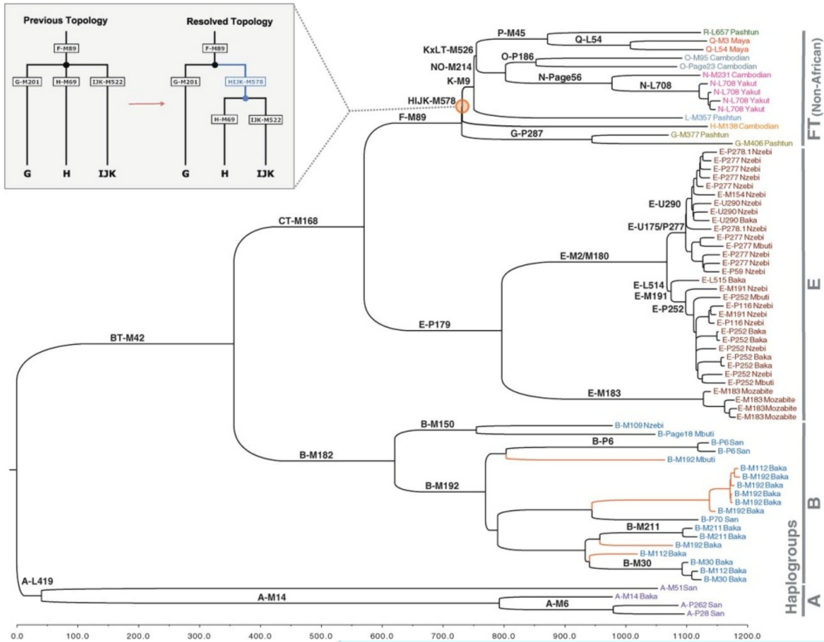
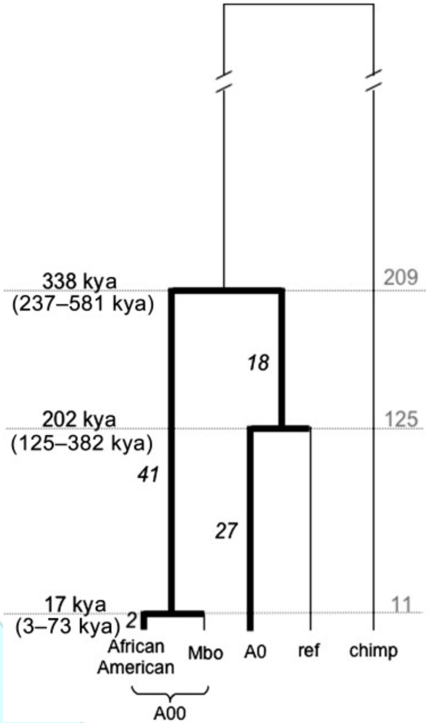
Human Genome Diversity, Coalescence & Haplotypes

