# Algorithms for Analyzing Intraspecific Sequence Variation

Srinath Sridhar

Computer Science Department
Carnegie Mellon University

March 2, 2009

# Outline

1. Motivation

2. Phylogeny Reconstruction
   - Definitions
   - Imperfect Phylogeny Reconstruction
   - Extensions
   - Empirical Results

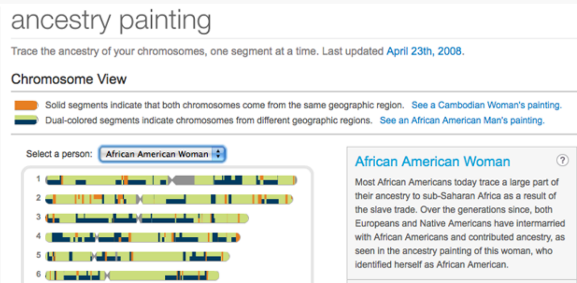
3. Population Substructure
   - Pure Populations
   - Admixture

# Intra-specific Variation

- How can we characterize and use genomic variation that exists within a single species to understand its *recent* history?

# Significance

- Fundamental to understanding of genome variation
- Disease association tests: ensure association of SNPs to cases/controls not underlying population substructure
- Direct to consumer genotyping: ancestry and life-time risks



ancestry painting

Trace the ancestry of your chromosomes, one segment at a time. Last updated April 23th, 2008.

Chromosome View

🟧 Solid segments indicate that both chromosomes come from the same geographic region. See a Cambodian Woman's painting.

▨ Dual-colored segments indicate chromosomes from different geographic regions. See an African American Man's painting.

Select a person: African American Woman

African American Woman

Most African Americans today trace a large part of their ancestry to sub-Saharan Africa as a result of the slave trade. Over the generations since, both Europeans and Native Americans have intermarried with African Americans and contributed ancestry, as seen in the ancestry painting of this woman, who identified herself as African American.

# Analysis of Genetic Variation

- Finding genetic variation
    - What forms of variation does the genome exhibit?
- Analyzing evolution of the genome
    - How does one genome transform to another?
- Analyzing genetic distribution in populations
    - How do the variants characterize sub-populations?

# Analysis of Genetic Variation

- Finding genetic variation
  - What forms of variation does the genome exhibit?
- Analyzing evolution of the genome
  - How does one genome transform to another?
- Analyzing genetic distribution in populations
  - How do the variants characterize sub-populations?

# Analysis of Genetic Variation

- Finding genetic variation
    - What forms of variation does the genome exhibit?
- Analyzing evolution of the genome
    - How does one genome transform to another?
- Analyzing genetic distribution in populations
    - How do the variants characterize sub-populations?

# Finding Genetic Variation

- Large segments of mouse genome missing or duplicated
- Newer form of large-scale variation
- Joint work with Cold Spring Harbor Labs; *Nature Genetics 2007*

### Citation

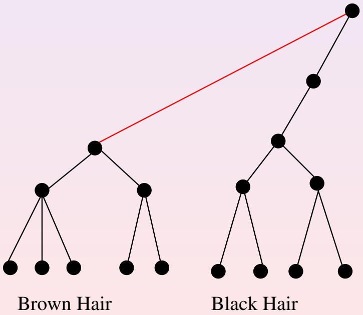
'Breakthrough of the year 2007' – *Science magazine*

# Evolution of Genome

## First Part of Talk

Phylogeny reconstruction

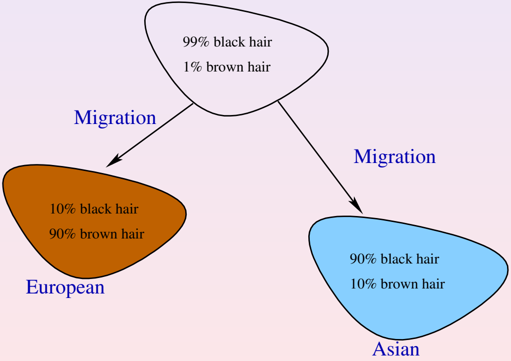
Vertex: an individual's Chromosome 2



# Genetic Distribution in Populations

## Second part of Talk

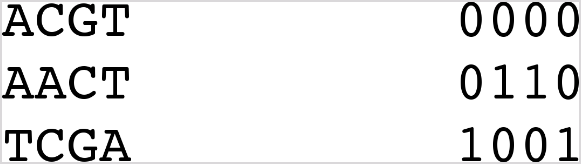
Substructure in populations



# Single Nucleotide Polymorphisms (SNPs)

- Variation due to single base change (SNPs)
- Only two bases per site
- Data-set represented by binary $n \times m$ matrix

