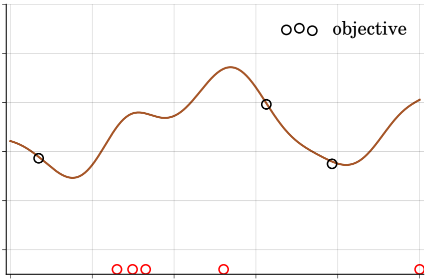
Example



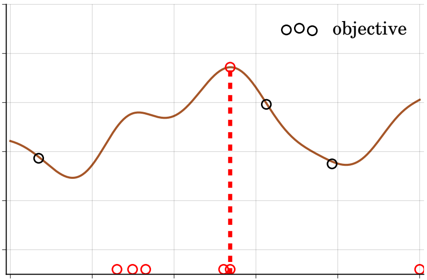
Example



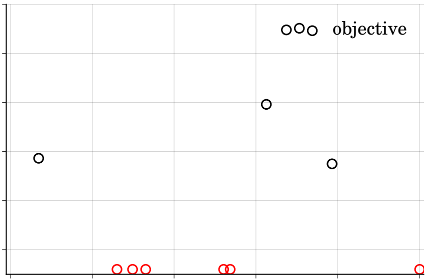
Example



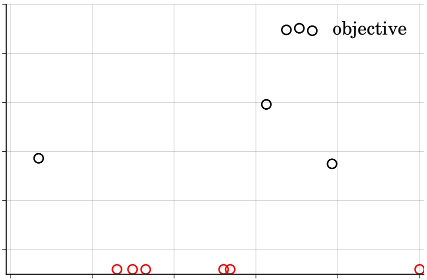
Example



Example

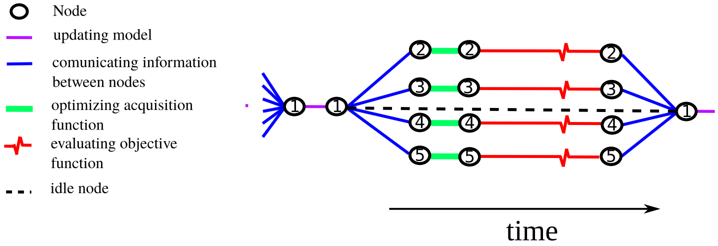


Example

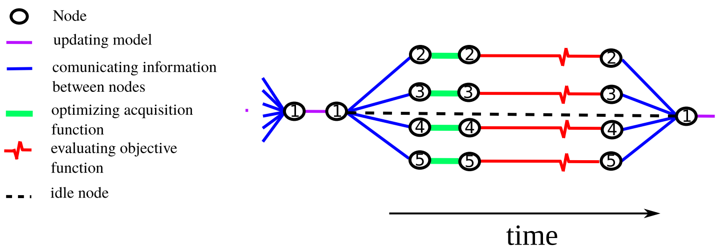


Each optimization problem can be done independently in a **different computer**.

Parallel Thompson sampling



Parallel Thompson sampling



Works in a fully **parallel and distributed manner**.

Thompson sampling with Gaussian processes (GPs)

GPs are a **non-parametric models**, so sampling the model parameters $\boldsymbol{\theta}$ and optimizing $\mathbf{E}_{p(y|\mathbf{x},\boldsymbol{\theta})}[y]$ is not possible.

We approximate the objective function f as $f(\mathbf{x}) \approx \boldsymbol{\Phi}(\mathbf{x})\boldsymbol{\theta}$ with random features

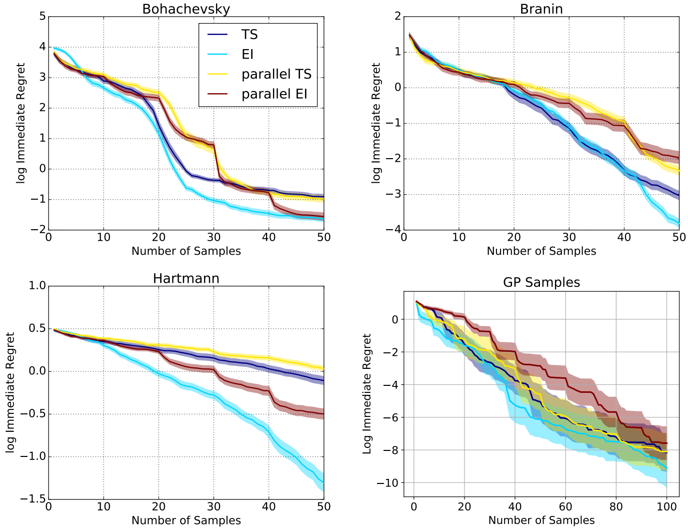
$$\boldsymbol{\Phi}(\mathbf{x}) = C\cos(\mathbf{W}^{\mathsf{T}}\mathbf{x} + \mathbf{b})\,, \tag{1}$$

