fastppm: fast tumor phylogeny regression via tree-structured dual dynamic programming

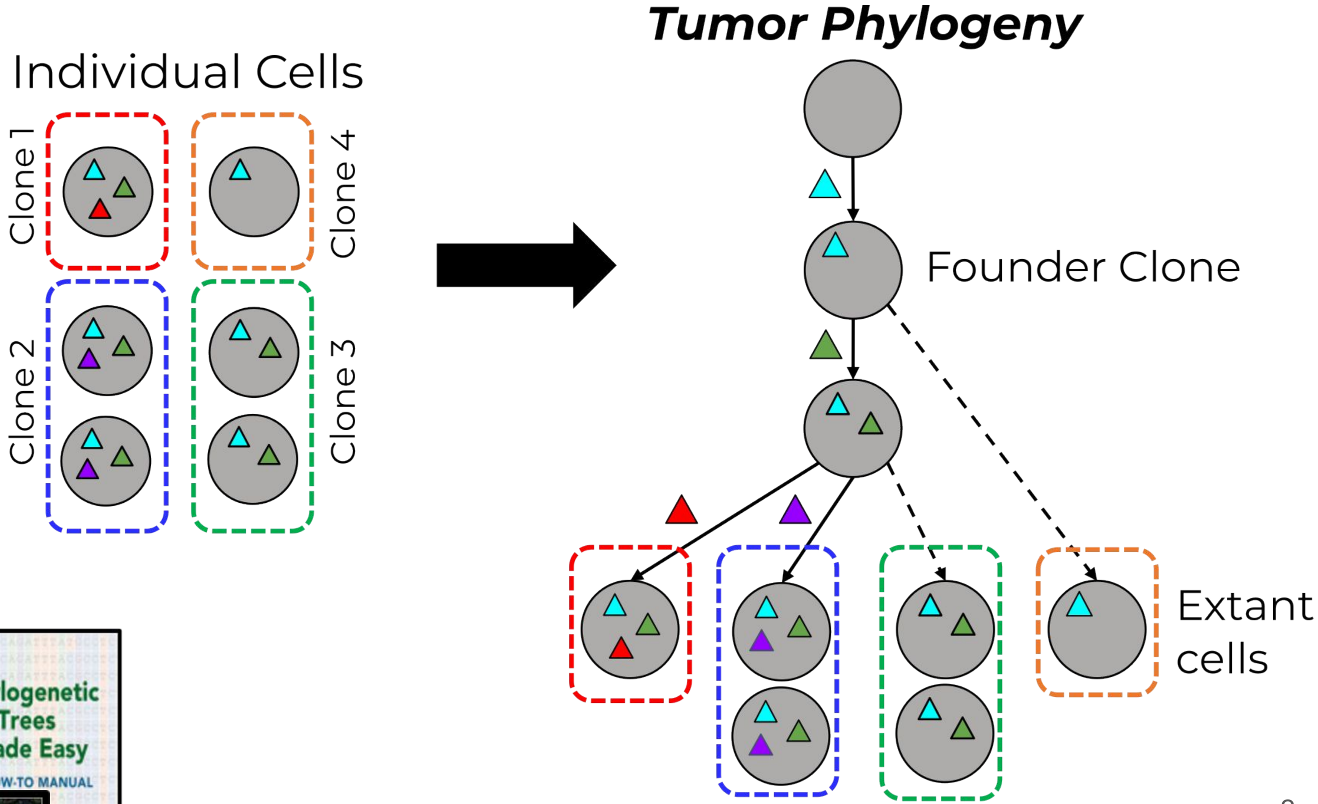
Henri Schmidt*[,1], Yuanyuan Qi*[,2], Ben Raphael[1], and Mohammed El-Kebir[2]

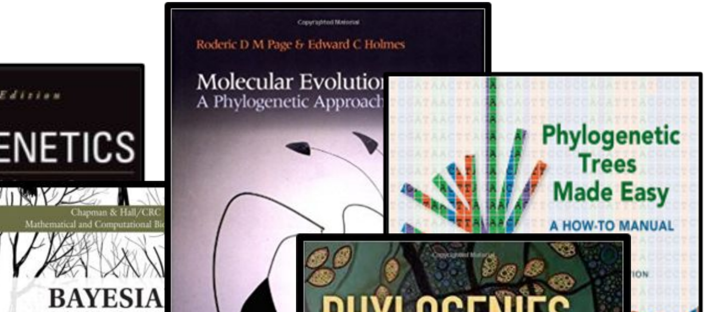
1 PRINCETON UNIVERSITY

2 UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

* Denotes equal contribution.

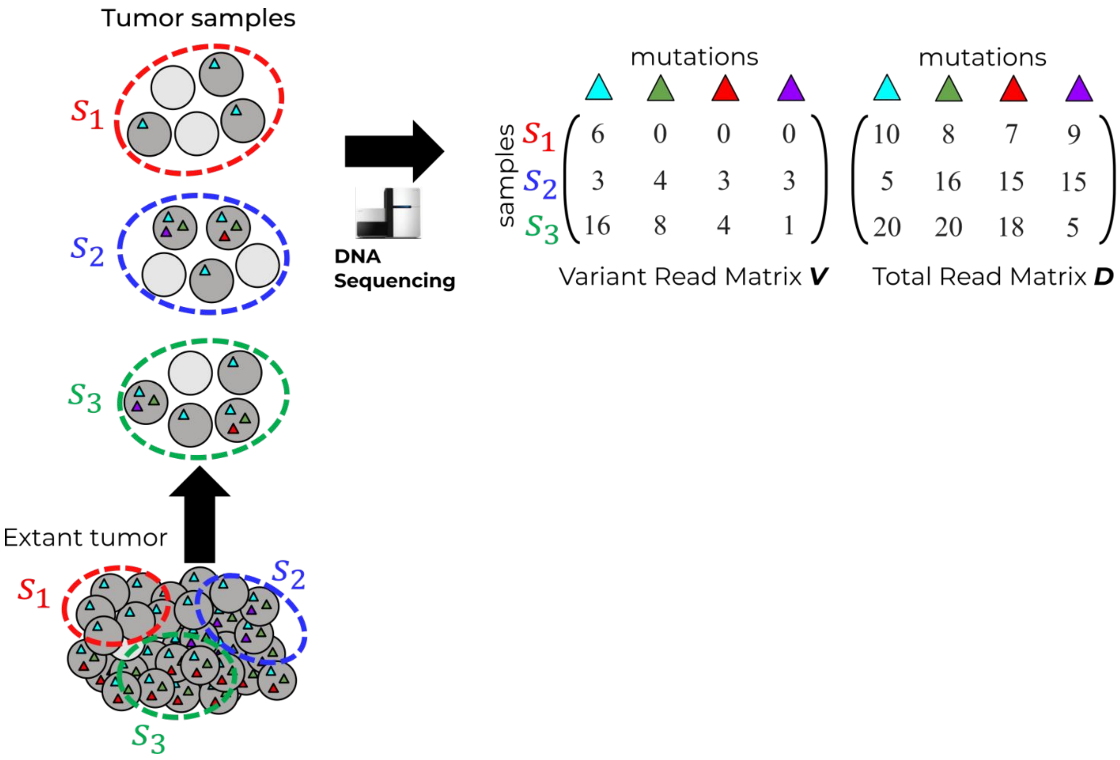
Reconstructing the evolutionary history of a tumor is a challenging and important open question