## Example



| | |
|---|---|
| ACGT | 0 0 0 0 |
| AACT | 0 1 1 0 |
| TCGA | 1 0 0 1 |

# Outline

1. Motivation
2. Phylogeny Reconstruction
   - Definitions
   - Imperfect Phylogeny Reconstruction
   - Extensions
   - Empirical Results
3. Population Substructure
   - Pure Populations
   - Admixture

# Outline

1. Motivation
2. Phylogeny Reconstruction
   - Definitions
   - Imperfect Phylogeny Reconstruction
   - Extensions
   - Empirical Results
3. Population Substructure
   - Pure Populations
   - Admixture

# Phylogeny Reconstruction

- Input matrix $I$: $n \times m$ binary
- Rows: taxa (chromosomes of individuals)
- Columns: sites (SNPs)
- Assume all sites contain both $0, 1$

# Phylogeny Reconstruction

## Definition

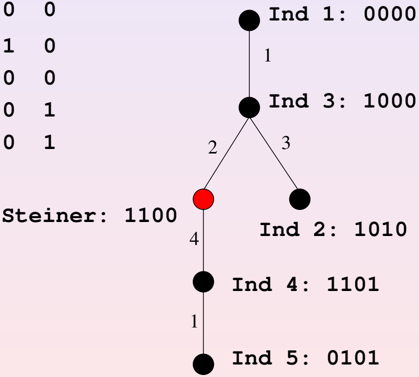
A *phylogeny* is an unrooted tree $T(V, E)$ where each vertex $v \in \{0,1\}^m$ represents a taxon and an edge represents a *single* mutation (Hamming distance 1). Then $\texttt{length}(T) = |E|$.

## Definition

A vertex $v$ that represents an input taxon is called a *terminal* vertex. Every other vertex is a *Steiner* vertex.

# Example

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Individual 1: | 0 | 0 | 0 | 0 |
| Individual 2: | 1 | 0 | 1 | 0 |
| Individual 3: | 1 | 0 | 0 | 0 |
| Individual 4: | 1 | 1 | 0 | 1 |
| Individual 5: | 0 | 1 | 0 | 1 |



# Imperfection of Phylogeny

Any phylogeny has length *at least* $m$

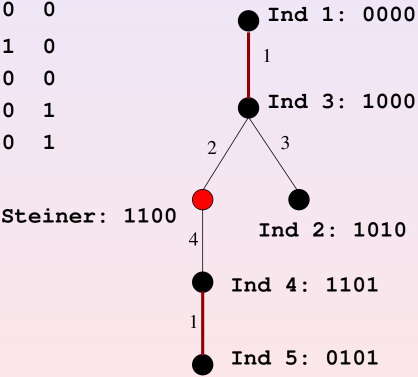
**Definition**

Phylogeny $T$ is called $q$-imperfect if $\texttt{length}(T) = m + q$.
Phylogeny $T$ is *perfect* if $\texttt{length}(T) = m$.

Imperfection $q \Leftrightarrow q$ *recurrent* mutations

# Example

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Individual 1: | 0 | 0 | 0 | 0 |
| Individual 2: | 1 | 0 | 1 | 0 |
| Individual 3: | 1 | 0 | 0 | 0 |
| Individual 4: | 1 | 1 | 0 | 1 |
| Individual 5: | 0 | 1 | 0 | 1 |



1–imperfect

# Outline

1. Motivation
2. Phylogeny Reconstruction
   - Definitions
   - Imperfect Phylogeny Reconstruction
   - Extensions
   - Empirical Results
3. Population Substructure
   - Pure Populations
   - Admixture

