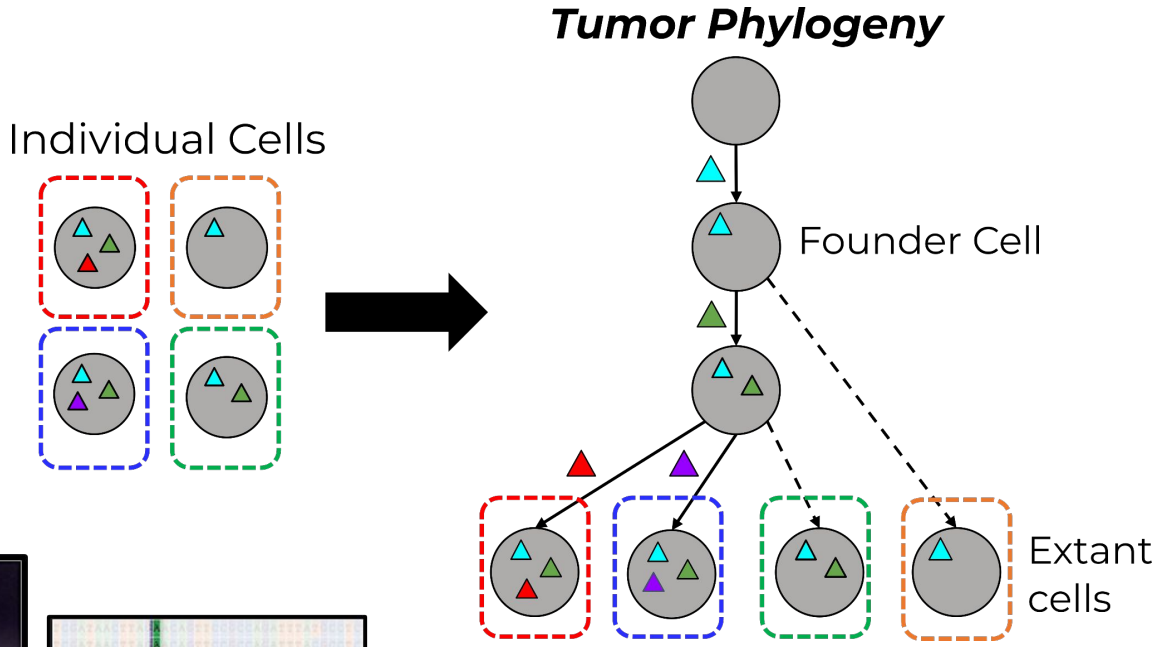
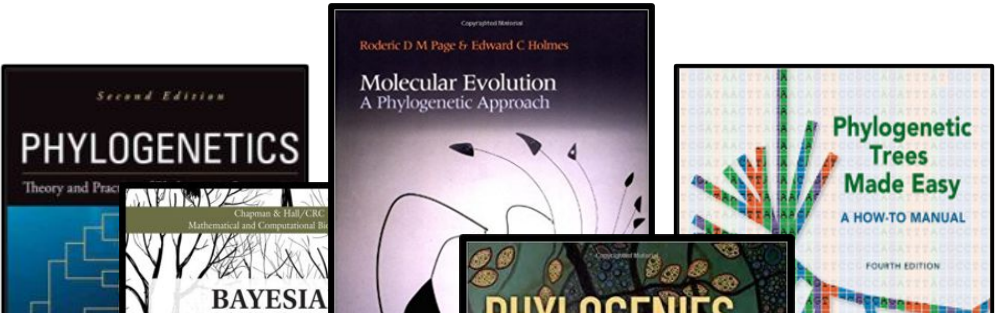
A regression based approach to phylogenetic reconstruction from multi-sample bulk DNA sequencing of tumors