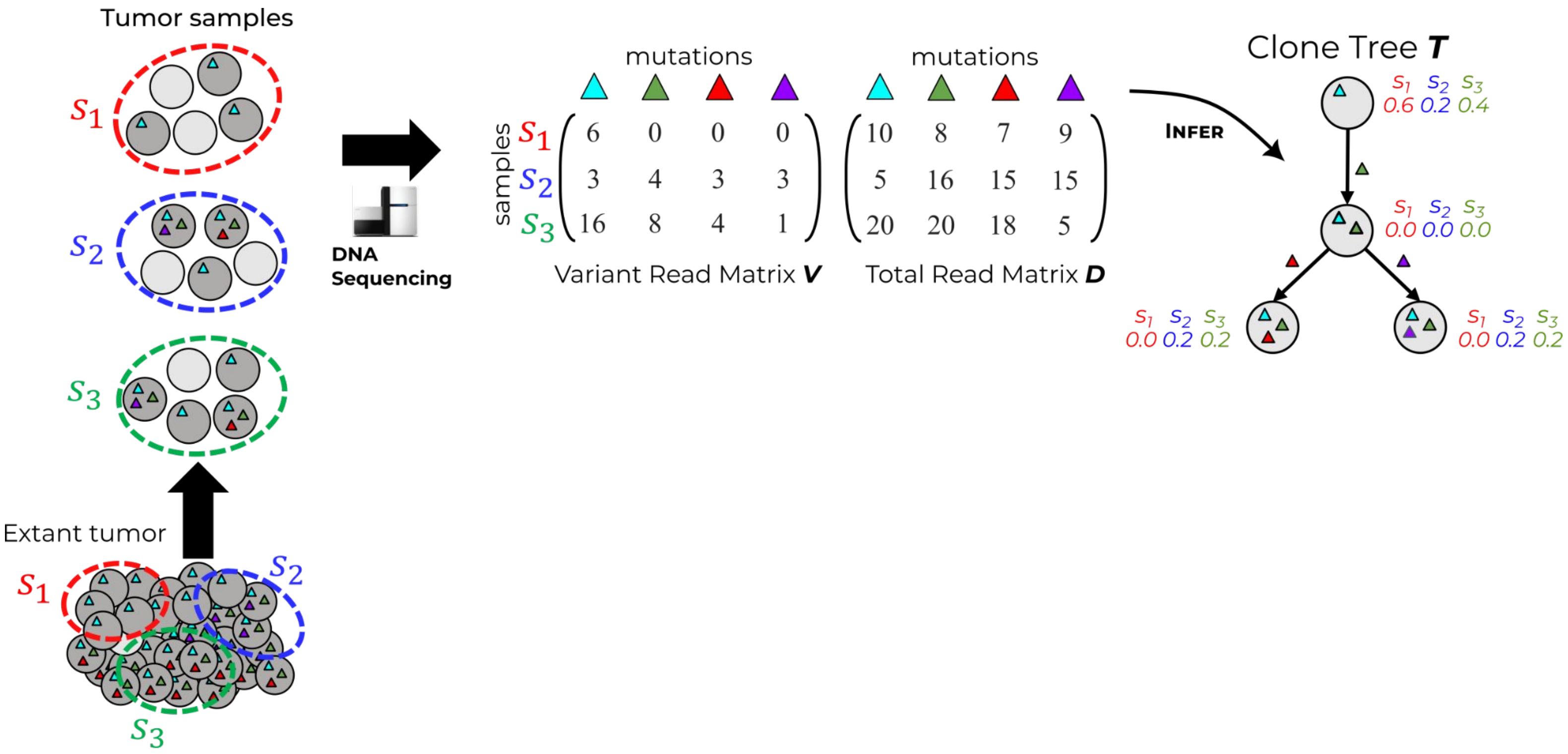


Clone = group of cells with identical genotypes

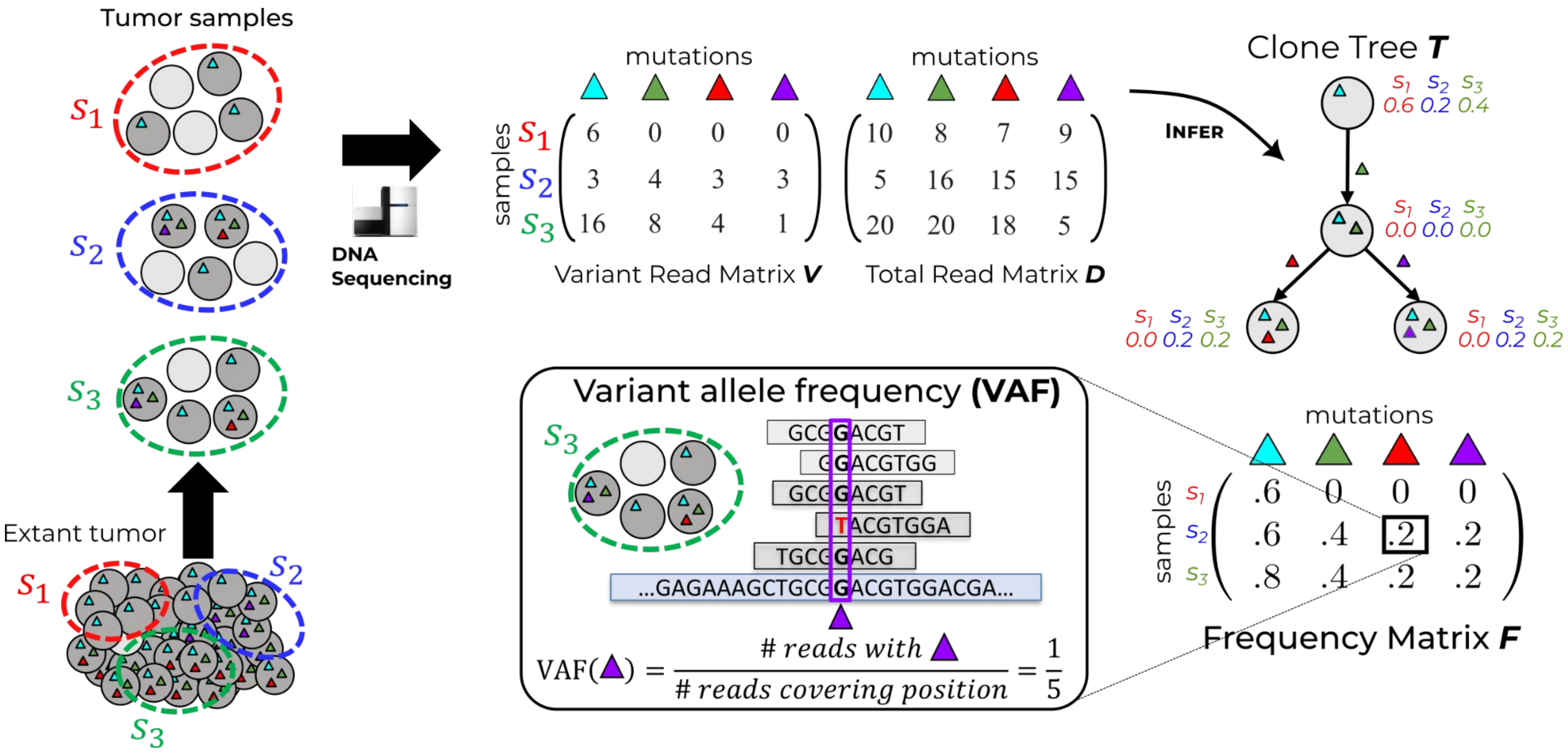
Bulk DNA sequencing yields a mixture of cells, requiring simultaneous inference of clones and their proportions



Bulk DNA sequencing yields a mixture of cells, requiring simultaneous inference of clones and their proportions



Bulk DNA sequencing yields a mixture of cells, requiring simultaneous inference of clones and their proportions

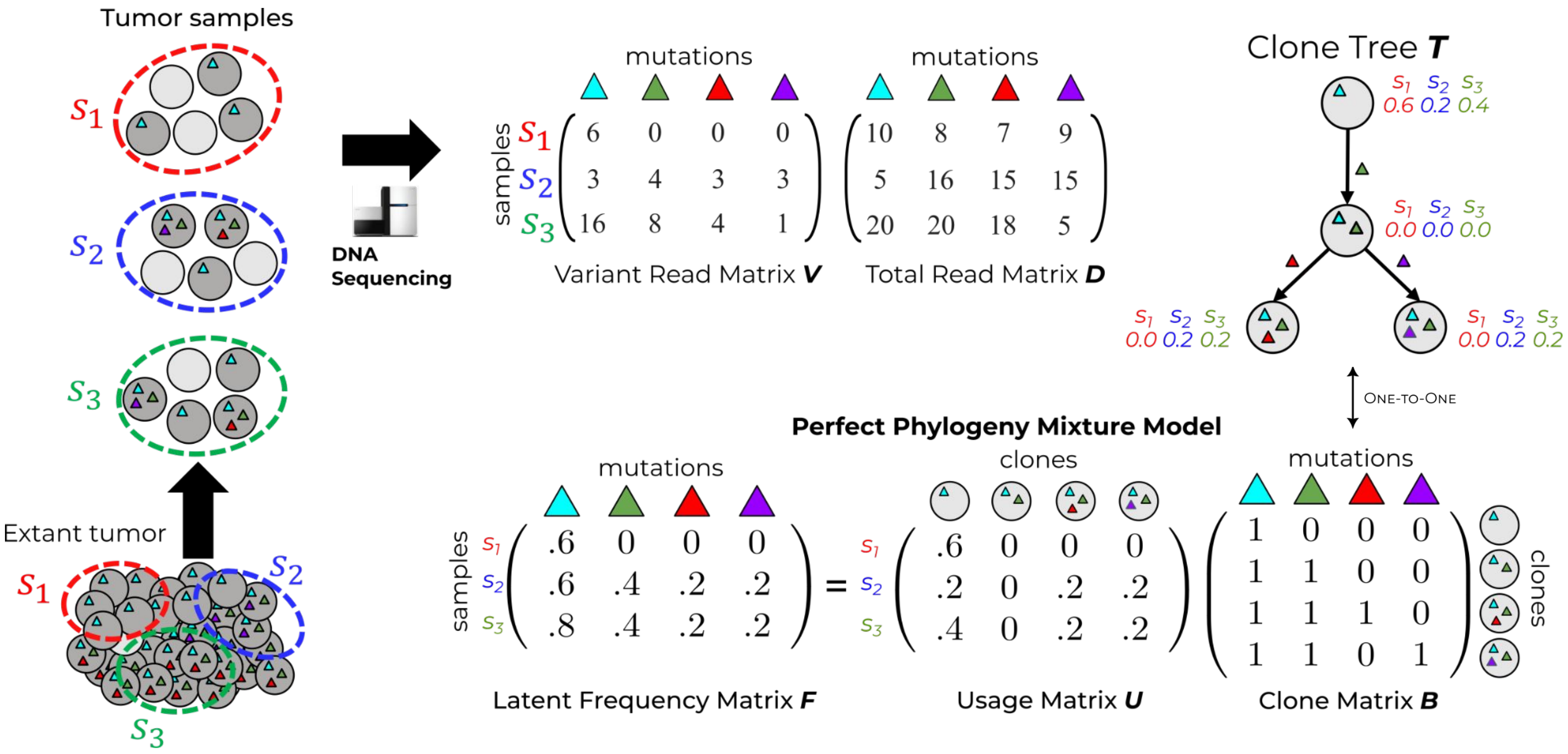


$$V = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{pmatrix} 6 & 0 & 0 & 0 \\ 3 & 4 & 3 & 3 \\ 16 & 8 & 4 & 1 \end{pmatrix} \qquad D = \begin{pmatrix} 10 & 8 & 7 & 9 \\ 5 & 16 & 15 & 15 \\ 20 & 20 & 18 & 5 \end{pmatrix}$$

$$\text{VAF}(\blacktriangle) = \frac{\#\ \textit{reads with}\ \blacktriangle}{\#\ \textit{reads covering position}} = \frac{1}{5}$$