# Problem Definition

- Input: $n \times m$ $\{0,1\}$-matrix $I$
- Output: phylogeny $T$ connecting all $n$ taxa of $I$
- Objective: minimize $\texttt{length}(T)$
- NP-complete, Steiner Minimum Tree over hypercubes
- Traditional approaches: Hill-climbing heuristics, brute-force

# Problem Definition

- Input: $n \times m$ $\{0,1\}$-matrix $I$, parameter $q$
- Output: phylogeny $T$ connecting all $n$ taxa of $I$
- Objective: minimize $\texttt{length}(T)$
- Assumption: $\texttt{length}(T^*) \leq m + q$ where $T^*$ is the optimal tree

# Results

| State | Imperf ($q$) | Time | Work |
|---|---|---|---|
| 2 | 0 | $O(nm)$ | Gusfield 92 |
| $k$ | $q$ | $m^{O(q)}2^{O(q^2k^2)}$ | Fernandez-Baca and Lagergren 03 |
| 2 | $q$ | $O(21^q + 8^q nm^2)$ | ICALP 06, TCBB 07 |

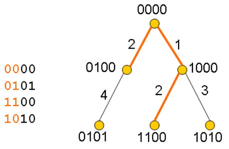
*Fixed Parameter Tractability*

Other: many heuristics Nearest-neighbor, Tree bisection and reconnection etc

# Imperfection

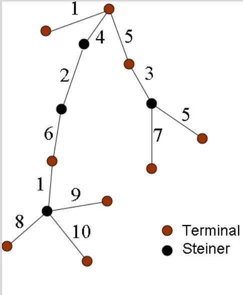
- `imperfect`(I) $=_{def}$ `imperfect`($T^*$) where $T^*$ is the optimal tree
- `imperfection`: number of duplicate edge labels

## Example



# Algorithm Overview

## Example



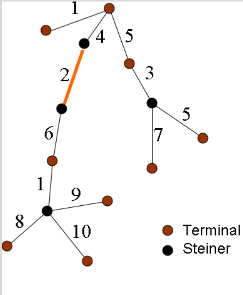
2-imperfect

## Algorithm

```
function buildTree(matrix $M$)
  1 If imperfect($M$) $= 0$ return $T^*_M$
  2 ‘Guess’ site $j$ that mutates exactly once
  3 ‘Guess’ adjacent vertices $u, v$
  4 Partition $M$ into $M0, M1$ using $j$
  5 Return buildTree($M0$) $\cup$
    buildTree($M1$) $\cup \{(u, v)\}$
```

# Algorithm Overview

## Example



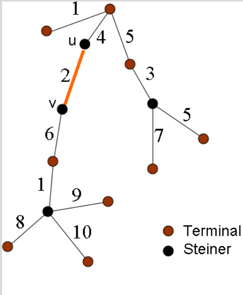
2-imperfect

## Algorithm

```
function buildTree(matrix $M$)
  1. If imperfect($M$) $= 0$ return $T^*_M$
  2. 'Guess' site $j$ that mutates exactly once
  3. 'Guess' adjacent vertices $u, v$
  4. Partition $M$ into $M0, M1$ using $j$
  5. Return buildTree($M0$) $\cup$
     buildTree($M1$) $\cup \{(u, v)\}$
```

# Algorithm Overview

## Example



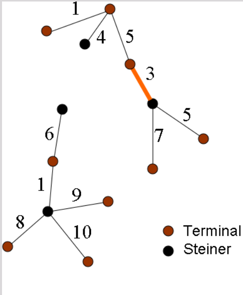
2-imperfect

## Algorithm

```
function buildTree(matrix $M$)
  1. If imperfect($M$) $= 0$ return $T^*_M$
  2. 'Guess' site $j$ that mutates exactly once
  3. 'Guess' adjacent vertices $u, v$
  4. Partition $M$ into $M0, M1$ using $j$
  5. Return buildTree($M0$) $\cup$
     buildTree($M1$) $\cup \{(u, v)\}$
```