Henri Schmidt and Benjamin J. Raphael
Department of Computer Science

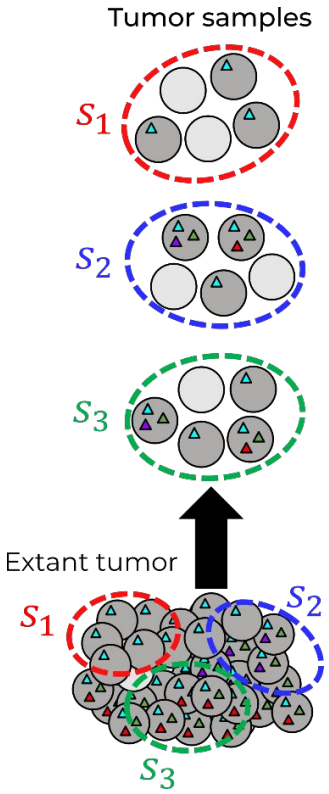
PRINCETON UNIVERSITY

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

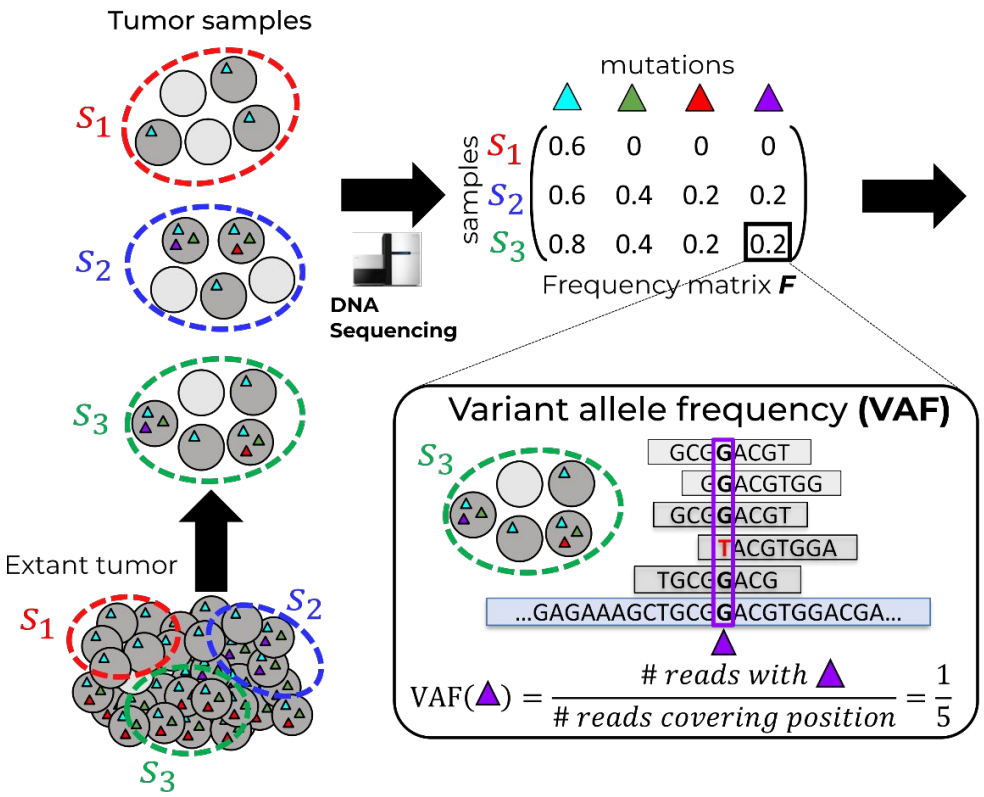




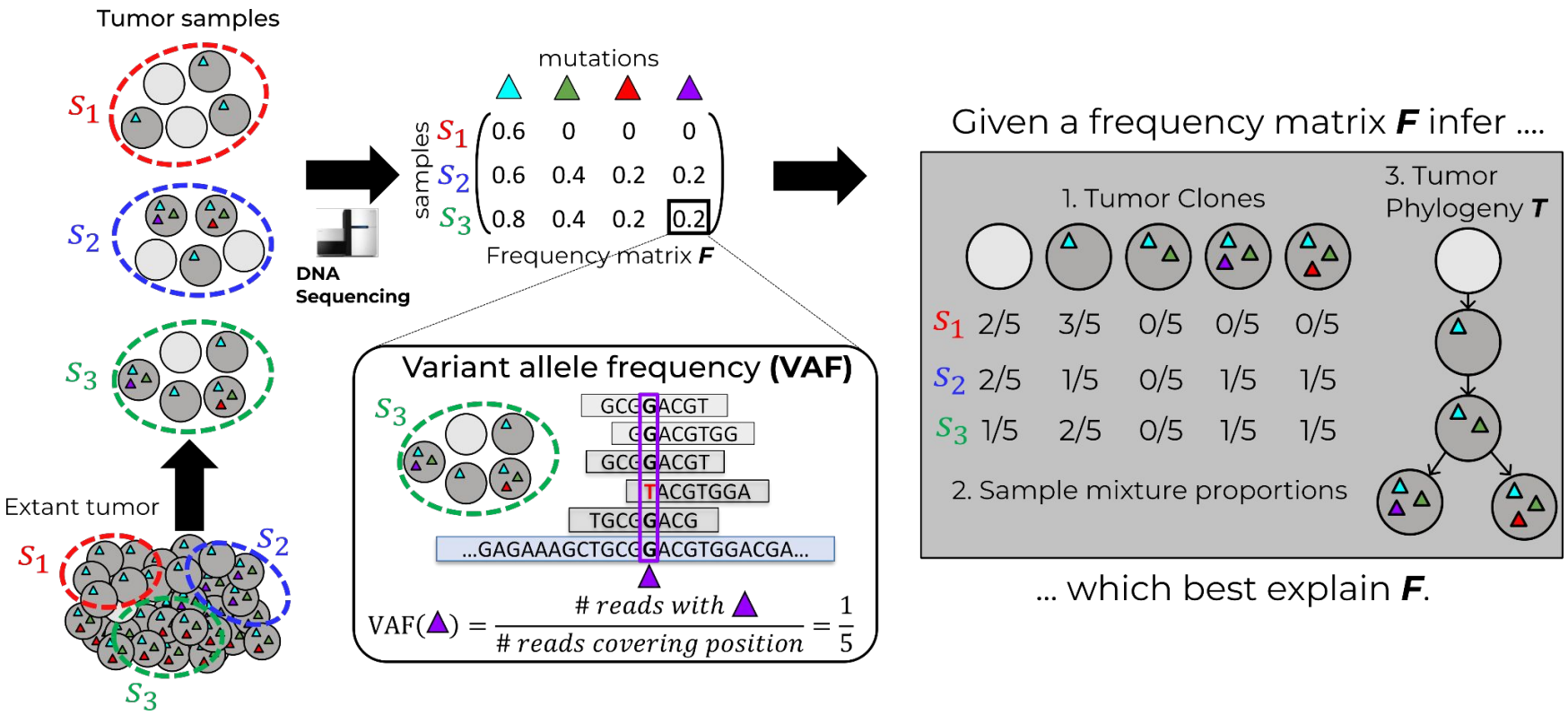
Bulk DNA sequencing yields a mixture of cells, requiring simultaneous inference of clones and their proportions



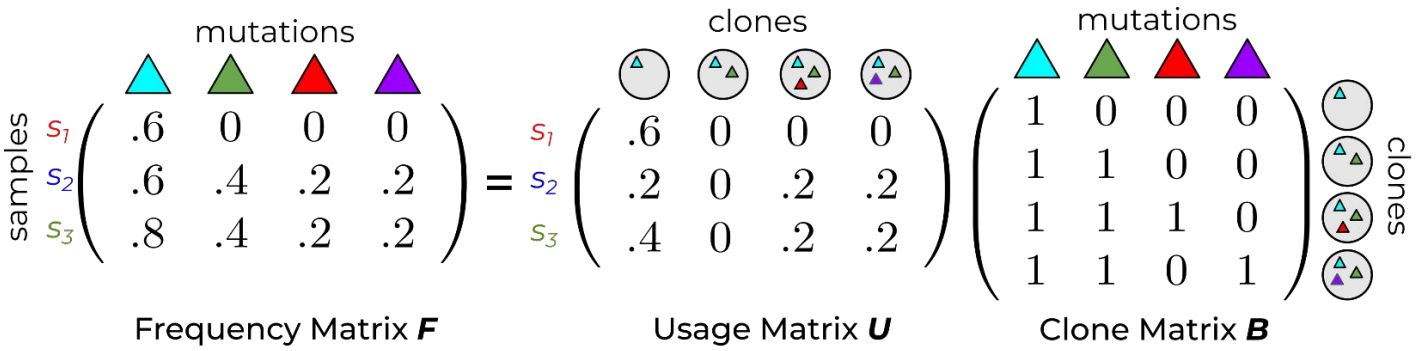
Bulk DNA sequencing yields a mixture of cells, requiring simultaneous inference of clones and their proportions



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

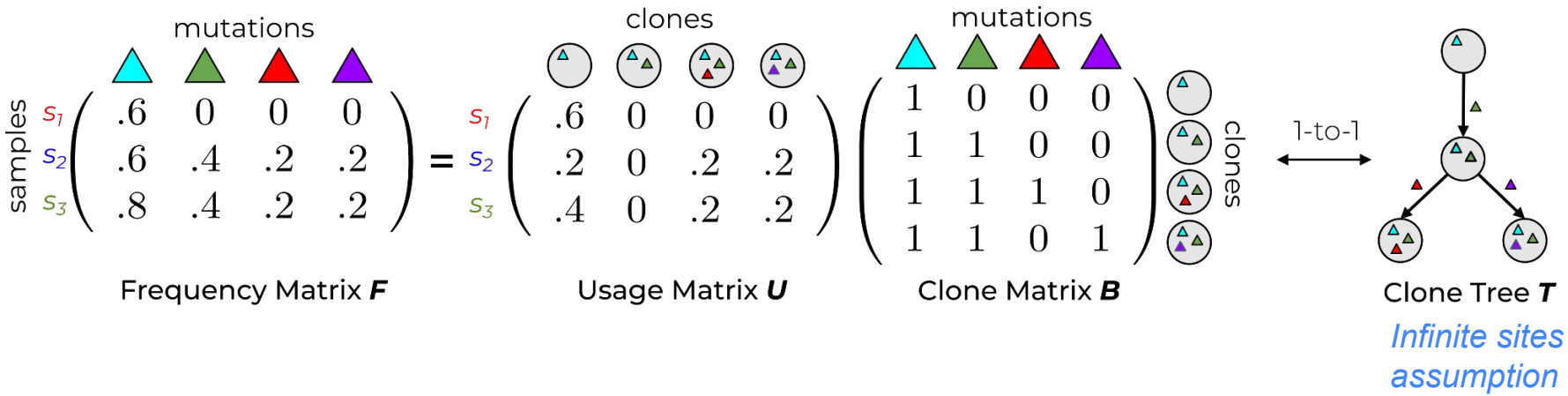


*Variant allele frequency (VAF) factorization model**



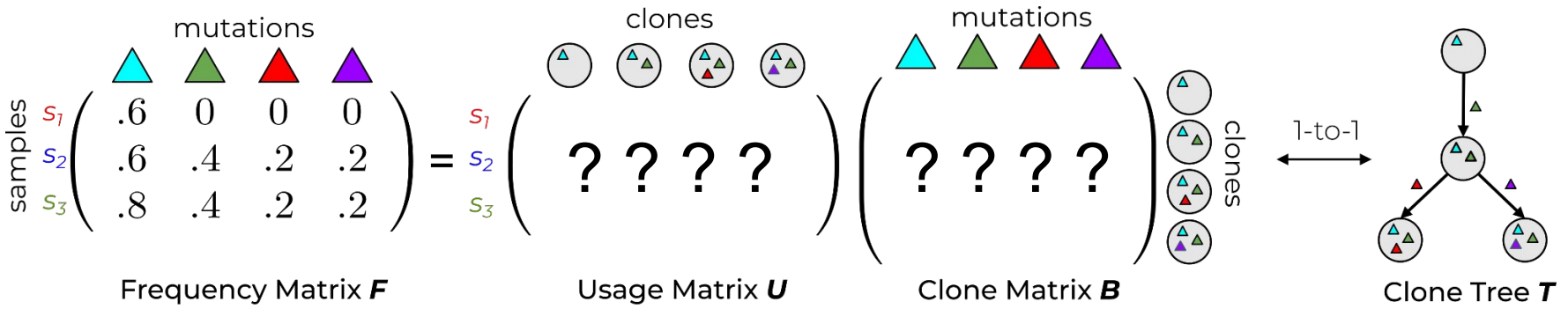
* This model is implicit or explicit in: **PhyloSub** *(Jiao et al., BMC Bioinform. 2014)*, **PhyloWGS** *(Deshwar et al., Genome Biol. 2015)*, **CITUP** *(Malikic et al., Bioinformatics 2015)*, **LICHeE** *(Popic et al., Genome Biol. 2015)*, **AncesTree** *(El-Kebir et al., Bioinformatics 2015)*, **Canopy** *(Jiang et al., PNAS 2016)*, **ClonEvol** *(Dang et al., Ann. Oncol. 2017)*, **CALDER** *(Myers et al., Cell Systems, 2019)*, **PairTree** *(Wintersinger et al., Blood Cancer Discovery, 2022)*, ***Orchard*** *(Kulman et al., 2023)*, **…**