Coalescence

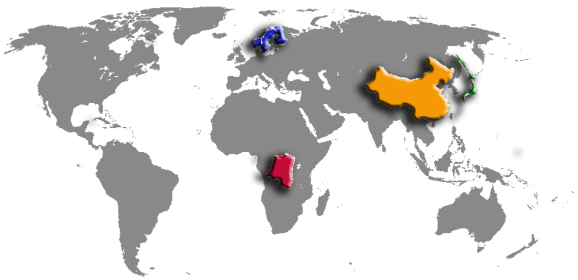


some coalescence



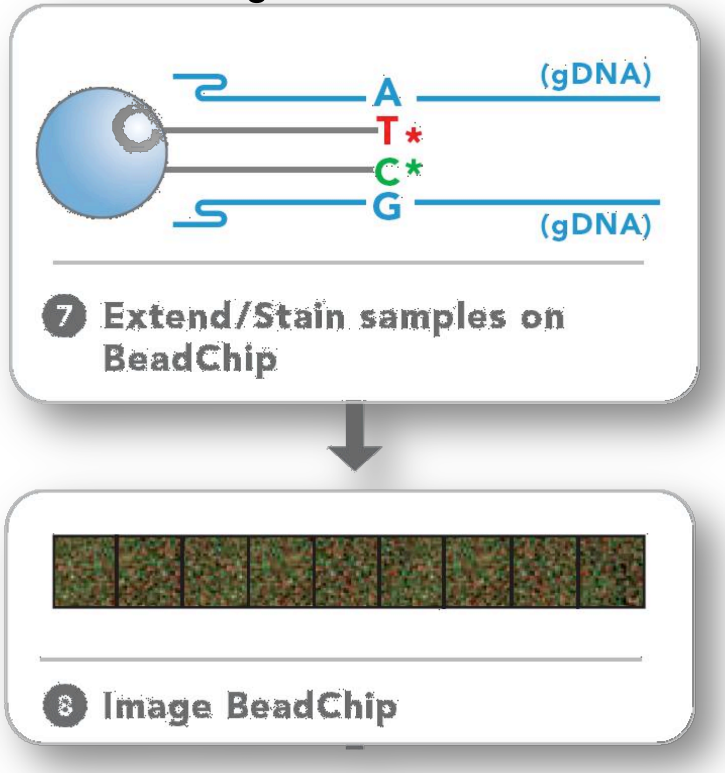
The HapMap Project

ASW	African ancestry in Southwest USA	90
CEU	Northern and Western Europeans (Utah)	180
CHB	Han Chinese in Beijing, China	90
CHD	Chinese in Metropolitan Denver	100
GIH	Gujarati Indians in Houston, Texas	100
JPT	Japanese in Tokyo, Japan	91
LWK	Luhya in Webuye, Kenya	100
MXL	Mexican ancestry in Los Angeles	90
MKK	Maasai in Kinyawa, Kenya	180
TSI	Toscani in Italia	100
YRI	Yoruba in Ibadan, Nigeria	100



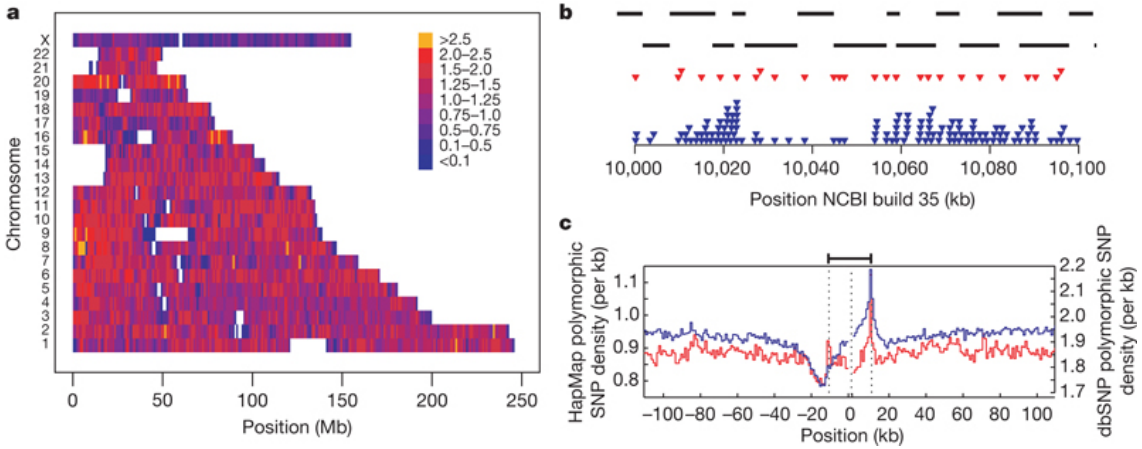
Genotying:

Probe a limited number (~1M) of known highly variable positions of the human genome