# Algorithm Overview

## Example



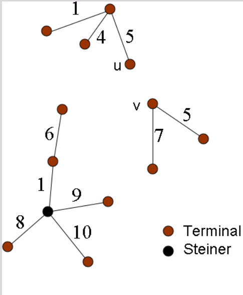
2-imperfect

## Algorithm

```
function buildTree(matrix $M$)
    1. If imperfect($M$) $= 0$ return $T^*_M$
    2. 'Guess' site $j$ that mutates exactly once
    3. 'Guess' adjacent vertices $u, v$
    4. Partition $M$ into $M0, M1$ using $j$
    5. Return buildTree($M0$) $\cup$ buildTree($M1$) $\cup \{(u, v)\}$
```

# Algorithm Overview

## Example



2-imperfect

## Algorithm

```
function buildTree(matrix $M$)
  1. If imperfect($M$) $= 0$ return $T^*_M$
  2. 'Guess' site $j$ that mutates exactly once
  3. 'Guess' adjacent vertices $u, v$
  4. Partition $M$ into $M0, M1$ using $j$
  5. Return buildTree($M0$) $\cup$
     buildTree($M1$) $\cup$ $\{(u, v)\}$
```

# Projections: If $\texttt{imperfect}(M) = 0$ return $T_M^*$

- Let $P(i, j)$ be projection of $I$ on sites $i, j$
- $\texttt{imperfect}(I) > 0$ iff $\exists i, j$ st $|P(i, j)| = 4$
- Implication: Easy to check if Gusfield's algorithm

## Example

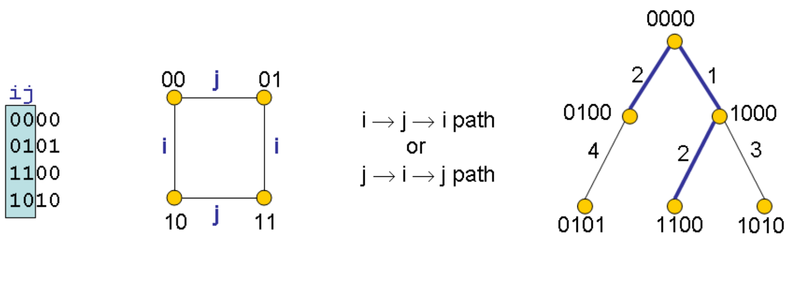
```
0000
0101
1100
1010
```

- $P(1, 2) = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$
- $P(3, 4) = \{(0, 0), (0, 1), (1, 0)\}$

# Projections: If $\texttt{imperfect}(M) = 0$ return $T^*_M$

- Sites $i, j$ *conflict* if $|P(i,j)| = 4$
- Idea: if $i, j$ conflict then $T^*$ contains $i \rightarrow j \rightarrow i$ or $j \rightarrow i \rightarrow j$ path

## Example



```
ij
0000
0101
1100
1010
```

i → j → i path
or
j → i → j path

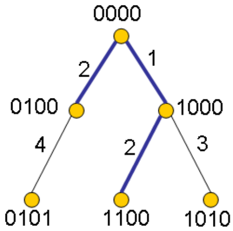
# ‘Guess’ site $j$ that mutates exactly once

- $K$: set of sites that conflict
- If $|K| \geq 2q$ then guess $j \leftarrow_{u.a.r} K$
- $\Pr[j$ occurs exactly once in $T^*] \geq 0.5$ (correct guess)

## Example

```
0000
0101
1100
1010
```

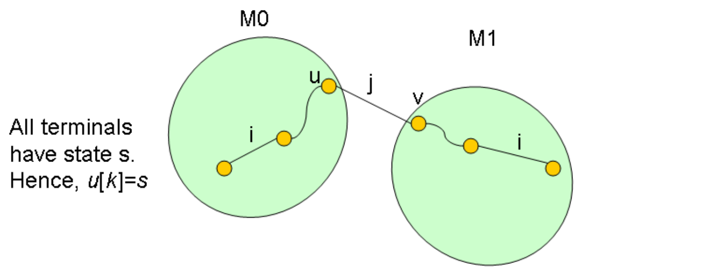
$K = \{1, 2\}, j = 1$



# ‘Guess’ adjacent vertices $u, v$

If all vertices in $M0$ contain state $s$ on site $k$ then $u[k] = s$ therefore $v[k] = s$