*Variant allele frequency (VAF) factorization model**



* This model is implicit or explicit in: **PhyloSub** *(Jiao et al., BMC Bioinform. 2014)*, **PhyloWGS** *(Deshwar et al., Genome Biol. 2015)*, **CITUP** *(Malikic et al., Bioinformatics 2015)*, **LICHeE** *(Popic et al., Genome Biol. 2015)*, **AncesTree** *(El-Kebir et al., Bioinformatics 2015)*, **Canopy** *(Jiang et al., PNAS 2016)*, **ClonEvol** *(Dang et al., Ann. Oncol. 2017)*, **CALDER** *(Myers et al., Cell Systems, 2019)*, **PairTree** *(Wintersinger et al., Blood Cancer Discovery, 2022)*, ***Orchard*** *(Kulman et al., 2023)*, **…**

*Variant allele frequency (VAF) factorization model**



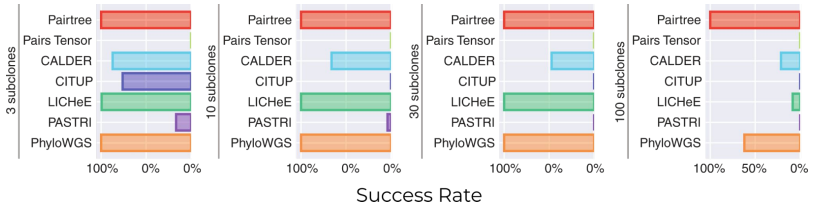
Infinite sites assumption

VAF factorization problem (VAFFP). Given an m-by-n frequency matrix F, find an m-by-n usage matrix U and clonal matrix B such that $F \approx UB$.

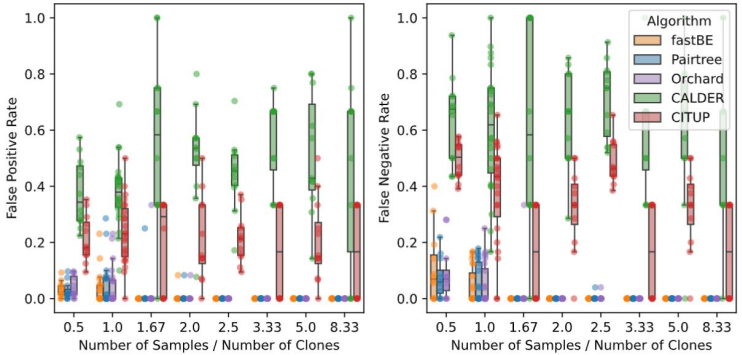
* This model is implicit or explicit in: **PhyloSub** *(Jiao et al., BMC Bioinform. 2014)*, **PhyloWGS** *(Deshwar et al., Genome Biol. 2015)*, **CITUP** *(Malikic et al., Bioinformatics 2015)*, **LICHeE** *(Popic et al., Genome Biol. 2015)*, **AncesTree** *(El-Kebir et al., Bioinformatics 2015)*, **Canopy** *(Jiang et al., PNAS 2016)*, **ClonEvol** *(Dang et al., Ann. Oncol. 2017)*, **CALDER** *(Myers et al., Cell Systems, 2019)*, **PairTree** *(Wintersinger et al., Blood Cancer Discovery, 2022)*, ***Orchard*** *(Kulman et al., 2023)*, **…**

Existing approaches for solving the VAF factorization problem, however, suffer from two important drawbacks

<u>Drawback 1.</u> Inability to scale to datasets with a large number of samples, clones, or mutations.



(Wintersinger et al. 2022): Existing methods fail to scale to datasets with more than 10 mutations!

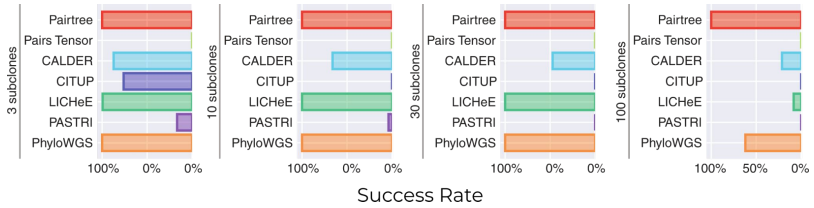


CALDER and CITUP perform poorly in terms of ancestral reconstruction accuracy, and do not improve as the ratio of samples to clones increases.

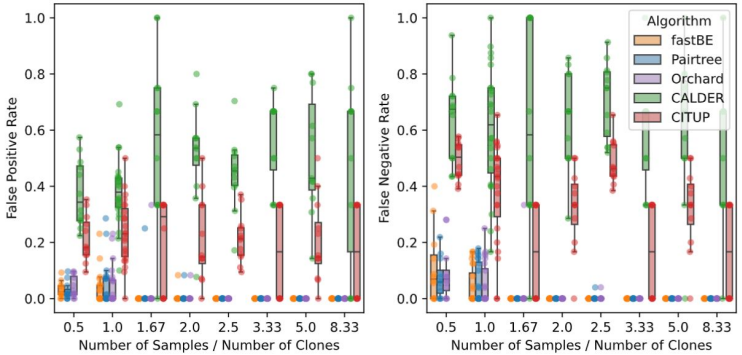
Existing approaches for solving the VAF factorization problem, however, suffer from two important drawbacks

<u>Drawback 1.</u> Inability to scale to datasets with a large number of samples, clones, or mutations.

<u>Drawback 2.</u> Poor phylogenetic reconstruction accuracy and little robustness to error.



(Wintersinger et al. 2022): Existing methods fail to scale to datasets with more than 10 mutations!



CALDER and CITUP perform poorly in terms of ancestral reconstruction accuracy, and do not improve as the ratio of samples to clones increases.

Contributions.

A **structured regression model** and a new method, **fastBE (fast bulk evolution),** for phylogenetic reconstruction from bulk DNA sequencing data, which:

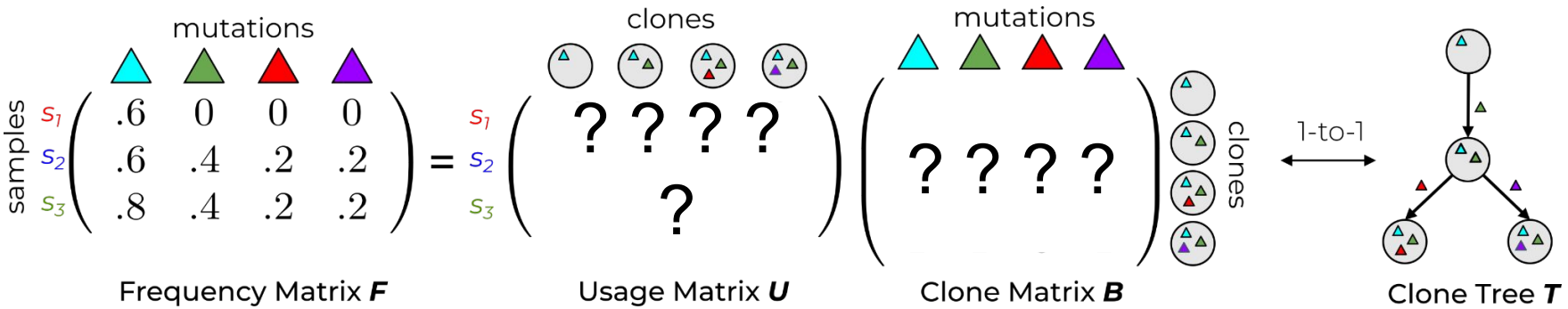
1. Scales to large instances ***containing thousands of mutations and hundreds of samples****.*