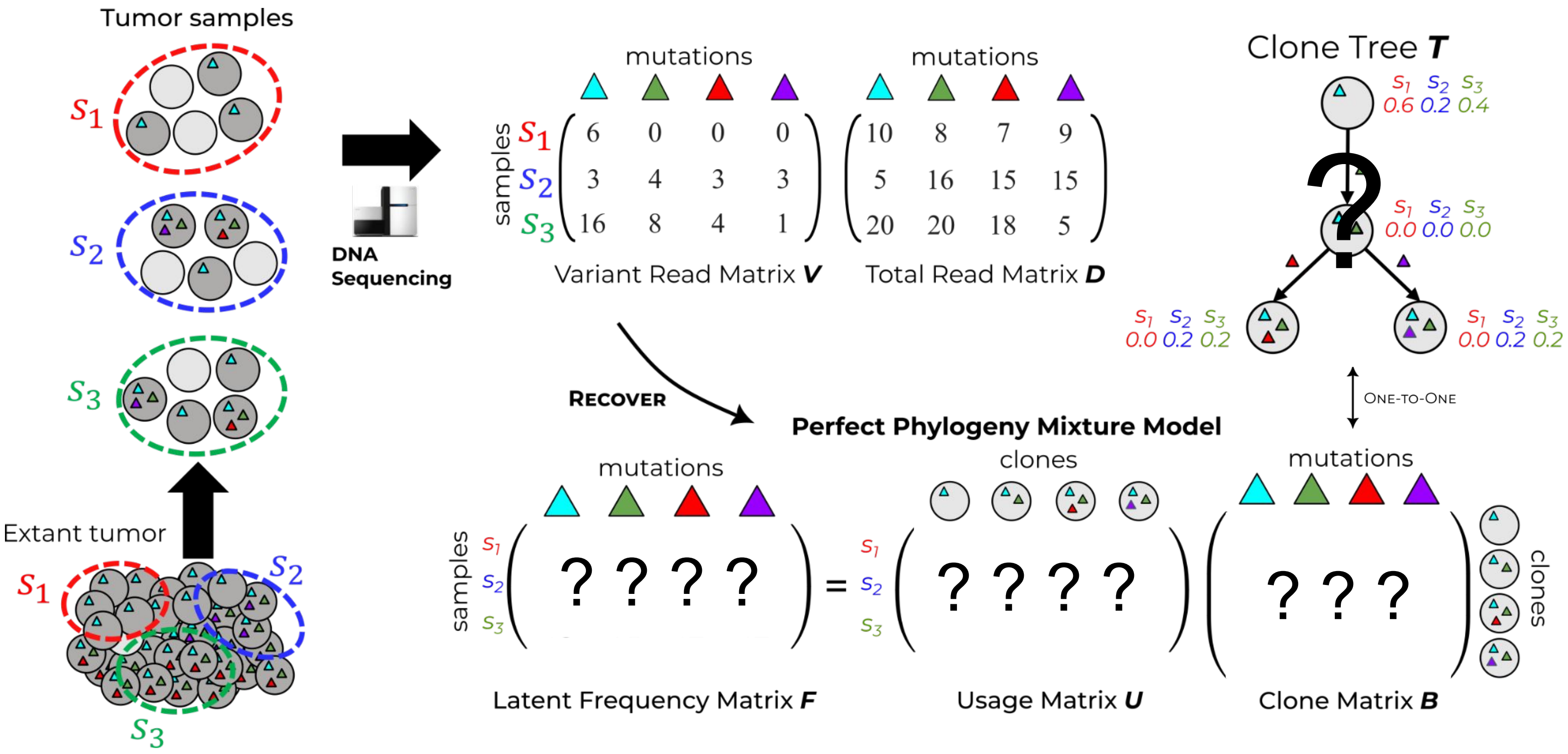
$$F = \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} \begin{pmatrix} .6 & 0 & 0 & 0 \\ .6 & .4 & \boxed{.2} & .2 \\ .8 & .4 & .2 & .2 \end{pmatrix}$$

Bulk DNA sequencing yields a mixture of cells, requiring simultaneous inference of clones and their proportions



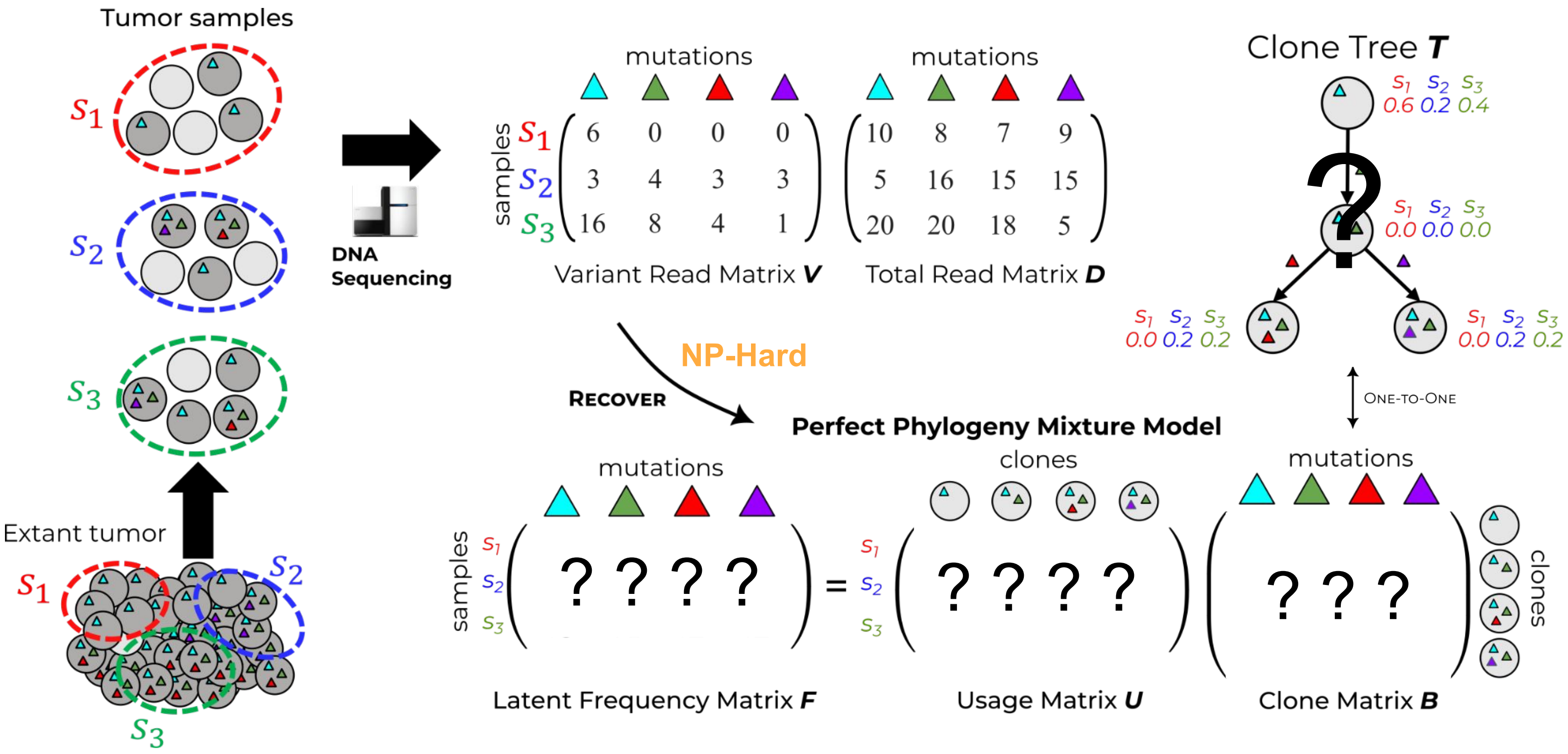
* This model is implicit or explicit in: **PhyloSub** *(Jiao et al.. 2014)*, **PhyloWGS** *(Deshwar et al., 2015)*, **CITUP** *(Malikic et al., 2015)*, **LICHeE** *(Popic et al.,. 2015)*, **AncesTree** *(El-Kebir et al.2015)*, **Canopy** *(Jiang et al.,2016)*, **PairTree** *(Wintersinger et al., 2022)*, ***Orchard*** *(Kulman et al., 2023)*, ***fastBE*** *(Schmidt et al. 2024)*, …

Bulk DNA sequencing yields a mixture of cells, requiring simultaneous inference of clones and their proportions



* This model is implicit or explicit in: **PhyloSub** *(Jiao et al.. 2014)*, **PhyloWGS** *(Deshwar et al., 2015)*, **CITUP** *(Malikic et al., 2015)*, **LICHeE** *(Popic et al.,. 2015)*, **AncesTree** *(El-Kebir et al.2015)*, **Canopy** *(Jiang et al.,2016)*, **PairTree** *(Wintersinger et al., 2022)*, ***Orchard*** *(Kulman et al., 2023)*, ***fastBE*** *(Schmidt et al. 2024), …*

Bulk DNA sequencing yields a mixture of cells, requiring simultaneous inference of clones and their proportions



* This model is implicit or explicit in: **PhyloSub** *(Jiao et al.. 2014)*, **PhyloWGS** *(Deshwar et al., 2015)*, **CITUP** *(Malikic et al., 2015)*, **LICHeE** *(Popic et al.,. 2015)*, **AncesTree** *(El-Kebir et al.2015)*, **Canopy** *(Jiang et al.,2016)*, **PairTree** *(Wintersinger et al., 2022)*, ***Orchard*** *(Kulman et al., 2023)*, ***fastBE*** *(Schmidt et al. 2024)*, …

The perfect phylogeny regression problem

mutations

$$\text{samples}\quad \begin{array}{c|c} & \begin{array}{cccc} \color{cyan}\blacktriangle & \color{green}\blacktriangle & \color{red}\blacktriangle & \color{purple}\blacktriangle \end{array} \\ \begin{array}{c} \color{red}{s_1} \\ \color{blue}{s_2} \\ \color{green}{s_3} \end{array} & \begin{pmatrix} 6 & 0 & 0 & 0 \\ 3 & 4 & 3 & 3 \\ 16 & 8 & 4 & 1 \end{pmatrix} \end{array}$$

Variant Read Matrix ***V***

$$\text{samples}\quad \begin{array}{c|c} & \begin{array}{cccc} \color{cyan}\blacktriangle & \color{green}\blacktriangle & \color{red}\blacktriangle & \color{purple}\blacktriangle \end{array} \\ \begin{array}{c} \color{red}{s_1} \\ \color{blue}{s_2} \\ \color{green}{s_3} \end{array} & \begin{pmatrix} 10 & 8 & 7 & 9 \\ 5 & 16 & 15 & 15 \\ 20 & 20 & 18 & 5 \end{pmatrix} \end{array}$$

Total Read Matrix ***D***

The perfect phylogeny regression problem

mutations

$$\begin{array}{c|cccc} & \triangle & \triangle & \triangle & \triangle \\ \hline s_1 & 6 & 0 & 0 & 0 \\ s_2 & 3 & 4 & 3 & 3 \\ s_3 & 16 & 8 & 4 & 1 \end{array}$$

samples

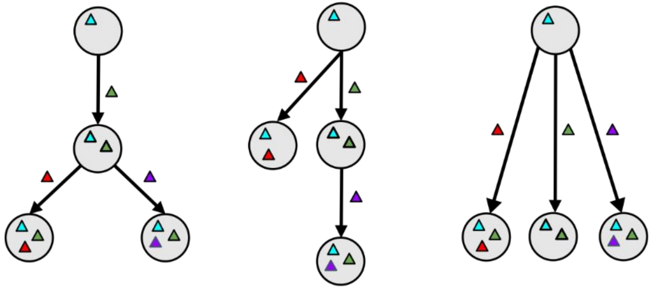
Variant Read Matrix $\boldsymbol{V}$

$$\begin{array}{c|cccc} & \triangle & \triangle & \triangle & \triangle \\ \hline s_1 & 10 & 8 & 7 & 9 \\ s_2 & 5 & 16 & 15 & 15 \\ s_3 & 20 & 20 & 18 & 5 \end{array}$$

samples

Total Read Matrix $\boldsymbol{D}$

Candidate Tree Set $\mathscr{T} = \{T_1, T_2, T_3, \ldots\}$



Clone Tree $\boldsymbol{T_1}$ Clone Tree $\boldsymbol{T_2}$ Clone Tree $\boldsymbol{T_3}$