7 Extend/Stain samples on BeadChip

8 Image BeadChip



Linkage Disequilibrium & Haplotype Blocks

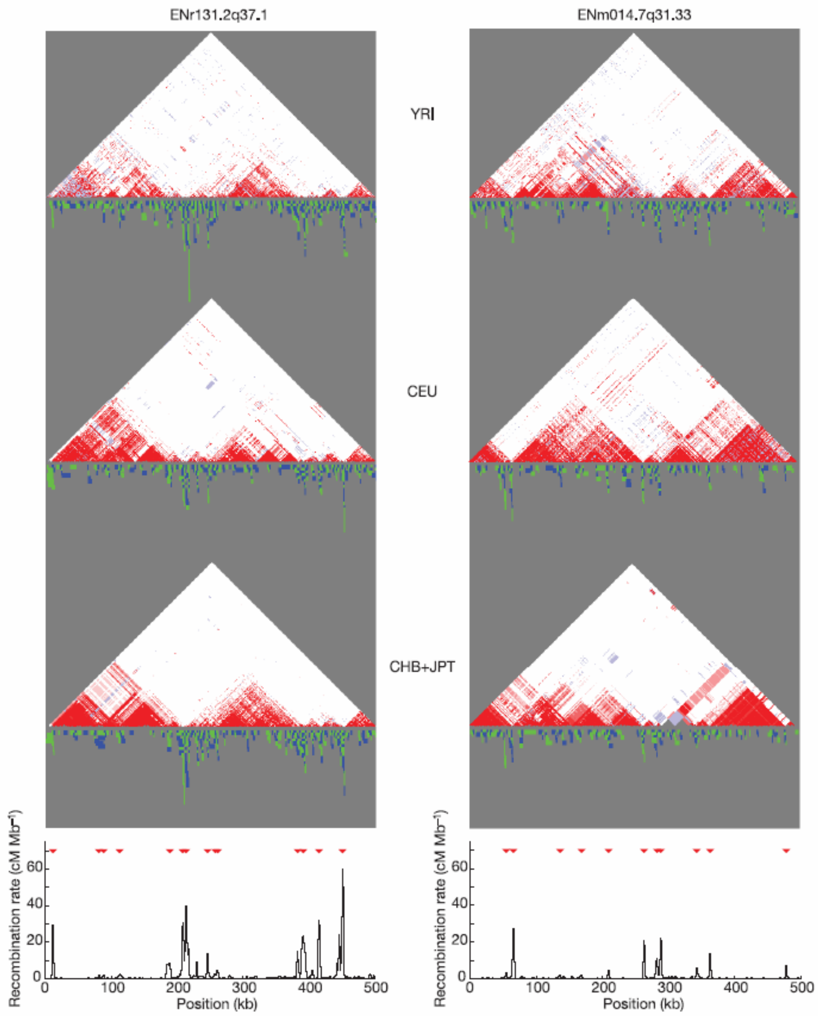
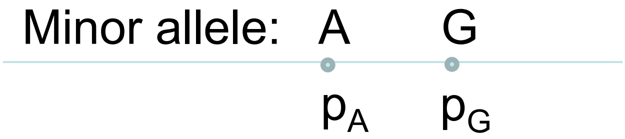


Figure 8 | **Comparison of linkage disequilibrium and recombination for two ENCODE regions.** For each region (ENr131.2q37.1 and ENm014.7q31.33), D' plots for the YRI, CEU and CHB+JPT analysis panels are shown: white, $D' < 1$ and LOD < 2; blue, $D' = 1$ and LOD < 2; pink, $D' < 1$ and LOD ≥ 2; red, $D' = 1$ and LOD ≥ 2. Below each of these plots is shown the intervals where distinct obligate recombination events must have occurred (blue and green indicate adjacent intervals). Stacked intervals represent regions where there are multiple recombination events in the sample history. The bottom plot shows estimated recombination rates, with hotspots shown as red triangles[46].



Minor allele: A G

p_A p_G

Linkage Disequilibrium (LD):