where $\mathbf{W}, \mathbf{b} \sim p(\mathbf{W}, \mathbf{b})$, a distribution specified by the GP covariance function.

The prior for $\boldsymbol{\theta}$ is $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The resulting **Bayesian linear regression model** is a parametric approximation to the original GP model.

Results with Gaussian process based models

Batch size: 10



Results with Bayesian neural networks

Data sets:

- CEP: Harvard Clean Energy Project data, 2.3M molecules.
- One-dose: percentage cell growth relative to control, 27,000 molecules.
- Malaria: drug concentration giving half max response, 19,000 molecules.

Batch sizes: 500 (CEP) and 200 (Malaria and One-dose).

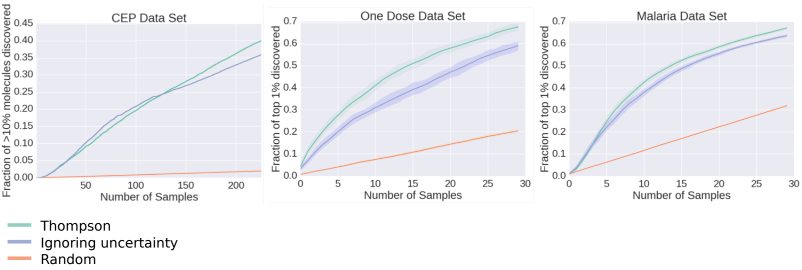
Results with Bayesian neural networks

Data sets:

- CEP: Harvard Clean Energy Project data, 2.3M molecules.
- One-dose: percentage cell growth relative to control, 27,000 molecules.
- Malaria: drug concentration giving half max response, 19,000 molecules.

Batch sizes: 500 (CEP) and 200 (Malaria and One-dose).

Fraction of top 10% (CEP) or 1% (Malaria and One-dose) molecules found:



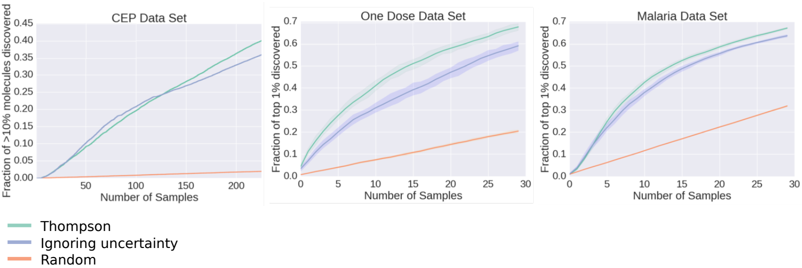
Results with Bayesian neural networks

Data sets:

- CEP: Harvard Clean Energy Project data, 2.3M molecules.
- One-dose: percentage cell growth relative to control, 27,000 molecules.
- Malaria: drug concentration giving half max response, 19,000 molecules.

Batch sizes: 500 (CEP) and 200 (Malaria and One-dose).

Fraction of top 10% (CEP) or 1% (Malaria and One-dose) molecules found:



BO gives 20× gains over random in CEP. Using uncertainty always helps.

Comparison with ϵ-greedy sampling

Table: Average rank and standard errors obtained by each method.

Method	**Rank**
$\epsilon = 0.01$	3.42±0.28
$\epsilon = 0.025$	3.02±0.25
$\epsilon = 0.05$	2.86±0.23
$\epsilon = 0.075$	3.20±0.26
Thompson	**2.51±0.20**

Take home message

Parallel Thompson sampling...

1. Batch BO method that runs in a **parallel and distributed manner**.
2. Can handle large batch sizes and large molecule libraries.
3. Comparable to non-scalable approaches in small problems with GPs.
4. Outperforms other alternative scalable approaches in large scale settings.

Thanks!