The perfect phylogeny regression problem

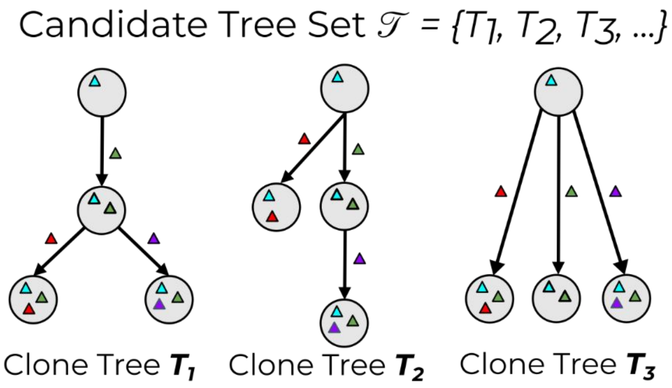
mutations

$$\text{samples}\;\begin{array}{c} S_1 \\ S_2 \\ S_3 \end{array}\begin{pmatrix} 6 & 0 & 0 & 0 \\ 3 & 4 & 3 & 3 \\ 16 & 8 & 4 & 1 \end{pmatrix}$$

Variant Read Matrix ***V***

$$\text{samples}\;\begin{array}{c} S_1 \\ S_2 \\ S_3 \end{array}\begin{pmatrix} 10 & 8 & 7 & 9 \\ 5 & 16 & 15 & 15 \\ 20 & 20 & 18 & 5 \end{pmatrix}$$

Total Read Matrix ***D***



Score & rank trees

SCORES

L($\mathbf{F_1}$ | **V**, **D**) = 0.25

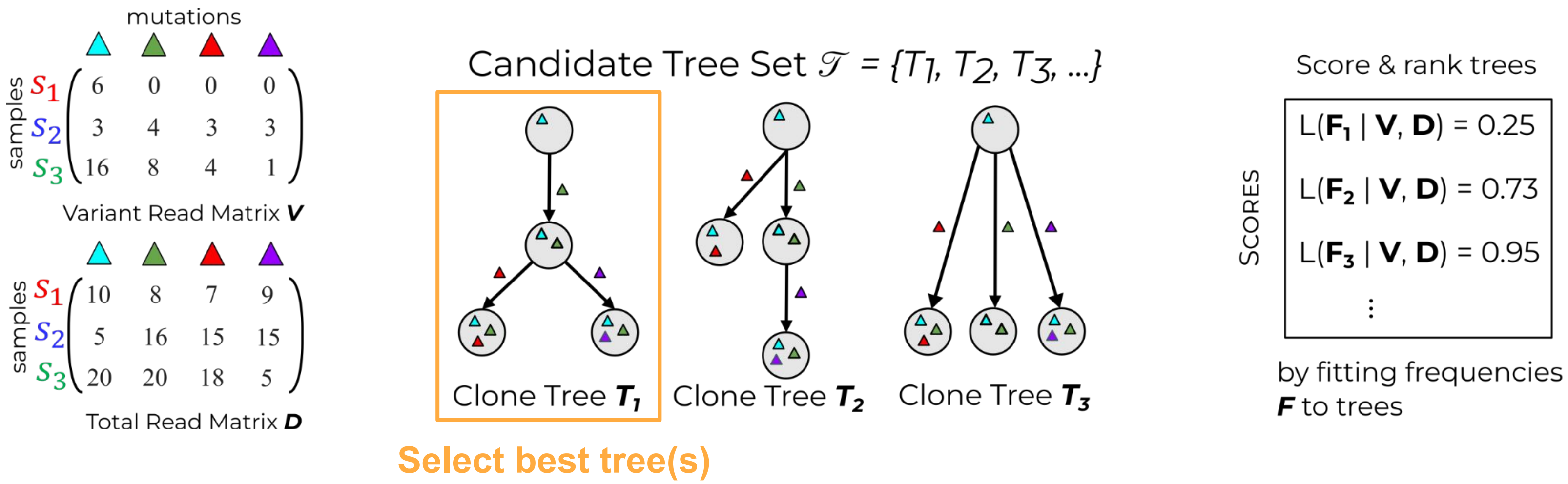
L($\mathbf{F_2}$ | **V**, **D**) = 0.73

L($\mathbf{F_3}$ | **V**, **D**) = 0.95

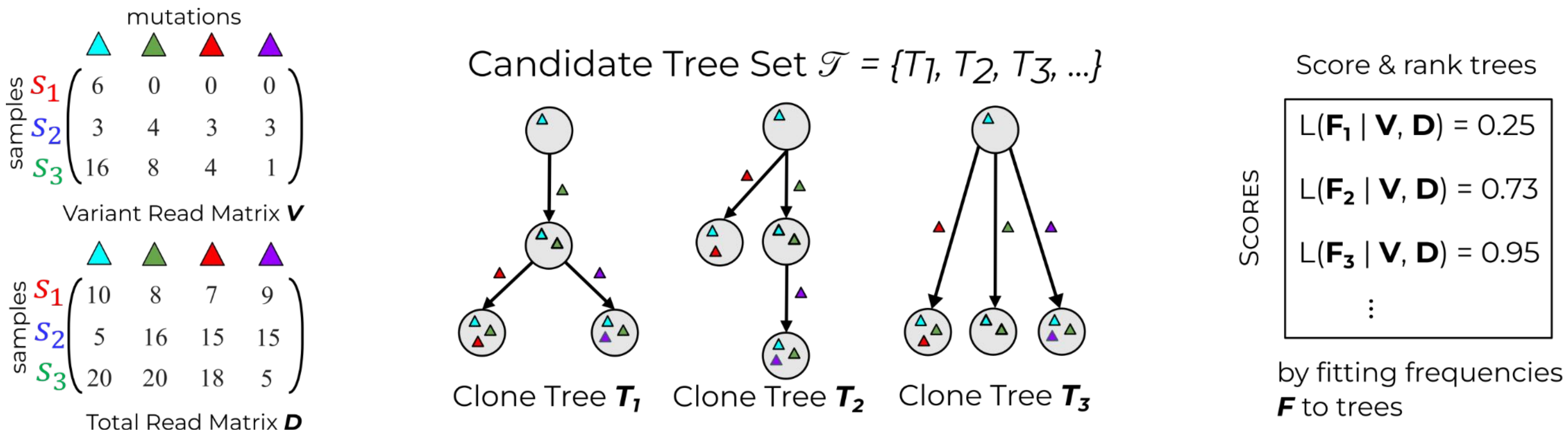
⋮

by fitting frequencies ***F*** to trees

The perfect phylogeny regression problem



The perfect phylogeny regression problem



Trees are scored by **repeatedly*** solving the ***perfect phylogeny regression problem***:

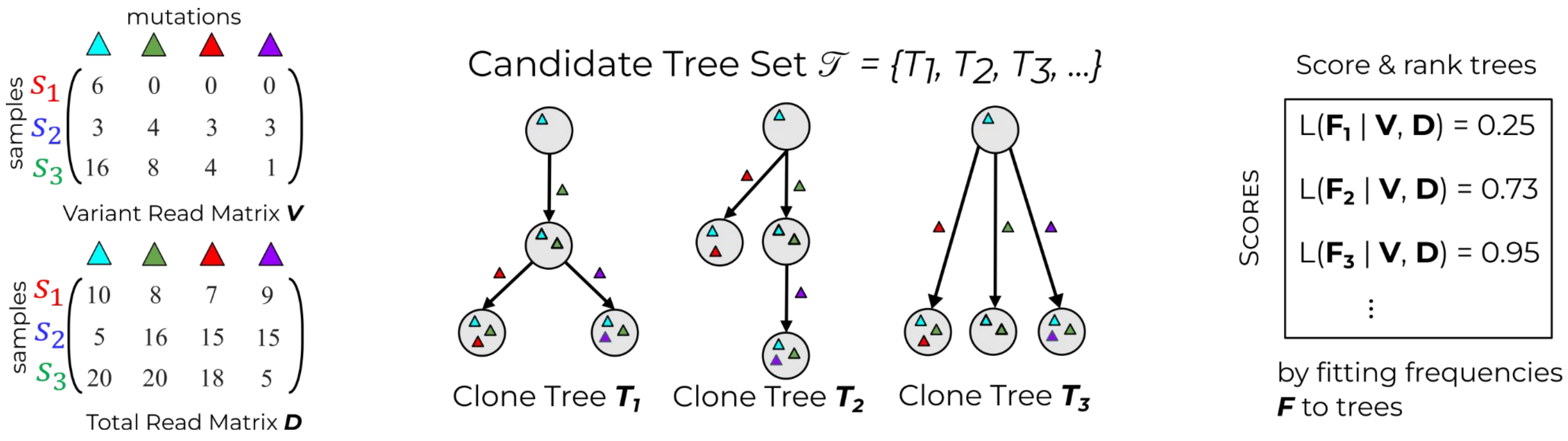
B is fixed

(PPR)

$$\min_{\mathbf{F},\mathbf{U}}\{L(\mathbf{F} \mid \mathbf{V},\mathbf{D}) : \mathbf{F} = \mathbf{U}\boxed{\mathbf{B}}, \mathbf{U} \geq 0, \mathbf{U}\mathbb{1} \leq 1\}$$

* This approach is taken in CITUP (Malikic et al., 2015), LICHeE (Popic et al.,. 2015), AncesTree (El-Kebir et al.2015), PairTree (Wintersinger et al., 2022), Orchard (Kulman et al., 2023), fastBE (Schmidt et al. 2024), and Sapling (Qi et al. 2024)

The perfect phylogeny regression problem



Trees are scored by **repeatedly*** solving the ***perfect phylogeny regression problem***:

B is fixed

(PPR)

$$\min_{\mathbf{F},\mathbf{U}}\{L(\mathbf{F} \mid \mathbf{V},\mathbf{D}) : \mathbf{F} = \mathbf{U}\boxed{\mathbf{B}}, \mathbf{U} \geq 0, \mathbf{U}\mathbb{1} \leq 1\}$$

Solving the PPR problem is the **key computational bottleneck** in phylogeny inference algorithms.

* This approach is taken in CITUP (Malikic et al., 2015), LICHeE (Popic et al.,. 2015), AncesTree (El-Kebir et al.2015), PairTree (Wintersinger et al., 2022), Orchard (Kulman et al., 2023), fastBE (Schmidt et al. 2024), and Sapling (Qi et al. 2024)

The perfect phylogeny regression problem

For example, when a least-squares loss is used

$$L(\mathbf{F} \mid \mathbf{V}, \mathbf{D}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(f_{ij} - \frac{v_{ij}}{d_{ij}} \right)^2$$

e.g. as in **CITUP** *(Malikic et al., 2015)*, **PairTree** *(Wintersinger et al., 2022), and* ***Orchard*** *(Kulman et al., 2023)*

the PPR problem is solved in polynomial time using either quadratic programming or specialized solvers.