$$D = P(A \text{ and } G) - p_A p_G$$

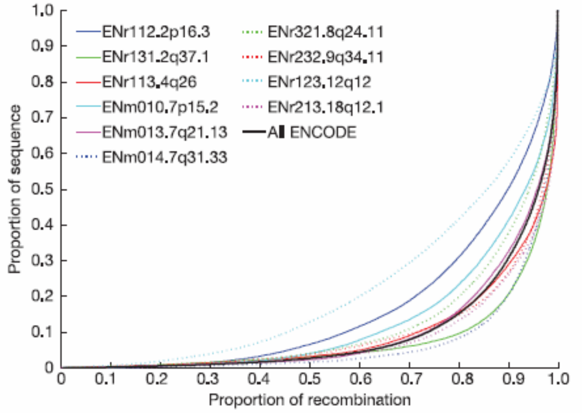
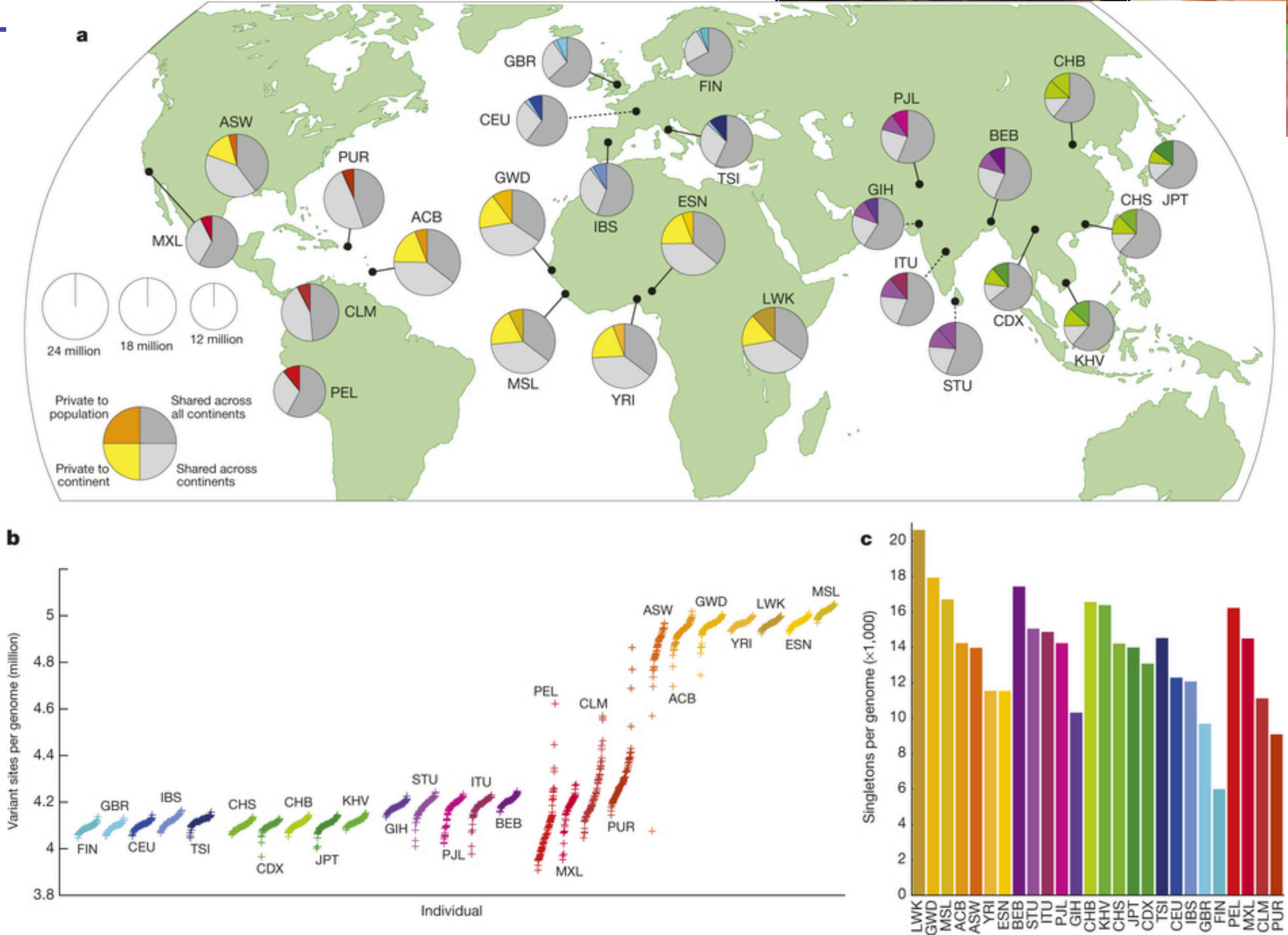
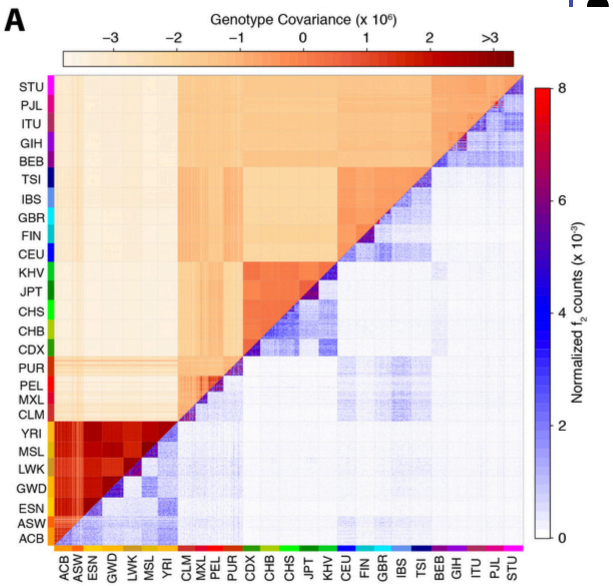