2. ***Accurately*** *reconstructs the phylogenetic tree while staying* ***robust*** *to error in the frequency matrix.*

Method	Reference	Scalable?	Accurate?
fastBE	This work	**Yes**	**Yes**
CITUP	(Malikic et al. 2015)	No	**Depends**
LICHeE	(Popic et al. 2015)	No	Unknown
AncesTree	(El-Kebir et al. 2015)	No	Unknown
CALDER	(Myers et al. 2019)	No	**Depends**
PhyloWGS	(Deshwar et al. 2015)	**Moderately**	No
PASTRI	(Satas et al. 2017)	No	No
Pairtree	(Wintersinger et al. 2022)	**Moderately**	**Yes**
Orchard	(Kulman et al. 2023)	**Yes**	**Yes**

*A summary of where existing methods land in terms of scalability and accuracy. *fastBE is several orders of magnitude faster than Orchard.*

The ℓ_1-*VAF factorization problem* (ℓ_1-*VAFFP*)

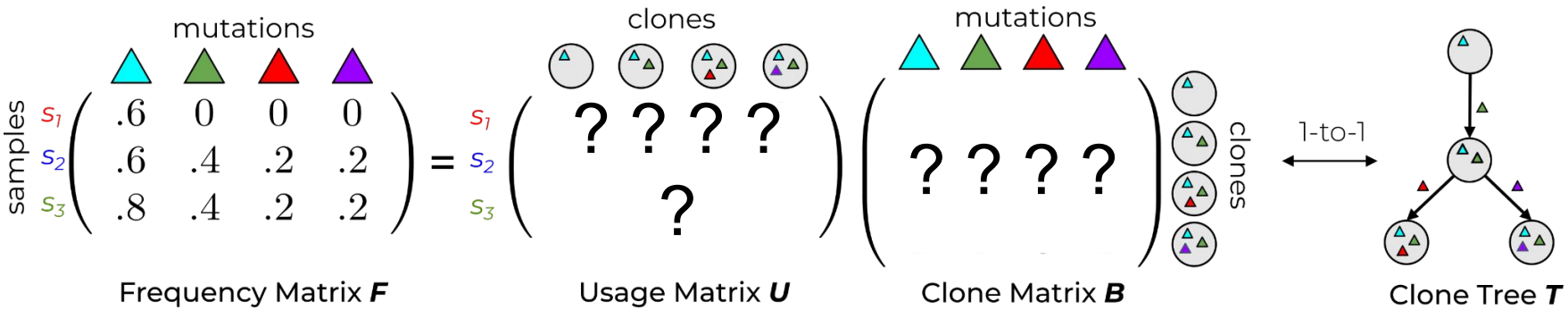


NP-Hard

ℓ_1-VAF factorization problem. Given an m-by-n frequency matrix F, find an m-by-n usage matrix U^* and n-clonal matrix B^* such that,

$$U^*, B^* = \arg\min_{U,B}\{\|F - UB\|_1 : U \geq 0, U\mathbf{1} \leq 1, B \text{ is a clonal matrix.}\}.$$

The ℓ_1-*VAF factorization problem* (ℓ_1-*VAFFP*)



NP-Hard

ℓ_1-VAF factorization problem. Given an m-by-n frequency matrix F, find an m-by-n usage matrix U^* and n-clonal matrix B^* such that,

$$U^*, B^* = \arg\min_{U,B}\{\|F - UB\|_1 : U \geq 0, U\mathbf{1} \leq 1, B \text{ is a clonal matrix.}\}.$$

Differences in problem formulation from existing combinatorial methods:

- No hard constraints on the error matrix $\varepsilon = F - UB$, as opposed to CALDER or AncesTree
- ℓ_1-norm of error matrix ε induces sparsity, as opposed to CITUP which uses the ℓ_2-norm
- ℓ_1-norm implies robustness to error in frequency matrix F, which no method

A structured regression model for the ℓ_1-*VAFFP*

To solve the NP-hard ℓ_1-VAFFP, we draw an analogy to the *structured regression models* used in distance based phylogenetics, which solves the NP-hard minimum evolution problem:

Minimum Evolution

Observed Data

$$\begin{array}{cc} & \text{species} \\ & \begin{array}{ccccc} s_1 & s_2 & s_3 & s_4 & s_5 \end{array} \\ \text{species} \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} & \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} \\ - & d_{22} & d_{23} & d_{24} & d_{25} \\ - & - & d_{33} & d_{34} & d_{35} \\ - & - & - & d_{44} & d_{45} \\ - & - & - & - & d_{55} \end{pmatrix} \end{array}$$

Distance Matrix ***D***

$$(d_{11}, d_{1,2}, \ldots, d_{5,5})$$

Distance Vector ***d***

Problem: *Find the tree whose induced distances best match the observed distance matrix D.*

A structured regression model for the ℓ_1-*VAFFP*

To solve the NP-hard ℓ_1-VAFFP, we draw an analogy to the *structured regression models* used in distance based phylogenetics, which solves the NP-hard minimum evolution problem:

Minimum Evolution

Observed Data

$$\begin{array}{cc} & \text{species} \\ & \begin{array}{ccccc} s_1 & s_2 & s_3 & s_4 & s_5 \end{array} \\ \text{species} \begin{array}{c} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} & \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} \\ - & d_{22} & d_{23} & d_{24} & d_{25} \\ - & - & d_{33} & d_{34} & d_{35} \\ - & - & - & d_{44} & d_{45} \\ - & - & - & - & d_{55} \end{pmatrix} \end{array}$$

Distance Matrix ***D***

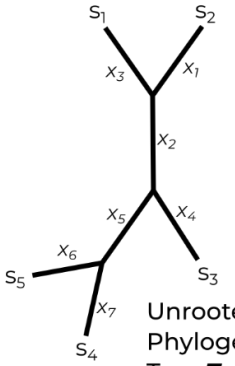
$$(d_{11}, d_{1,2}, \ldots, d_{5,5})$$

Distance Vector ***d***

Problem: *Find the tree whose induced distances best match the observed distance matrix D.*

Procedure for Distance Based Phylogenetics

1. Fix a tree *T*



Unrooted Phylogenetic Tree ***T***

2. Construct linear system

$$\widehat{d}_{1,2} = x_1 + x_3$$
$$\widehat{d}_{1,4} = x_2 + x_3 + x_5 + x_7$$
$$\widehat{d}_{1,5} = x_2 + x_3 + x_5 + x_6$$
$$\vdots$$
$$\widehat{d}_{4,5} = x_6 + x_7$$

where x_i are the non-negative branch lengths

equivalently

$$\widehat{d} = A_T x$$

3. Regression problem

Find branch lengths x_i such that the distance,

$$\|d - A_T x\|_2$$

is minimized.

4. Perturb tree and repeat

A structured regression model for the ℓ_1-*VAFFP*

Replacing the distance matrix with the frequency matrix *F* and the branch lengths with the usage proportions *U* suggests the following *structured regression model* for the ℓ_1-VAFFP:

Observed Data

mutations

ℓ_1 ***VAFFP***

samples

$$\begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \end{pmatrix}$$

Frequency Matrix ***F***

Problem: *Find the tree and usage matrix which best describes the frequency matrix* ***F.***

A structured regression model for the ℓ_1-*VAFFP*

Replacing the distance matrix with the frequency matrix *F* and the branch lengths with the usage proportions *U* suggests the following *structured regression model* for the ℓ_1-VAFFP:

Observed Data

mutations

ℓ_1 ***VAFFP***

samples

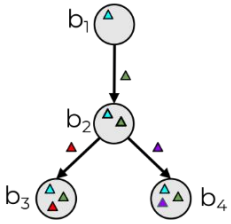
$$\begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \end{pmatrix}$$

Frequency Matrix ***F***

Problem: *Find the tree and usage matrix which best describes the frequency matrix* ***F.***

Procedure for Tumor Phylogenetics

1. Fix a tree *T*



4-Clonal Tree ***T***

2. Construct linear system

$$\widehat{f}_1 = u_{1,1}b_1 + u_{1,2}b_2 + u_{1,3}b_3 + u_{1,4}b_4$$
$$\widehat{f}_2 = u_{2,1}b_1 + u_{2,2}b_2 + u_{2,3}b_3 + u_{2,4}b_4$$
$$\widehat{f}_3 = u_{3,1}b_1 + u_{3,2}b_2 + u_{3,3}b_3 + u_{3,4}b_4$$

where $u_{i,j}$ are non-negative usage proportions

3. Regression problem

Given F and B_T, find the usage matrix U such that the error,

$$\|F - UB_T\|_1$$

is minimized.