The perfect phylogeny regression problem

For example, when a least-squares loss is used

$$L(\mathbf{F} \mid \mathbf{V}, \mathbf{D}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(f_{ij} - \frac{v_{ij}}{d_{ij}} \right)^2$$

e.g. as in **CITUP** *(Malikic et al., 2015)*, **PairTree** *(Wintersinger et al., 2022), and* ***Orchard*** *(Kulman et al., 2023)*

the PPR problem is solved in polynomial time using either quadratic programming or specialized solvers.

Pitfall #1: Does not model read counts

The perfect phylogeny regression problem

For example, when a least-squares loss is used

$$L(\mathbf{F} \mid \mathbf{V}, \mathbf{D}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(f_{ij} - \frac{v_{ij}}{d_{ij}} \right)^2$$

e.g. as in **CITUP** *(Malikic et al., 2015)*, **PairTree** *(Wintersinger et al., 2022), and* ***Orchard*** *(Kulman et al., 2023)*

the PPR problem is solved in polynomial time using either quadratic programming or specialized solvers.

Pitfall #1: Does not model read counts

For example, when a binomial loss is used (i.e. model $v_{ij} \sim \text{Binomial}(f_{ij}, d_{ij})$)

$$L(\mathbf{F} \mid \mathbf{V}, \mathbf{D}) = -\sum_{i=1}^{m} \sum_{j=1}^{n} (v_{ij} \log(f_{ij}) + (d_{ij} - v_{ij}) \log(1 - f_{ij}))$$

e.g. as in **Sapling** (Qi et al. 2024) or **PairTree** *(Wintersinger et al., 2022)*

the PPR problem is solved using **general purpose** convex optimization software.

The perfect phylogeny regression problem

For example, when a least-squares loss is used

$$L(\mathbf{F} \mid \mathbf{V}, \mathbf{D}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(f_{ij} - \frac{v_{ij}}{d_{ij}} \right)^2$$

e.g. as in **CITUP** *(Malikic et al., 2015)*, **PairTree** *(Wintersinger et al., 2022), and* ***Orchard*** *(Kulman et al., 2023)*

the PPR problem is solved in polynomial time using either quadratic programming or specialized solvers.

Pitfall #1: Does not model read counts

For example, when a binomial loss is used (i.e. model $v_{ij} \sim \text{Binomial}(f_{ij}, d_{ij})$)

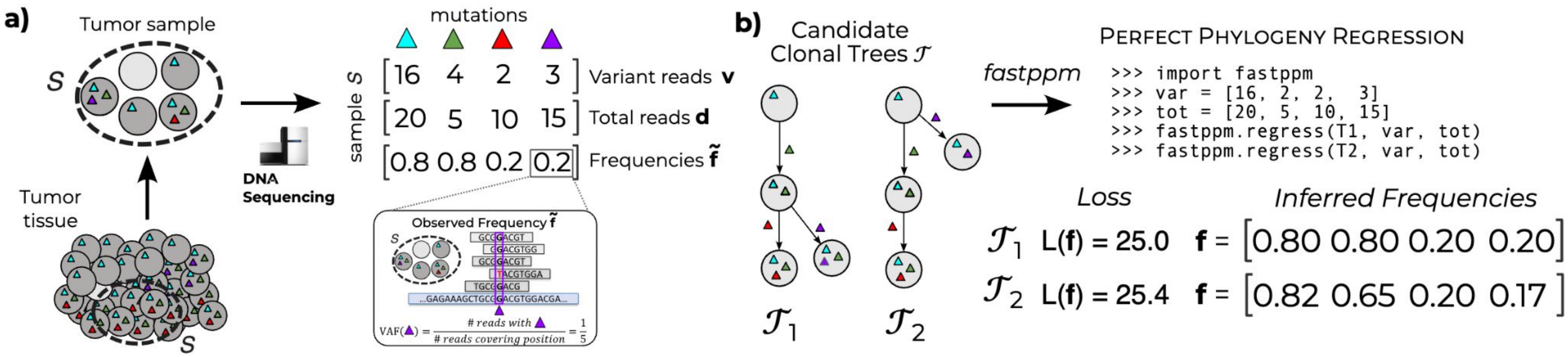
$$L(\mathbf{F} \mid \mathbf{V}, \mathbf{D}) = -\sum_{i=1}^{m} \sum_{j=1}^{n} \left(v_{ij} \log(f_{ij}) + (d_{ij} - v_{ij}) \log(1 - f_{ij}) \right)$$

e.g. as in **Sapling** (Qi et al. 2024) or **PairTree** *(Wintersinger et al., 2022)*

the PPR problem is solved using **general purpose** convex optimization software.

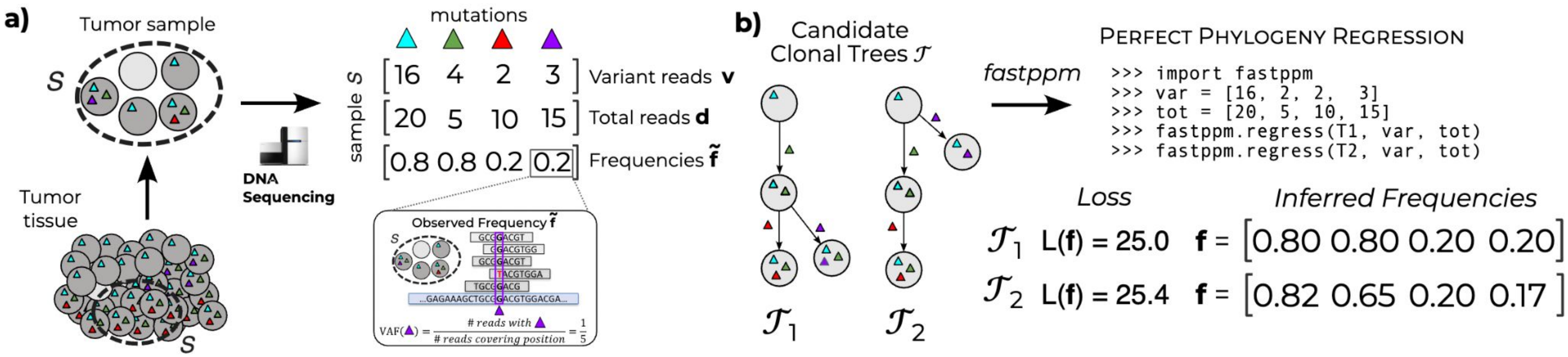
Pitfall #2: General purpose solvers are slow

Contributions



We introduce a new approach to the Perfect Phylogeny Regression problem, ***fastppm***, using ***tree structured dual dynamic programming (TSDDP)***.

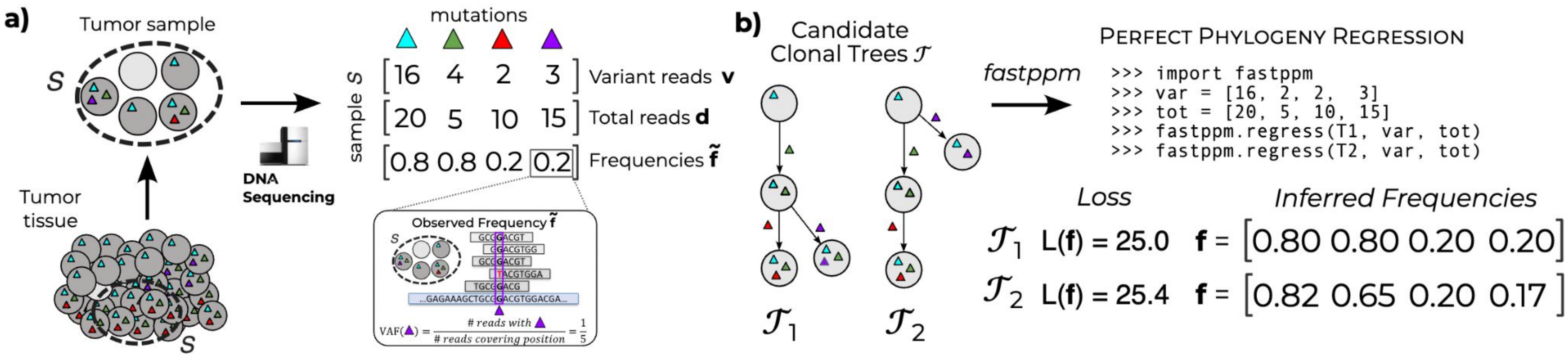
Contributions



We introduce a new approach to the Perfect Phylogeny Regression problem, ***fastppm***, using ***tree structured dual dynamic programming (TSDDP)***. Compared to existing methods:

1. *fastppm* provides **asymptotic** and **empirical** (50-100x speed-up) speedups for the most commonly used L_2 and L_1 loss functions.

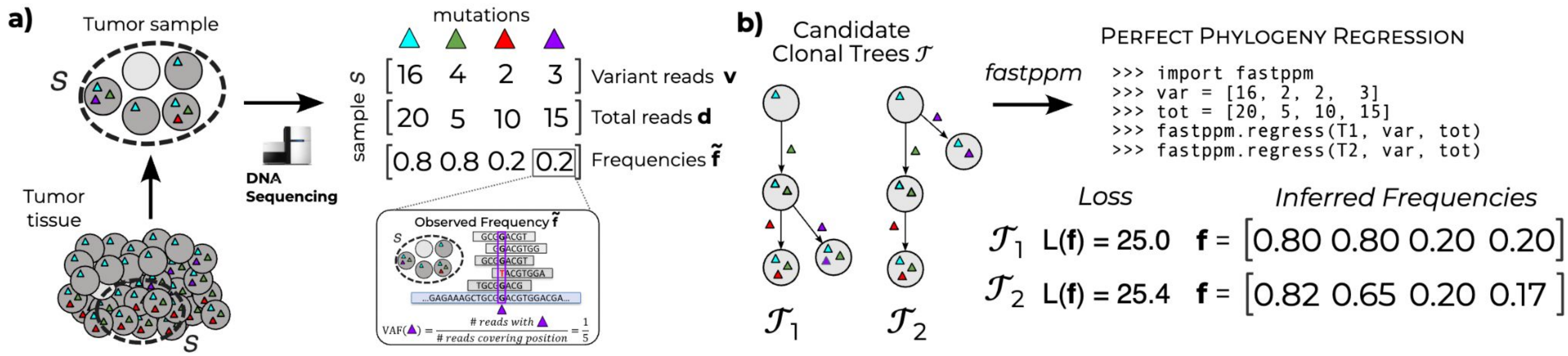
Contributions



We introduce a new approach to the Perfect Phylogeny Regression problem, ***fastppm***, using ***tree structured dual dynamic programming (TSDDP)***. Compared to existing methods:

1. *fastppm* provides **asymptotic** and **empirical** (50-100x speed-up) speedups for the most commonly used L_2 and L_1 loss functions.
2. *fastppm* is able to model arbitrary, **convex loss functions**, while maintaining its performance.