## Example



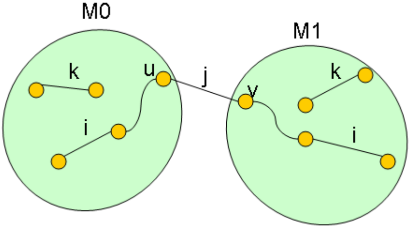
All terminals have state s. Hence, *u*[*k*]=*s*

# 'Guess' adjacent vertices $u, v$

- If both $M0$ and $M1$ contain both states on site $k$ then guess $u[k] \leftarrow_{u.a.r} \{0, 1\}$ (Pr[correct guess] $= 0.5$)
- If $t$ guesses performed then $\texttt{imperfect}(M0) + \texttt{imperfect}(M1) \leq \texttt{imperfect}(M)$ - $t$

## Example

M0 contains terminals $v_1$, $v_2$ St $v_1[k] \neq v_2[k]$



M1 contains terminals $v_3$, $v_4$ St $v_3[k] \neq v_4[k]$

## Analysis

- Each guess has success probability 0.5
- Each guess reduces imperfection by at least 1
- `imperfect`$(I) = q$
- Pr[algorithm finds $T_I^*] \geq 0.25^q$
- Recap: Running time: exponential in $q$ polynomial in $n, m$
- Can be derandomized by enumeration

# Outline

1. Motivation
2. Phylogeny Reconstruction
   - Definitions
   - Imperfect Phylogeny Reconstruction
   - Extensions
   - Empirical Results
3. Population Substructure
   - Pure Populations
   - Admixture

# Results

Genotypes: Conflated combinations of $\{0,1\}^m$ sequences

| Imperf ($q$) | Time | Work |
|---|---|---|
| 0 | $O(nm\alpha(n,m))$ | Gusfield 2003 |
| 0 | $O(nm^2)$ | Eskin, Halperin and Karp 2004 |
| 0 | $O(nm)$ | Ding, Filkov and Gusfield 2005 |
| 1 | $O(nm^3)$ | Song, Wu and Gusfield 2005 |
| $q$, 1 site | $O(nm^{q+2})$ | Satya et al. 2006 |
| $q$ | $nm^{O(q)}$ | Sridhar, Blelloch, Ravi, Schwartz 2006 |

# Outline

1. Motivation

2. Phylogeny Reconstruction
   - Definitions
   - Imperfect Phylogeny Reconstruction
   - Extensions
   - Empirical Results

3. Population Substructure
   - Pure Populations
   - Admixture

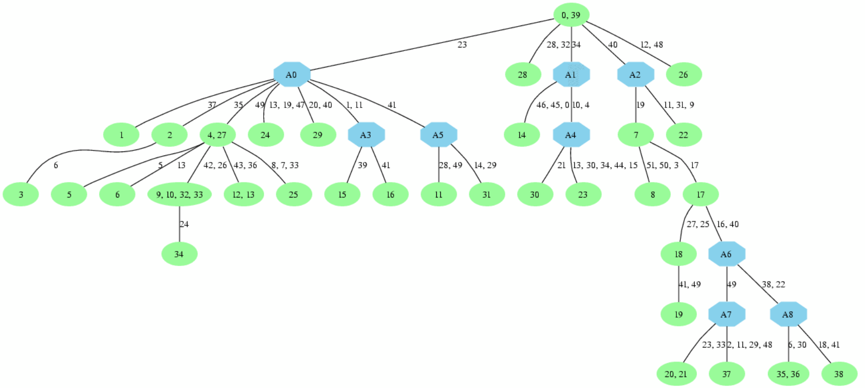
# Phylogenies

Practical ILP based algorithm (S, Lam, Blelloch, Ravi, Schwartz 07)

| | | | time(secs) | | | |
|---|---|---|---|---|---|---|
| Data Set | input | $q$ | FPT | ILP | pars | penny |
| human Y | 150 × 49 | 1 | 0.02 | 0.02 | 2.55 | — |
| bacterial | 17 × 1510 | 7 | 4.61 | 0.08 | 0.06 | — |
| chimp mtDNA | 24 × 1041 | 2 | 0.14 | 0.08 | 2.63 | — |
| chimp Y | 15 × 98 | 1 | 0.02 | 0.02 | 0.03 | — |
| human mtDNA | 40 × 52 | 21 | — | 13.39 | 11.24 | — |
| human mtDNA | 395 × 830 | 14 | — | 53.4 | 712.95 | — |
| human mtDNA | 13 × 390 | 6 | 9.75 | 0.02 | 0.41 | 1160.97 |
| human mtDNA | 33 × 405 | 4 | 1.36 | 0.09 | 0.59 | — |