4. Perturb tree and repeat

A structured regression model for the ℓ_1-*VAFFP*

Replacing the distance matrix with the frequency matrix *F* and the branch lengths with the usage proportions *U* suggests the following *structured regression model* for the ℓ_1-VAFFP:

Observed Data

mutations

ℓ_1 ***VAFFP***

samples

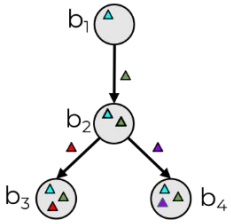
$$\begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \end{pmatrix}$$

Frequency Matrix ***F***

Problem: *Find the tree and usage matrix which best describes the frequency matrix* ***F.***

Procedure for Tumor Phylogenetics

1. Fix a tree *T*



4-Clonal Tree ***T***

2. Construct linear system

$$\widehat{f}_1 = u_{1,1}b_1 + u_{1,2}b_2 + u_{1,3}b_3 + u_{1,4}b_4$$
$$\widehat{f}_2 = u_{2,1}b_1 + u_{2,2}b_2 + u_{2,3}b_3 + u_{2,4}b_4$$
$$\widehat{f}_3 = u_{3,1}b_1 + u_{3,2}b_2 + u_{3,3}b_3 + u_{3,4}b_4$$

where $u_{i,j}$ are non-negative usage proportions

3. Regression problem

Given F and B_T, find the usage matrix U such that the error,

$$\|F - UB_T\|_1$$

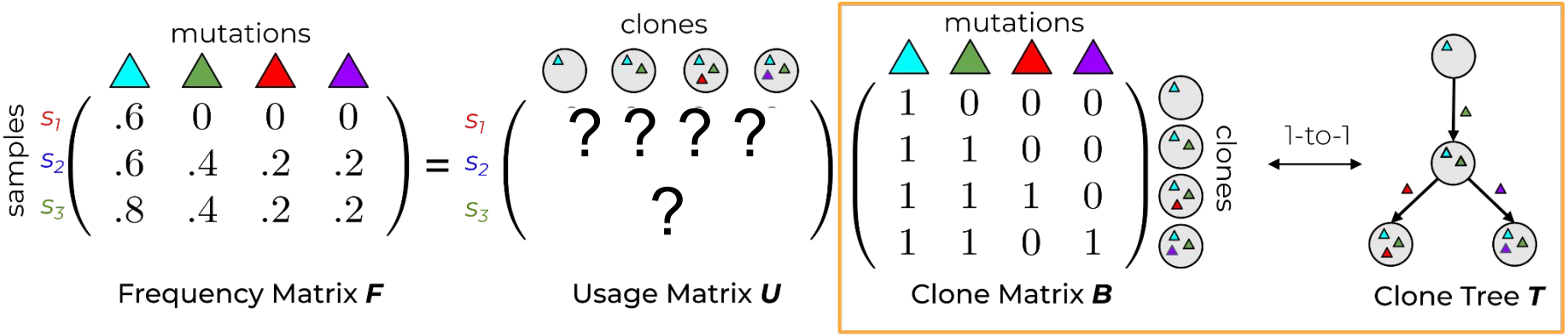
is minimized.

4. Perturb tree and repeat

To make this procedure scale, we need an *efficient algorithm* for the regression problem.

A structured regression problem: the ℓ_1-VAF regression problem

The tree is now fixed



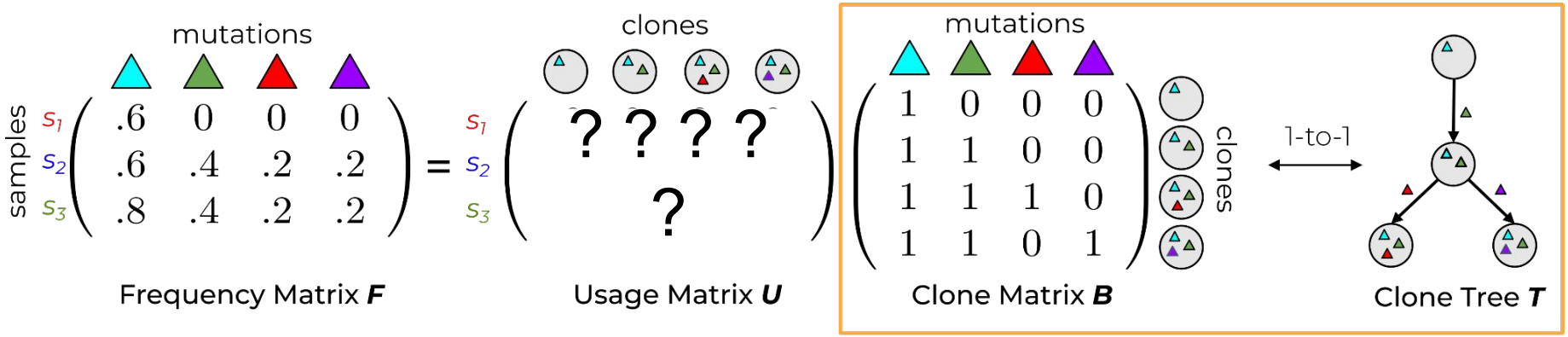
Polynomial Time

ℓ_1-VAF regression problem. Given an m-by-n frequency matrix F and an n-by-n clonal matrix B, find an m-by-n usage matrix U^* such that,

$$U^* = \arg\min_U \{\|F - UB\|_1 : U \geq 0, U\mathbf{1} \leq 1\}.$$

A structured regression problem: the ℓ_1-VAF regression problem

The tree is now fixed



Polynomial Time

ℓ_1-VAF regression problem. Given an m-by-n frequency matrix F and an n-by-n clonal matrix B, find an m-by-n usage matrix U^* such that,

$$U^* = \arg\min_U \{\|F - UB\|_1 : U \geq 0, U\mathbf{1} \leq 1\}.$$

In contrast to the ℓ_1-VAF factorization problem, this regression problem is solvable in polynomial time via linear programming…

A structured regression problem: the ℓ_1-VAF regression problem

Polynomial Time

ℓ_1-VAF regression problem. Given an m-by-n frequency matrix F and an n-by-n clonal matrix B, find an m-by-n usage matrix U^* such that,

$$U^* = \arg\min_U \{\|F - UB\|_1 : U \geq 0, U\mathbf{1} \leq 1\}.$$

- ℓ_1-VAF regression problem is solvable in polynomial time with a *linear*

...

A structured regression problem: the ℓ_1-VAF regression problem

Polynomial Time

ℓ_1-VAF regression problem. Given an m-by-n frequency matrix F and an n-by-n clonal matrix B, find an m-by-n usage matrix U^* such that,

$$U^* = \arg\min_U \{\|F - UB\|_1 : U \geq 0, U\mathbf{1} \leq 1\}.$$

- ℓ_1-VAF regression problem is solvable in polynomial time with a *linear program*
- Linear programming does not exploit the structure of the clonal matrix *B*…

A structured regression problem: the ℓ_1-VAF regression problem

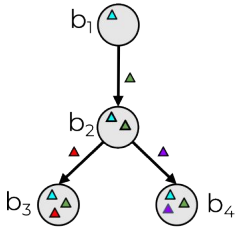
Polynomial Time

ℓ_1-VAF regression problem. Given an m-by-n frequency matrix F and an n-by-n clonal matrix B, find an m-by-n usage matrix U^* such that,

$$U^* = \arg\min_U \{\|F - UB\|_1 : U \geq 0, U\mathbf{1} \leq 1\}.$$

- ℓ_1-VAF regression problem is solvable in polynomial time with a *linear program*
- Linear programming does not exploit the structure of the clonal matrix *B*…

1. Fix a tree *T*



4-Clonal Tree ***T***

2. Regression problem

Given F and B_T, find the usage matrix U such that the error,

$$\|F - UB_T\|_1$$

is minimized.