Contributions



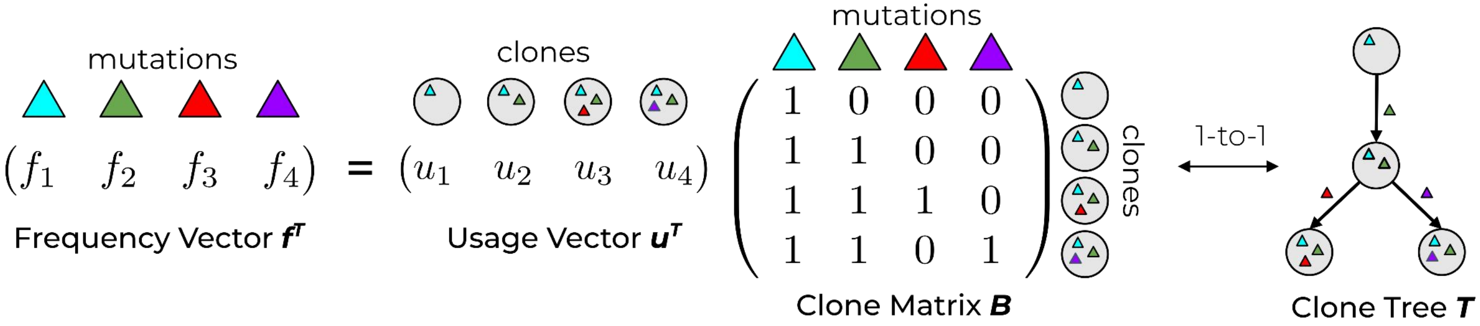
We introduce a new approach to the Perfect Phylogeny Regression problem, ***fastppm***, using ***tree structured dual dynamic programming (TSDDP)***. Compared to existing methods:

1. *fastppm* provides **asymptotic** and **empirical** (50-100x speed-up) speedups for the most commonly used L_2 and L_1 loss functions.
2. *fastppm* is able to model arbitrary, **convex loss functions**, while maintaining its performance.

On simulated data, replacing existing solvers with *fastppm* yields up to **400x** speed-ups and enables fast + accurate phylogenetic inference from **low-coverage** bulk DNA sequencing data.

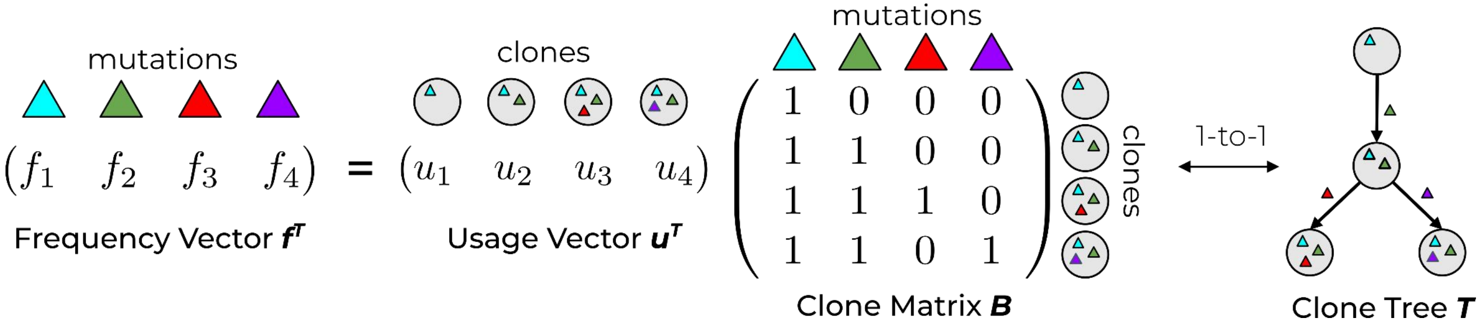
Tree structured dual dynamic programming (TSDDP)

For simplicity, we study the PPR problem in the case of a single sample:



Tree structured dual dynamic programming (TSDDP)

For simplicity, we study the PPR problem in the case of a single sample:

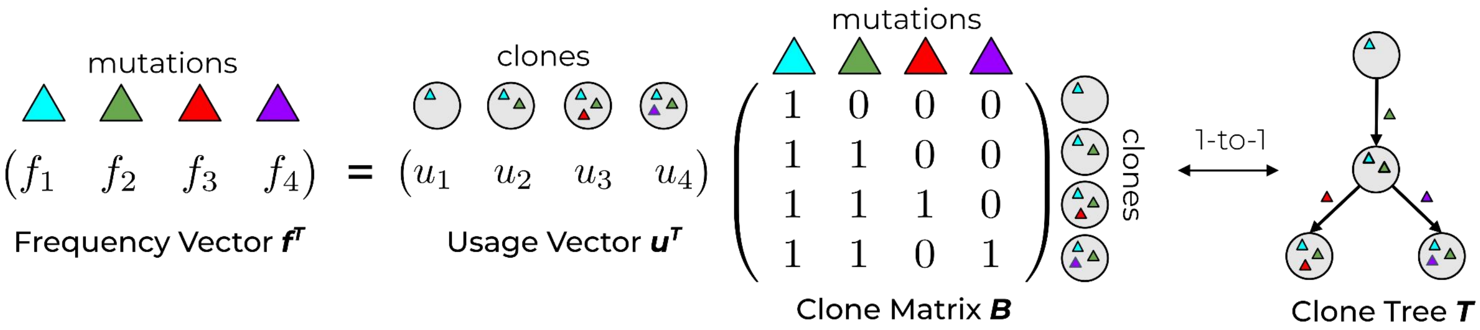


Then, our optimization problem is

(PPR) $$\min_{\mathbf{f},\mathbf{u}\in\mathbb{R}^n} \{\textstyle\sum_{i=1}^{n} L_i(f_i) : \mathbf{f}^T = \mathbf{u}^T\mathbf{B}, \mathbf{u} \geq 0, \mathbf{u}^T\mathbb{1} \leq 1\}.$$

Tree structured dual dynamic programming (TSDDP)

For simplicity, we study the PPR problem in the case of a single sample:



Then, our optimization problem is

(PPR) $$\min_{\mathbf{f},\mathbf{u}\in\mathbb{R}^n} \{\textstyle\sum_{i=1}^{n} L_i(f_i) : \mathbf{f}^T = \mathbf{u}^T\mathbf{B}, \mathbf{u} \geq 0, \mathbf{u}^T\mathbb{1} \leq 1\}.$$

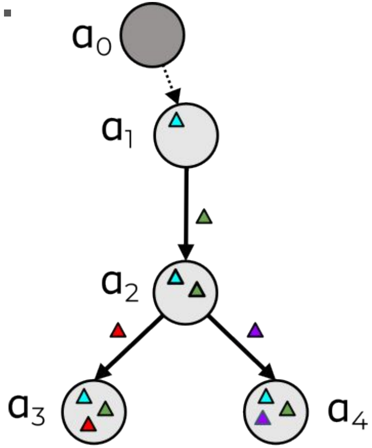
In TSDDP, we first construct the **dual** optimization problem

(Dual-PPR) $$\max_{\boldsymbol{\alpha}\in\mathbb{R}^{n+1}} \{-\alpha_0 + \sum_{i=1}^{n} h_i(\alpha_i - \alpha_{\pi(i)}) : \boldsymbol{\alpha} \geq 0\},$$

where $\pi(i)$ is the parent of vertex i in T and h_i is conjugate to L_i.

Tree structured dual dynamic programming (TSDDP)

Example.



Clone Tree ***T***

VARIABLES:

a_0, a_1, a_2, a_3, a_4 ON VERTICES OF **T**

OBJECTIVE:

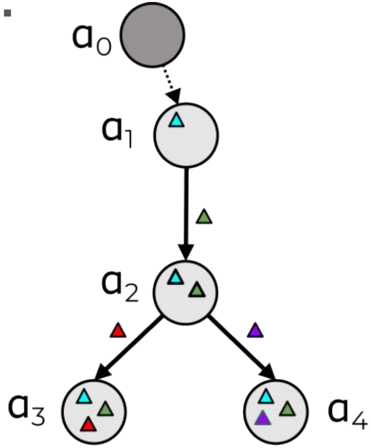
$$h_1(a_1-a_0) + h_2(a_2-a_1) + h_3(a_3-a_2) + h_4(a_4-a_2)$$

CONSTRAINTS:

$$a_0 \geq 0,\ a_1 \geq 0,\ a_2 \geq 0,\ a_3 \geq 0,\ a_4 \geq 0$$

Tree structured dual dynamic programming (TSDDP)

Example.



Clone Tree ***T***

VARIABLES:

a_0, a_1, a_2, a_3, a_4 ON VERTICES OF **T**

OBJECTIVE:

$h_1(a_1\text{-}a_0) + h_2(a_2\text{-}a_1) + h_3(a_3\text{-}a_2) + h_4(a_4\text{-}a_2)$

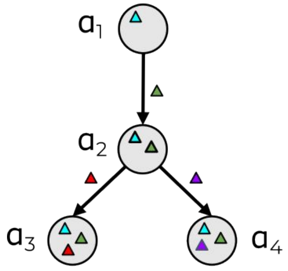
CONSTRAINTS:

$a_0 \geq 0,\ a_1 \geq 0,\ a_2 \geq 0,\ a_3 \geq 0,\ a_4 \geq 0$

Key Idea: Solve the **dual** problem with a **bottom-up dynamic programming algorithm** over the clone tree **T**.

Tree structured dual dynamic programming (TSDDP)

Key Idea: Solve the **dual** problem with a **bottom-up dynamic programming algorithm** over the clone tree **T**.