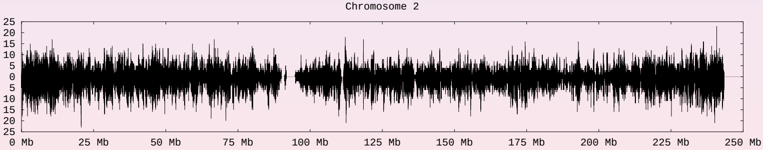
# Webserver: Phylogeny Reconstruction

- Buddhists and Muslims of Ladakh: 52 mtDNA SNPs



# Genome-Wide Scan (Sridhar and Schwartz 2008)

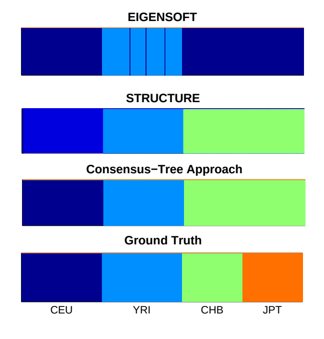
- Sliding window across whole genome
- Construct phylogeny for each window
- Chromosome 2 imperfection on Central Europeans (top) and Africans (bottom)



$x$-axis: genomic position, $y$-axis: imperfection

# Recent Work

- Tsai et al. used our method to cluster sub-populations
- CEU: Central Europeans, YRI: Yoruba Africans,
  CHB: Han Chinese, JPT: Japanese from Tokyo



# Empirical Results

- Solved millions of problem instances spanning whole genome
- Provided fine-scale mutation rates across genome
- Software used hundreds of times online
- Exciting new avenues
    - Find sub-populations
    - Find rapidly evolving regions of the genome

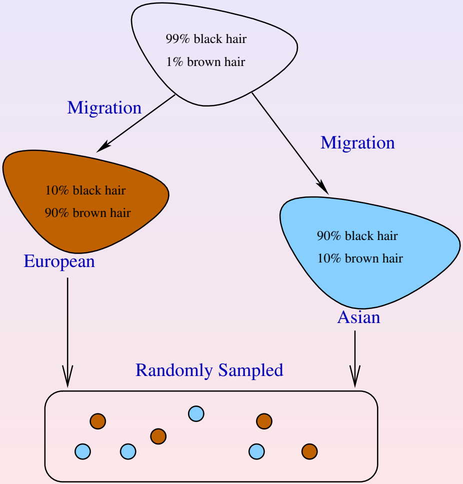
# Outline

1. Motivation
2. Phylogeny Reconstruction
   - Definitions
   - Imperfect Phylogeny Reconstruction
   - Extensions
   - Empirical Results
3. Population Substructure
   - Pure Populations
   - Admixture

# Outline

1. Motivation
2. Phylogeny Reconstruction
   - Definitions
   - Imperfect Phylogeny Reconstruction
   - Extensions
   - Empirical Results
3. Population Substructure
   - Pure Populations
   - Admixture

# Problem Overview



# Example

- Two populations: ‘Asians’ ($p$) and ‘Europeans’ ($q$)
- For simplicity, consider two SNPs with state 1 probabilities:
  - $(p_1, p_2) = (0.4, 0.1)$ (Asians)
  - $(q_1, q_2) = (0.3, 0.5)$ (Europeans)
- Randomly sampled European, SNP 2 has state 1: $0.5$

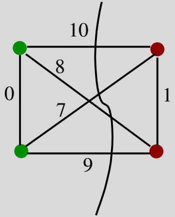
# Problem Definition

- Input: $n \times m$-matrix $G$
- Output: classification $\hat{\theta} : \{1, \ldots, n\} \to \{0, 1\}$
- Errors: $\min \sum_{i=1}^{n} |\theta(i) - \hat{\theta}(i)|$
  $\theta$ is the correct classification
- Want to minimize errors (no training data)