3. Perturb tree and repeat

… Since regression problem is solved many times, need an extremely fast algorithm

An ultrafast algorithm for the ℓ_1-VAF regression problem

By exploiting the structure in the clonal matrix ***B*** appearing in the regression problem…

Theorem 1. *Given a clonal tree $\mathcal{T}$ with n vertices and an m-by-n frequency matrix F, the minimum*

$$L_1^*(F, B_{\mathcal{T}}) = \min \{\|F - UB_{\mathcal{T}}\|_1 : U \geq 0, U\mathbb{1} \leq 1\}$$

can be found in $O(mnd)$ where d is the depth of $\mathcal{T}$.

… we obtain an algorithm for the ℓ_1-VAF regression problem which outperforms state-of-the-art linear programming solvers in both theory and practice.

An ultrafast algorithm for the ℓ_1-VAF regression problem

By exploiting the structure in the clonal matrix ***B*** appearing in the regression problem…

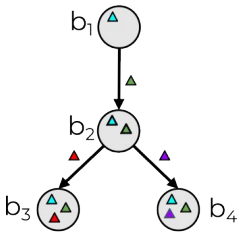
Theorem 1. *Given a clonal tree $\mathcal{T}$ with n vertices and an m-by-n frequency matrix F, the minimum*

$$L_1^*(F, B_{\mathcal{T}}) = \min \{\|F - UB_{\mathcal{T}}\|_1 : U \geq 0, U\mathbb{1} \leq 1\}$$

can be found in $O(mnd)$ where d is the depth of $\mathcal{T}$.

… we obtain an algorithm for the ℓ_1-VAF regression problem which outperforms state-of-the-art linear programming solvers in both theory and practice.

1. Fix a tree *T*



4-Clonal Tree ***T***

2. Regression problem

Given F and B_T,
find the usage matrix
U such that the error,

$$\|F - UB_T\|_1$$

is minimized.

Our fast regression algorithm also serves as a useful "primitive" and "building block" in the development of other methods.

An ultrafast algorithm for the ℓ_1-VAF regression problem

Second, our regression algorithm is more efficient in the *online* setting where the tree undergoes slight perturbations…

Corollary 1. *Given a clonal tree $\mathcal{T}$ with n vertices and an m-by-n frequency matrix F, the following queries can be efficiently answered after $O(mnd)$ pre-processing time using $O(mnd)$ space.*

(i) For a subtree prune-and-regraft (SPR) operation on vertices i and j which results in a tree $\mathcal{T}'$, the minimum $L_1^*(F, B_{\mathcal{T}'})$ can be queried in $O(md \cdot \max\{d(i), d(j)\})$ time.

(ii) For the operation of attaching a new vertex j as a child of a vertex i to obtain a tree $\mathcal{T}'$ and appending a corresponding column to the frequency matrix F to obtain F', the minimum $L_1^*(F', B_{\mathcal{T}'})$ can be queried in $O(md \cdot d(i))$ time.

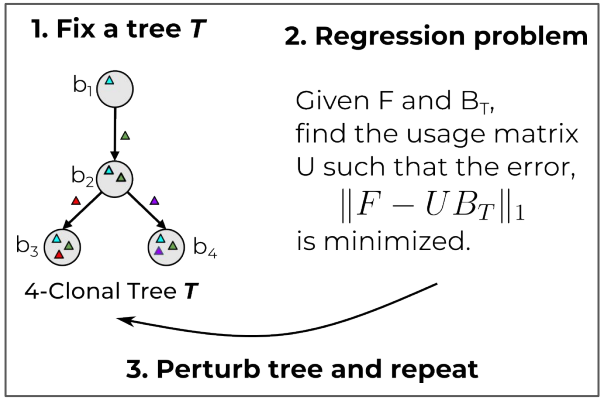
An ultrafast algorithm for the ℓ_1-VAF regression problem

Second, our regression algorithm is more efficient in the *online* setting where the tree undergoes slight perturbations…

Corollary 1. *Given a clonal tree $\mathcal{T}$ with n vertices and an m-by-n frequency matrix F, the following queries can be efficiently answered after $O(mnd)$ pre-processing time using $O(mnd)$ space.*

(i) For a subtree prune-and-regraft (SPR) operation on vertices i and j which results in a tree $\mathcal{T}'$, the minimum $L_1^*(F, B_{\mathcal{T}'})$ can be queried in $O(md \cdot \max\{d(i), d(j)\})$ time.

(ii) For the operation of attaching a new vertex j as a child of a vertex i to obtain a tree $\mathcal{T}'$ and appending a corresponding column to the frequency matrix F to obtain F', the minimum $L_1^*(F', B_{\mathcal{T}'})$ can be queried in $O(md \cdot d(i))$ time.



… making our regression algorithm fit for solving the harder factorization problem.

An ultrafast algorithm for the ℓ_1-VAF regression problem

Theorem 1. *Given a clonal tree $\mathcal{T}$ with n vertices and an m-by-n frequency matrix F, the minimum*

$$L_1^*(F, B_{\mathcal{T}}) = \min \{\|F - UB_{\mathcal{T}}\|_1 : U \geq 0, U\mathbb{1} \leq 1\}$$

can be found in $O(mnd)$ where d is the depth of $\mathcal{T}$.

A technical comparison of our algorithm to other approaches:

1. Depth $d \approx n^{1/2}\log(n)$ for almost all trees *(Chung et al., Journal of Graph Theory, 2012)*
2. Fastest LP solvers have $O(mn^{2.5})$ time complexity: outperform both asymptotically and empirically
3. Fastest known algorithm *(Jia et al. NeurIPS 2018)* for the ℓ_2 regression problem runs in $O(mn^2)$ time – does not handle online setting

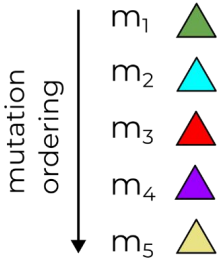
fastBE a scalable method for the ℓ_1-VAF factorization problem

Using our structured regression framework, we develop a simple greedy algorithm, ***fastBE (fast Bulk Evolution)***, for the ℓ_1-VAF factorization problem…

fastBE a scalable method for the ℓ_1-VAF factorization problem

Using our structured regression framework, we develop a simple greedy algorithm, ***fastBE (fast Bulk Evolution)***, for the ℓ_1-VAF factorization problem…

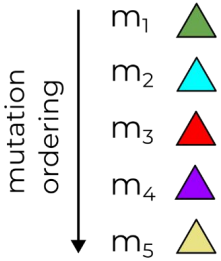
Step 1. Fix a mutation ordering



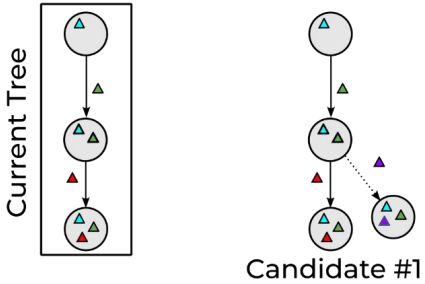
fastBE a scalable method for the ℓ_1-VAF factorization problem

Using our structured regression framework, we develop a simple greedy algorithm, ***fastBE (fast Bulk Evolution)***, for the ℓ_1-VAF factorization problem…

Step 1. Fix a mutation ordering



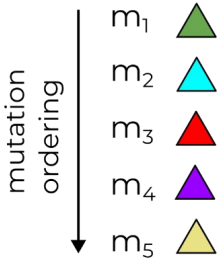
Step 2. For each mutation, enumerate all candidate trees obtained by the mutation's addition