Clone Tree ***T***

VARIABLES:

a_0, a_1, a_2, a_3, a_4 ON VERTICES OF **T**

OBJECTIVE:

$h_1(a_1\text{-}a_0) + h_2(a_2\text{-}a_1) + h_3(a_3\text{-}a_2) + h_4(a_4\text{-}a_2)$

CONSTRAINTS:

$a_0 \geq 0, a_1 \geq 0, a_2 \geq 0, a_3 \geq 0, a_4 \geq 0$

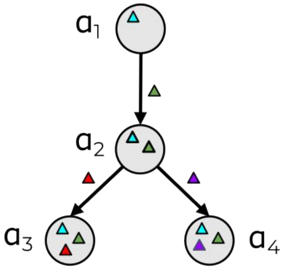
Specifically, we define

$$J_i(\gamma) \triangleq \max_{\alpha \geq 0} \{\textstyle\sum_{j \in D(i)} h_j(\alpha_j - \alpha_{\pi(j)}) : \alpha_{\pi(i)} = \gamma\}$$

which is the ***optimal solution to the dual problem for the subtree rooted at vertex i, provided the dual variable of the parent of vertex i takes value γ.***

Tree structured dual dynamic programming (TSDDP)

Key Idea: Solve the **dual** problem with a **bottom-up dynamic programming algorithm** over the clone tree **T**.



Clone Tree ***T***

VARIABLES:

$\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4$ ON VERTICES OF **T**

OBJECTIVE:

$h_1(\alpha_1-\alpha_0) + h_2(\alpha_2-\alpha_1) + h_3(\alpha_3-\alpha_2) + h_4(\alpha_4-\alpha_2)$

CONSTRAINTS:

$\alpha_0 \geq 0, \alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_3 \geq 0, \alpha_4 \geq 0$

Specifically, we define

$$J_i(\gamma) \triangleq \max_{\alpha \geq 0}\{\textstyle\sum_{j \in D(i)} h_j(\alpha_j - \alpha_{\pi(j)}) : \alpha_{\pi(i)} = \gamma\}$$

which is the ***optimal solution to the dual problem for the subtree rooted at vertex i, provided the dual variable of the parent of vertex i takes value γ.*** Then, this function satisfies the recurrence relation

$$J_i(\gamma) = \max_{\alpha_i \geq 0}\{h_i(\alpha_i - \gamma) + \textstyle\sum_{j \in \delta(i)} J_j(\alpha_i)\}$$

(Recurrence Relation)

and TSDDP then computes the functions J_i in a **bottom-up** fashion.

Tree structured dual dynamic programming (TSDDP)

For the *weighted* least squares loss, we solve the PPR problem in $\mathcal{O}(n^{3/2}\log(\log(n)))$ time* over classes of random trees:

Efficient Projection onto the Perfect Phylogeny Model

Bei Jia* jiabe@bc.edu
Surjyendu Ray raysc@bc.edu
Sam Safavi safavisa@bc.edu
José Bento jose.bento@bc.edu
Boston College

Best known: $\mathcal{O}(n^2)$
Our result: $\mathcal{O}(n^{3/2}\log(\log(n)))$

* *n* is the number of clones in the clone tree ***T.***

Tree structured dual dynamic programming (TSDDP)

For the *weighted* least squares loss, we solve the PPR problem in $\mathcal{O}(n^{3/2}log(log(n)))$ time* over classes of random trees:

Efficient Projection onto the Perfect Phylogeny Model

Bei Jia* jiabe@bc.edu
Surjyendu Ray raysc@bc.edu
Sam Safavi safavisa@bc.edu
José Bento jose.bento@bc.edu
Boston College

Best known: $\mathcal{O}(n^2)$
Our result: $\mathcal{O}(n^{3/2}\log(\log(n)))$

For the *piecewise linear loss* with *k* pieces, we solve the PPR problem in $\mathcal{O}(nlog^2(nk))$ time deterministically:

RESEARCH ARTICLE

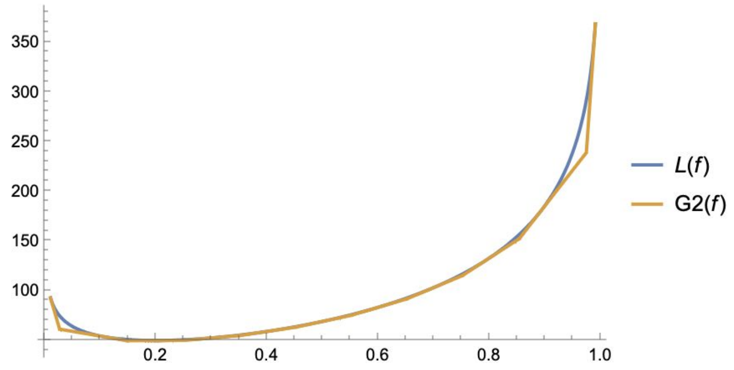
A regression based approach to phylogenetic reconstruction from multi-sample bulk DNA sequencing of tumors

Best known: $\mathcal{O}(n^{3/2})$
Our result: $\mathcal{O}(n\log^2(nk))$

* *n* is the number of clones in the clone tree ***T.***

Extensions to arbitrary, convex loss functions

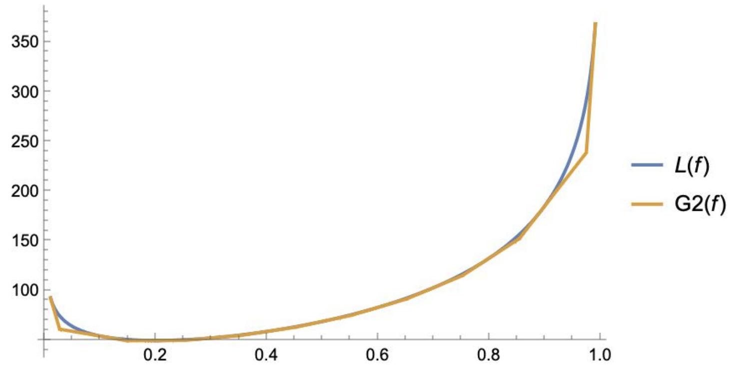
Approach #1: Piecewise Linear Approximation (*k*-PLA and PPLA)



Approximate one-dimensional convex loss function $L_i(f)$ with piecewise linear approximation using k pieces found via Taylor series expansion.

Extensions to arbitrary, convex loss functions

Approach #1: Piecewise Linear Approximation (*k*-PLA and PPLA)



Approximate one-dimensional convex loss function $L_i(f)$ with piecewise linear approximation using *k* pieces found via Taylor series expansion.

Approach #2: Structured Regression using Alternating Directions Method of Multipliers (ADMM)

ADMM:

$$\begin{aligned} x^{k+1} &:= \operatorname{argmin}_x L_\rho(x, z^k, y^k) && \textit{// x-minimization} \\ z^{k+1} &:= \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k) && \textit{// z-minimization} \\ y^{k+1} &:= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) && \textit{// dual update} \end{aligned}$$

Using ADMM, we reduce solving the Perfect Phylogeny Regression problem for **arbitrary convex loss functions to a sequence of L_2 subproblems**.

Fast regression under the perfect phylogeny model

Results (L_2 / least squares loss):

- *fastppm* achieved a *40-125x* speed up over the next best performing method *projectppm*.
- All methods achieved the exact same objective value on all instances.
- Blackbox convex optimization solvers were significantly slower than *projectppm* and *fastppm.*
- Excludes model build time which is in practice non-negligible for blackbox solvers.

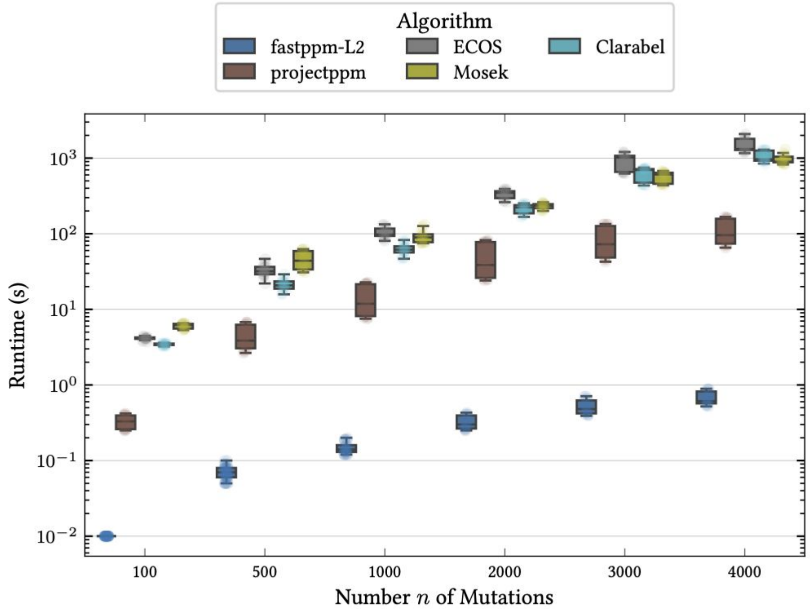


Figure: *Runtime of existing methods for the Perfect Phylogeny Regression problem when varying the number of clones/mutations.*

Improving existing phylogeny inference methods with *fastppm*

We replaced calls to existing perfect phylogeny regression algorithms in Sapling, CITUP, and Orchard with calls to *fastppm:*

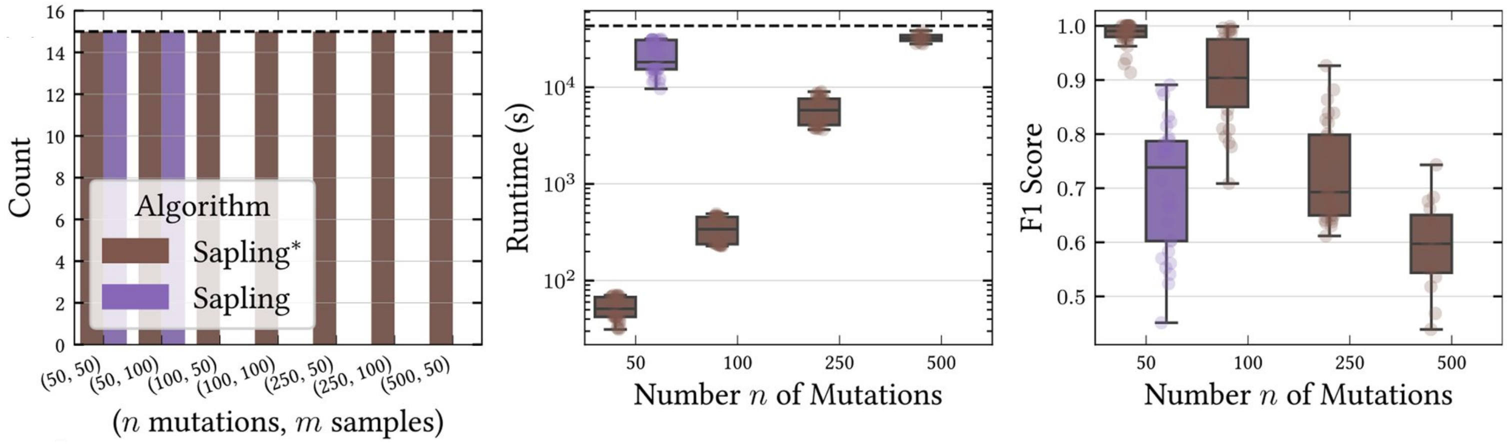


Figure: Number of successful instances within a twelve hour time limit, runtime, and accuracy of Sapling compared to Sapling* on simulated data.