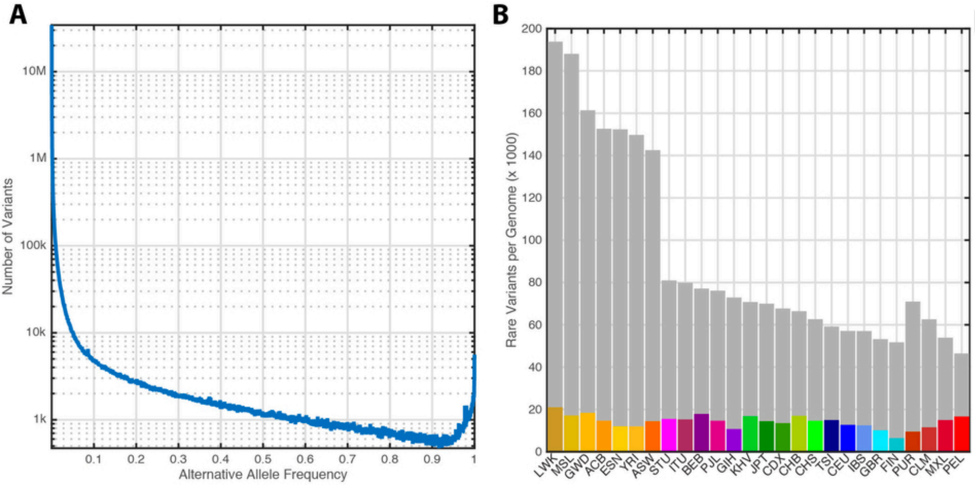
Figure 9 | **The distribution of recombination events over the ENCODE regions.** Proportion of sequence containing a given fraction of all recombination for the ten ENCODE regions (coloured lines) and combined (black line). For each line, SNP intervals are placed in decreasing order of estimated recombination rate[46], combined across analysis panels, and the cumulative recombination fraction is plotted against the cumulative proportion of sequence. If recombination rates were constant, each line would lie exactly along the diagonal, and so lines further to the right reveal the fraction of regions where recombination is more strongly locally concentrated.

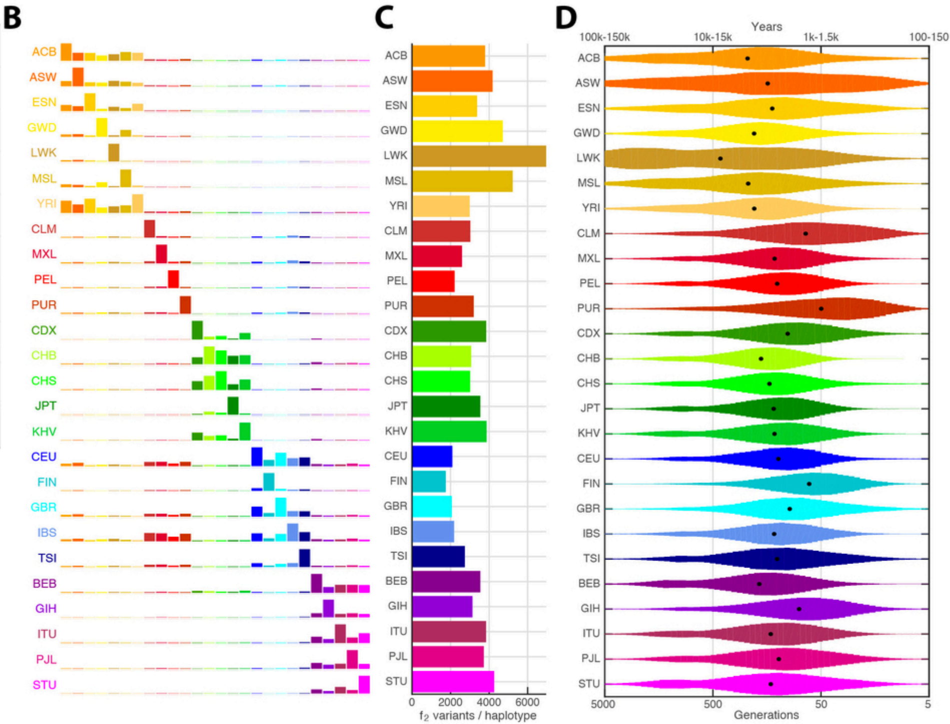
Population Sequencing – 1000 Genomes Project