# Graph Based (RECOMB 2007)

- Graph $G(V, E)$
  - Each vertex represents an individual
  - Edge distance captures genomic distance
- Perform max-cut on $G$

## Example



# Mathematical Properties

## Distance function properties

- Expected intra-distance$= 0$
- Expected inter-distance$= 2d^2$, where $d$ is the $L_2$ distance between the two populations

## Convergence

- When $m = \Omega(\frac{\log n}{\gamma^2})$ where

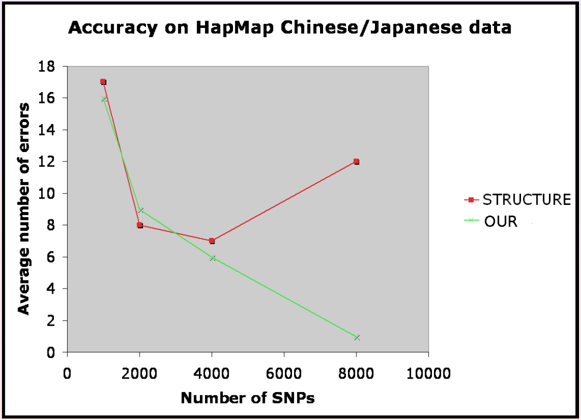
  $\gamma$: Expected (over SNPs) $L_2^2$ distance between populations
  $n$: number of individuals
  $m$: number of SNPs.

  - max-cut is the correct partition
  - max-cut can be found efficiently (polynomial time)

# Accuracy in practice (RECOMB 2007)

89 individuals: 45 Chinese, 44 Japanese

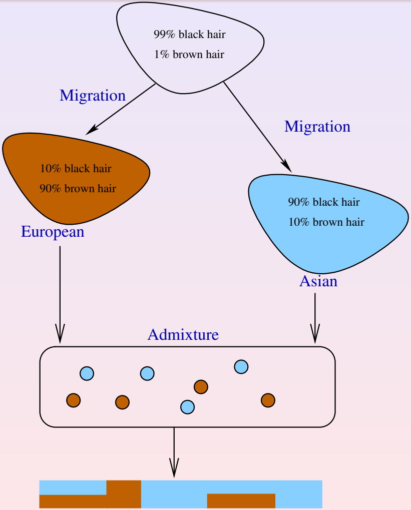
`structure`: Markov Chain Monte Carlo based (cited 1000+ times)



# Outline

1. Motivation
2. Phylogeny Reconstruction
   - Definitions
   - Imperfect Phylogeny Reconstruction
   - Extensions
   - Empirical Results
3. Population Substructure
   - Pure Populations
   - Admixture

# Admixture Example



# Problem Definition

- Input: $n \times m$ matrix $G$
- Output: classification
  $\hat{\theta} : \{1, \ldots, n\} \times \{1, \ldots, m\} \to \{0, 0.5, 1\}$
- Errors: $\theta(i,j) \neq \hat{\theta}(i,j)$
  $\theta$ is the correct classification
- Ancestry of every locus of every individual

# High Level Idea

- Sliding window of length $w$
- Predict ancestry $\hat{\theta} : \{0, 0.5, 1\}$ for local region
- Combine local predictions
- Software downloaded and used by hundreds of labs including Cornell, UCSF, Scripps, Harvard medical school etc.
- *American Journal of Human Genetics 2008*

# Recap of Contributions

- Finding polymorphisms: copy number variation in mouse
- Phylogeny Reconstruction
  - Fixed parameter tractability for haplotypes
  - Polynomial time (when $q$ is fixed) for genotypes
  - Integer Linear Programming for general problem
  - Genome-wide analysis of phylogenies
- Population Substructure
  - Pure populations: Poly-time, provably correct; outperforms other methods in accuracy (closely related populations) and run-time
  - Admixed populations: outperforms other methods in accuracy (well-separated ancestral populations) and significantly faster

# Conclusions and Future Work

- Finding variation
  - Finding copy number changes, reversals, deletions
- Analysis of Variation
  - Phylogenies over sub-populations
  - Richer population models
  - Selection
- Disease Association Tests
- Direct to consumer genotyping
  - No longer controlled studies
  - Identifying relationships: cousins, ancestry