Improving existing phylogeny inference methods with *fastppm*

We replaced calls to existing perfect phylogeny regression algorithms in Sapling, CITUP, and Orchard with calls to *fastppm:*

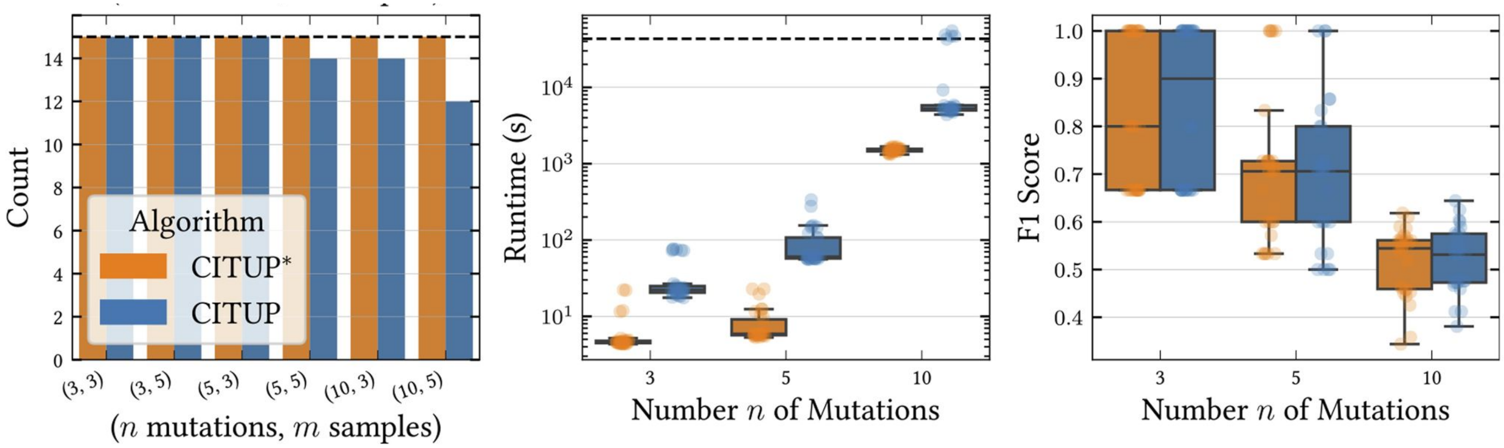


Figure: Number of successful instances within a twelve hour time limit, runtime, and accuracy of CITUP compared to CITUP* on simulated data.

Improving existing phylogeny inference methods with *fastppm*

We replaced calls to existing perfect phylogeny regression algorithms in Sapling, CITUP, and Orchard with calls to *fastppm:*

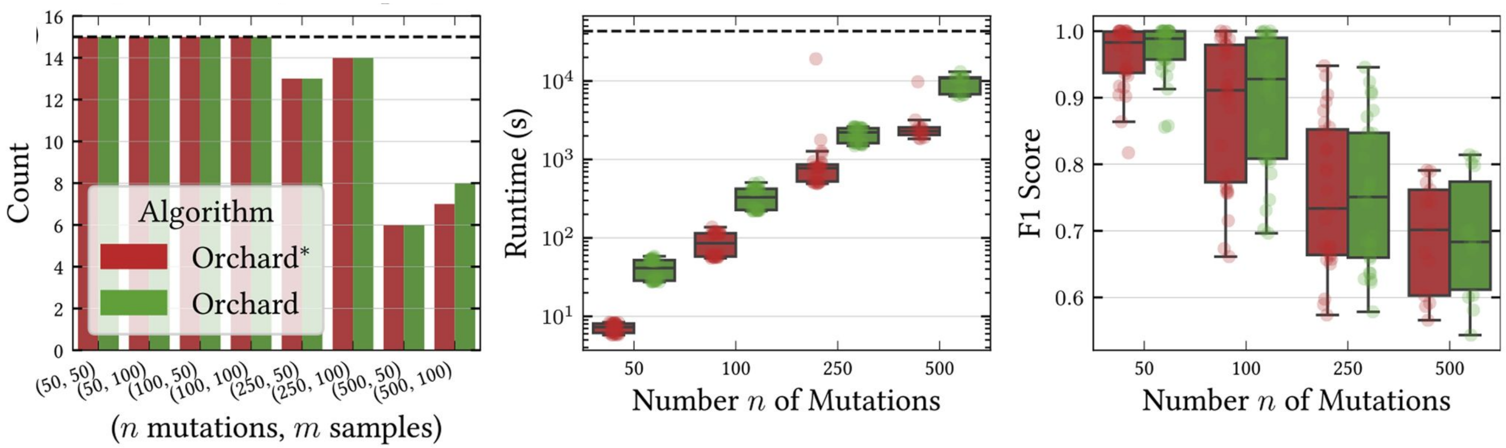
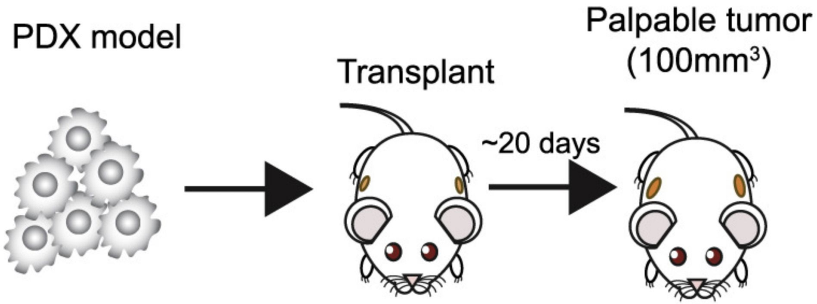


Figure: Number of successful instances within a twelve hour time limit, runtime, and accuracy of Orchard compared to Orchard* on simulated data.

fastppm improves frequency estimation in low coverage settings

Downsampled reads from a patient derived xenograft (POP66) is a mouse model of colorectal cancer (*n = 64* mutations, *m = 8* samples, 50x coverage):



Data from: *(Rehman et al. 2021)*

Method	Metric	Objective
Orchard*	$\|\|\tilde{\mathbf{F}} - \hat{\mathbf{F}}\|\|_F^2$	3.092
Orchard	$\|\|\tilde{\mathbf{F}} - \hat{\mathbf{F}}\|\|_F^2$	2.448
Sapling*	$\|\|\tilde{\mathbf{F}} - \hat{\mathbf{F}}\|\|_F^2$	**2.181**
Orchard*	$-\log \mathbb{P}(\mathbf{V} \mid \mathcal{T}, \hat{\mathbf{F}}, \mathbf{D})$	10790.9
Orchard	$-\log \mathbb{P}(\mathbf{V} \mid \mathcal{T}, \hat{\mathbf{F}}, \mathbf{D})$	10793.5
Sapling*	$-\log \mathbb{P}(\mathbf{V} \mid \mathcal{T}, \hat{\mathbf{F}}, \mathbf{D})$	**10720.6**

We applied Orchard, Orchard*, which use the L_2 loss, to Sapling*, which uses the binomial loss, to recover the clonal tree and mutation frequencies.

Thank You

Collaborators

Yuanyuan Qi (Co-first Author)

Mohammed El-Kebir

Ben Raphael

Group Members

Ben Raphael

Sereno Lopez-Darwin

Hirak Sarkar

Richard Zhang

Peter Halmos

Yihang Shen

W. Howard-Synder

Michael Wilson

Gillian Chu

Xinhao Liu

Henri Schmidt

Ahmed Shuaibi

Akhil Jakatdar

Gary Hu

Clover Zheng

Viola Chen

Julian Gold

The Raphael Lab

Mohammed El-Kebir

fastppm is implemented in C++ and is available on GitHub

Manuscript is accessible through *Bioinformatics*

SCRATCH

However, existing methods for the *PPR* problem are flawed

1. ***Do not directly*** model the read count data, which e.g. hinders analysis of ***low-coverage DNA sequencing***

However, existing methods for the *PPR* problem are flawed

1. ***Do not directly*** model the read count data, which e.g. hinders analysis of ***low-coverage DNA sequencing***

Example 1.

State-of-the-art phylogeny inference pipelines *CITUP, AncesTree, CALDER, Pairtree, Orchard,* and *fastBE* use the following two loss functions:

$$L_1(\mathbf{F}, \mathbf{V}, \mathbf{D}) = \sum_{i=1}^{m}\sum_{j=1}^{n} |f_{ij} - \tilde{f}_{ij}| \text{ where } \tilde{f}_{ij} = v_{ij}/d_{ij}$$

Observed frequency (VAF)

$$L_2(\mathbf{F}, \mathbf{V}, \mathbf{D}) = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}(f_{ij} - \tilde{f}_{ij})^2 \text{ where } \tilde{f}_{ij} = v_{ij}/d_{ij}, w_{ij} \geq 0$$

which do not directly model the read count data, instead collapsing it to a frequency.

However, existing methods for the *PPR* problem are flawed

1. ***Do not directly*** model the read count data, which e.g. hinders analysis of ***low-coverage DNA sequencing***
2. Employ ***slow***, black-box convex optimization software which ***do not exploit the structure of the regression problem***

Example 2.

In contrast, phylogeny inference pipelines (e.g. Sapling, PhyloWGS) which model observations using the probabilistic read-count model $v_{ij} \sim \text{Binomial}(f_{ij}, d_{ij})$, e.g.,

$$L_{\text{Bin}}(\mathbf{F}, \mathbf{V}, \mathbf{D}) = -\sum_{i=1}^{m}\sum_{j=1}^{n}[v_{ij}\log f_{ij} + (d_{ij} - v_{ij})\log(1 - f_{ij})].$$

must resort to **blackbox convex optimization software** which is prohibitively slow.

Tree structured dual dynamic programming (TSDDP)

Then, we solve the **dual** problem with a **bottom-up dynamic programming algorithm over T**.

(i) Fix a representation $\mathcal{R}(J_i)$ for each J_i.
(ii) Compute the representation $\mathcal{R}(J_i)$ at the leaf nodes.
(iii) Compute the representation $\mathcal{R}(J_i)$ at a node i provided the representations $\mathcal{R}(J_j)$ at all children $j \in \delta(i)$.
(iv) Solve the one-dimensional optimization problem $\max_{\alpha_0 \geq 0}\{-\alpha_0 + J_r(\alpha_0)\}$ using the representation of the root node $\mathcal{R}(J_r)$.

The TSDDP Algorithm