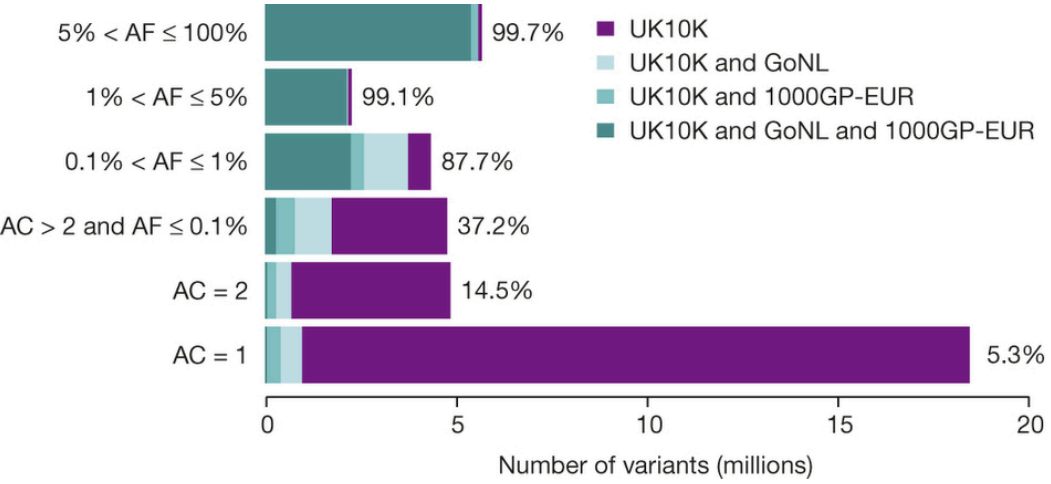
Population Sequencing – 1000 Genomes Project