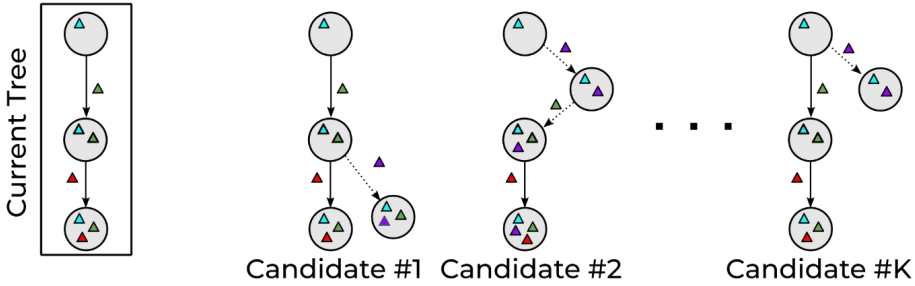
fastBE a scalable method for the ℓ_1-VAF factorization problem

Using our structured regression framework, we develop a simple greedy algorithm, ***fastBE (fast Bulk Evolution)***, for the ℓ_1-VAF factorization problem…

Step 1. Fix a mutation ordering



Step 2. For each mutation, enumerate all candidate trees obtained by the mutation's addition

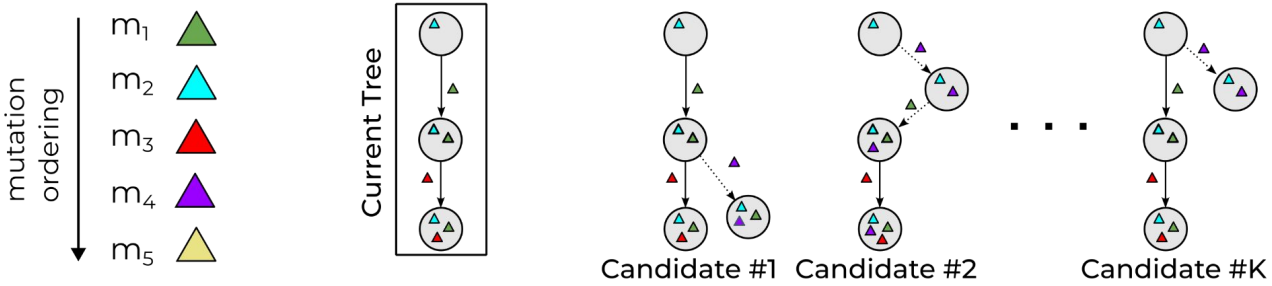


fastBE a scalable method for the ℓ_1-VAF factorization problem

Using our structured regression framework, we develop a simple greedy algorithm, ***fastBE (fast Bulk Evolution)***, for the ℓ_1-VAF factorization problem…

Step 1. Fix a mutation ordering

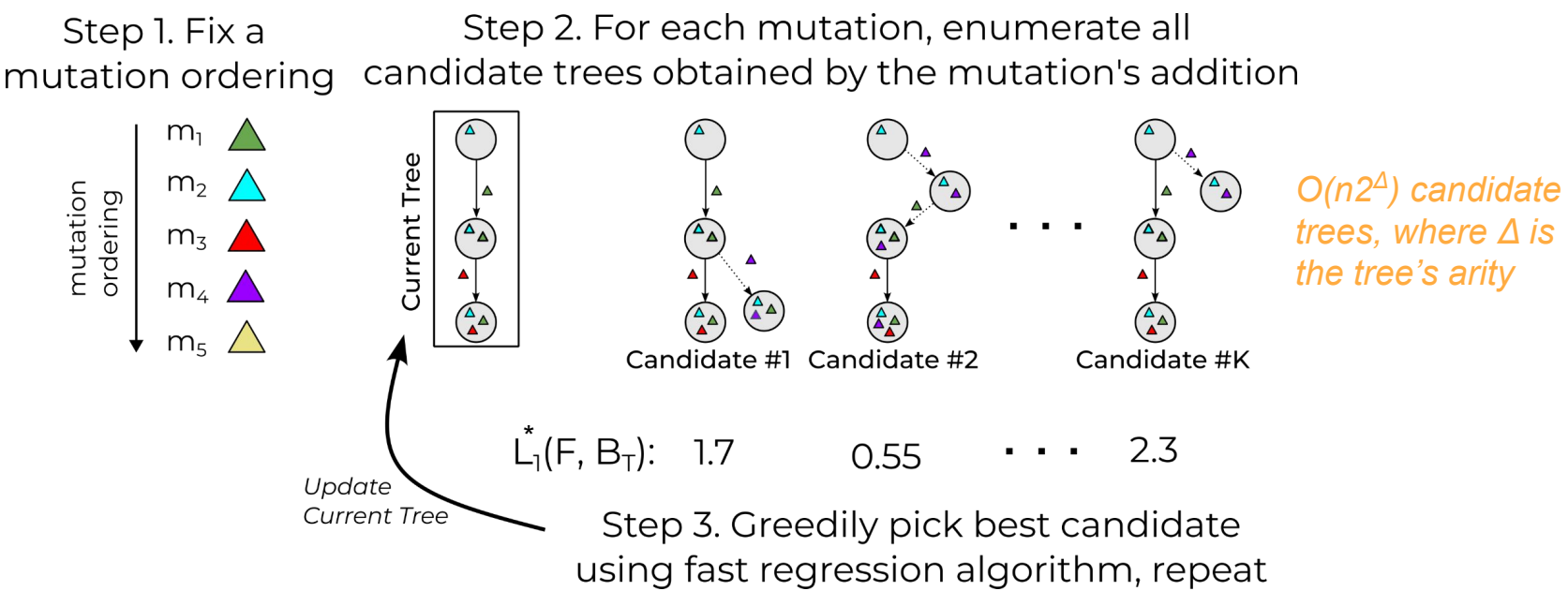
Step 2. For each mutation, enumerate all candidate trees obtained by the mutation's addition



$O(n2^{\Delta})$ candidate trees, where Δ is the tree's arity

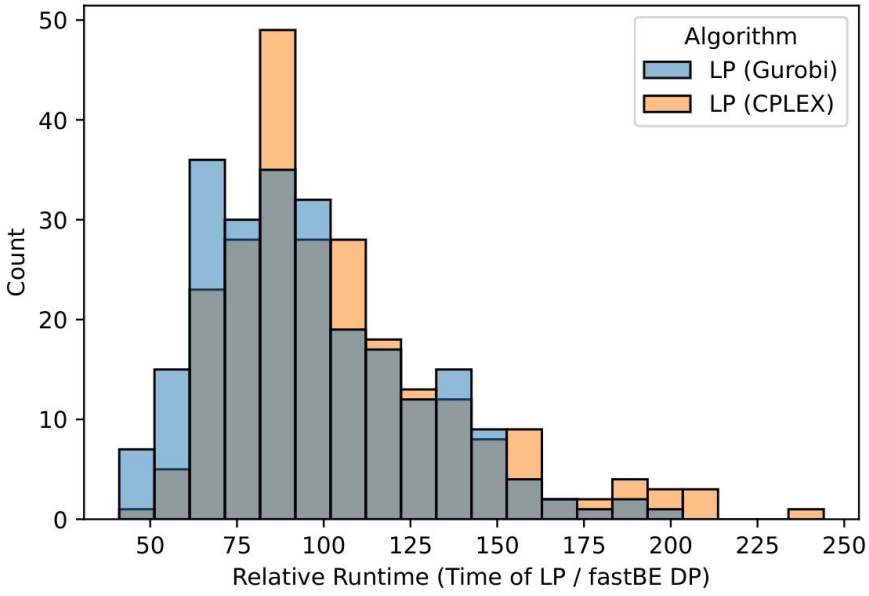
fastBE a scalable method for the ℓ_1-VAF factorization problem

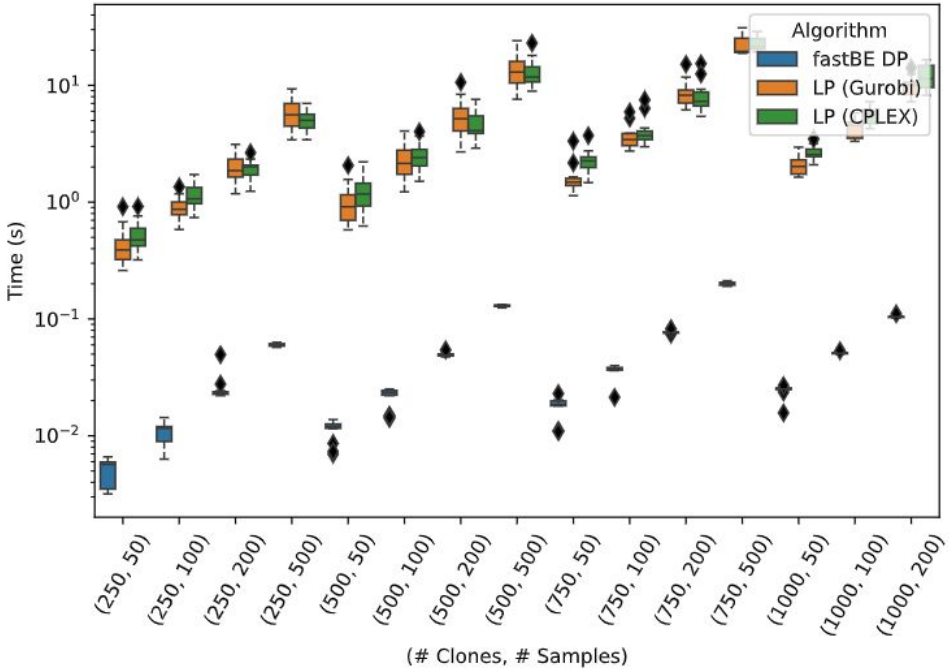
Using our structured regression framework, we develop a simple greedy algorithm, ***fastBE (fast Bulk Evolution)***, for the ℓ_1-VAF factorization problem…



Empirical Results

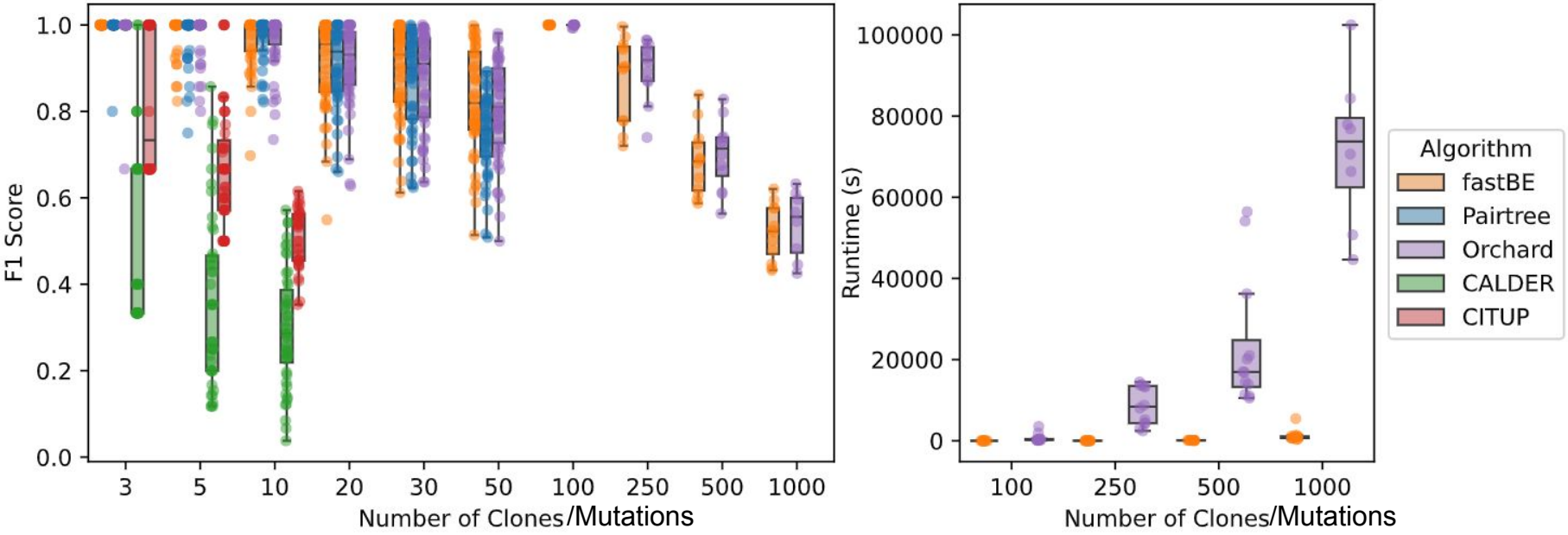
Our structured regression algorithm outperforms state of the art linear programming solvers





Left: Relative runtime to solve ℓ_1 VAF regression problem. Right: Absolute runtime to solve ℓ_1 VAF regression problem versus the number of samples and clones.

fastBE outperforms existing methods on simulated data

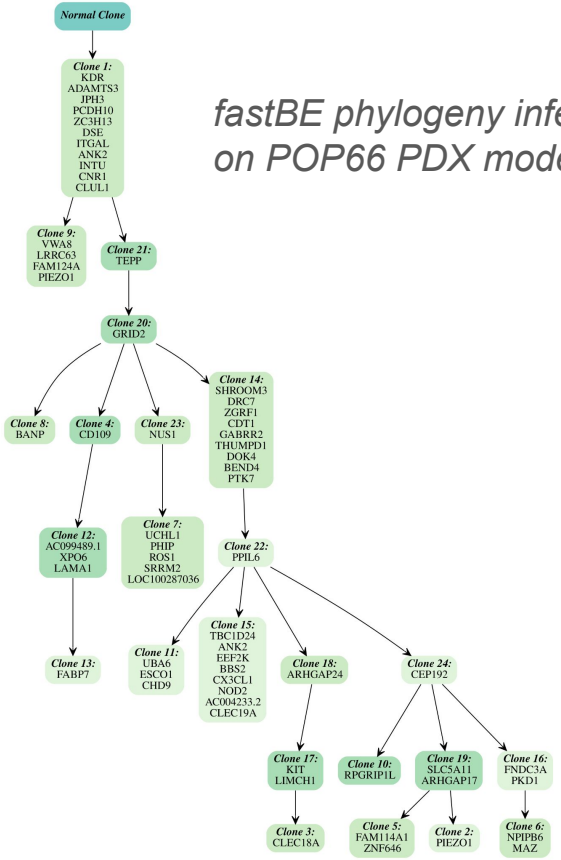


Left: Pairwise relationship error (F1) between simulated ground truth and inferred trees. Right: Wall clock runtime (s) of fastBE and Orchard on instances with ≥ 100 clones.

Evaluation on POP66 colorectal cancer model from *(Rehman et al. Cell, 2021)*

Sample	fastBE Violation V	Pairtree Violation V
Patient (P0)	**0.0761**	0.1888
Xenograft (G0)	**0.0180**	0.0854
Vehicle tumor 1	**0.3221**	0.6967
Vehicle tumor 2	**0.4277**	0.8584
CPT-11 Regrowth	**0.8822**	**0.8822**
CPT-11 Resistant #1	0.5149	**0.2640**
CPT-11 Resistant #2	**0.2704**	0.7282
CPT-11 Resistant #3	**0.4143**	0.6897

Total violation of the sum condition for the fastBE and Pairtree inferred phylogenetic trees on the POP66 colorectal cancer model.



fastBE phylogeny inferred on POP66 PDX model

Conclusion & Future Work

Contributions:

- We developed a structured regression framework and associated theory for phylogenetic reconstruction from bulk DNA sequencing data
- Using this framework, we developed a method, *fastBE,* that efficiently infers phylogenies and outperforms existing methods in terms of both time and accuracy on simulated and real data

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

The manuscript is available on bioRxiv

Thank You

Group Members

Ben Raphael	Gillian Chu
Metin Balaban	Xinhao Liu
Cong Ma	**Henri Schmidt**
Uyen Mai	Ahmed Shuaibi
Palash Sashittal	Alexander Strzalkowski
Uthsav Chitra	Akhil Jakatdar
Sereno Lopez-Darwin	Gary Hu
Hirak Sarkar	Clover Zheng
Richard Zhang	Viola Chen
Peter Halmos	Julian Gold

The Raphael Lab

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

The manuscript is available on bioRxiv