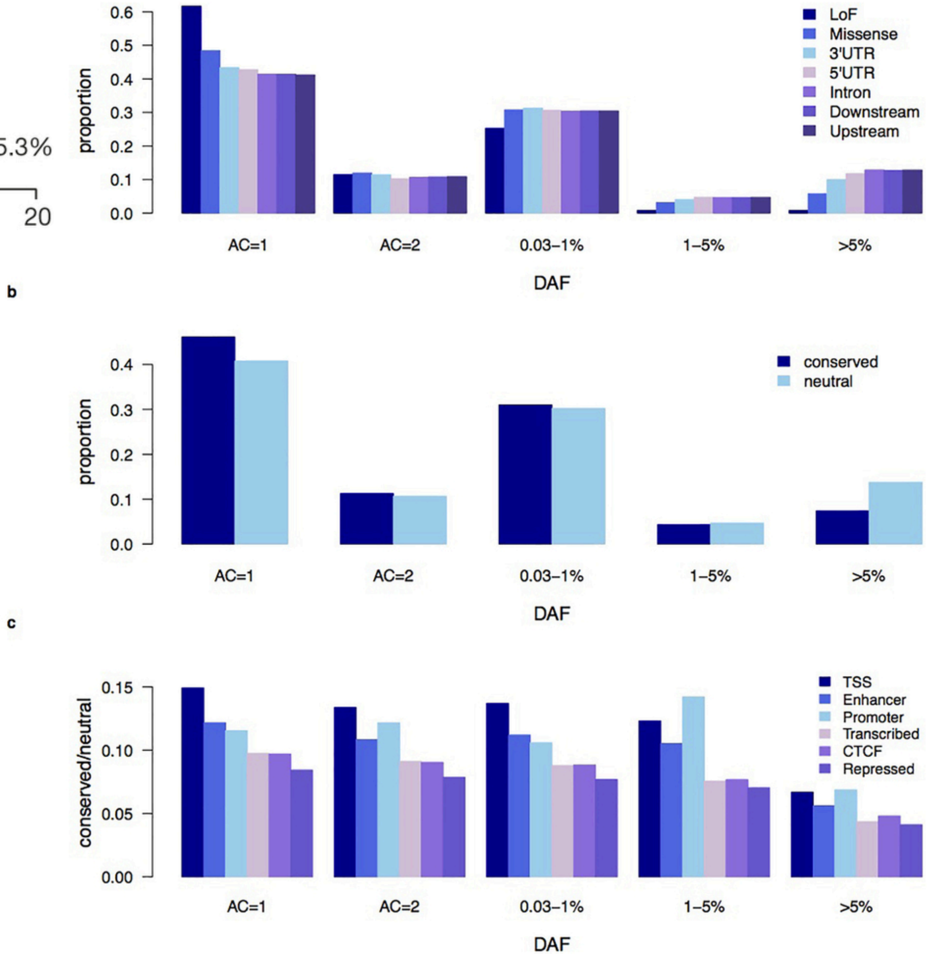




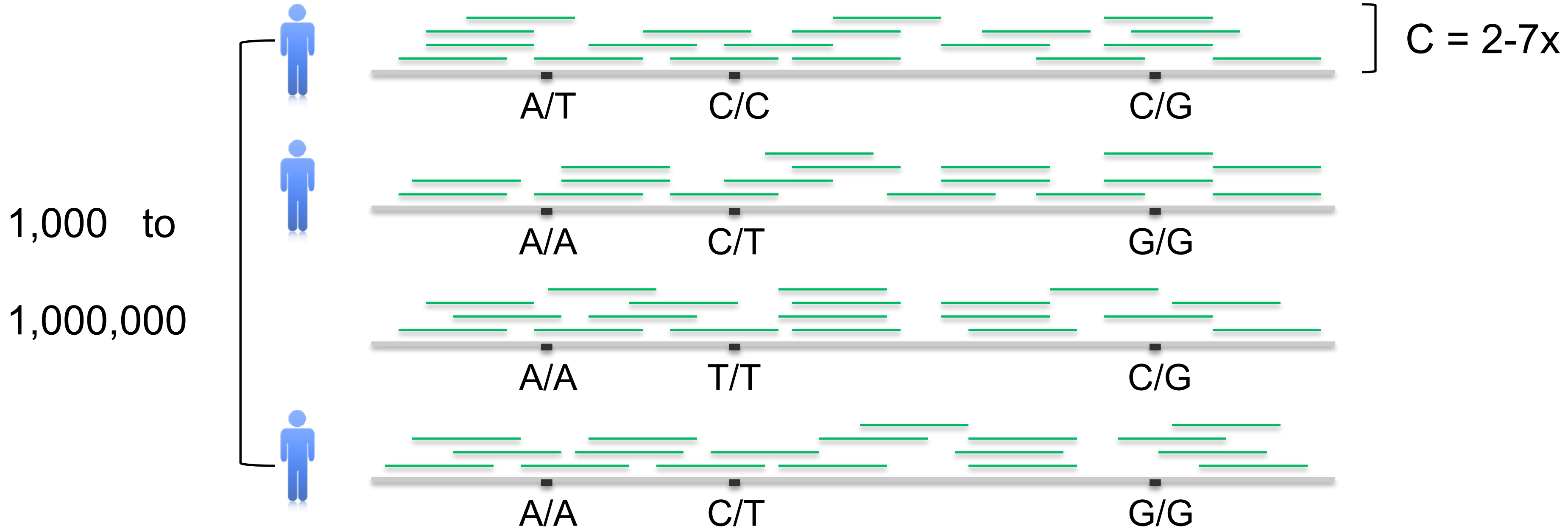


Population Sequencing – UK10K

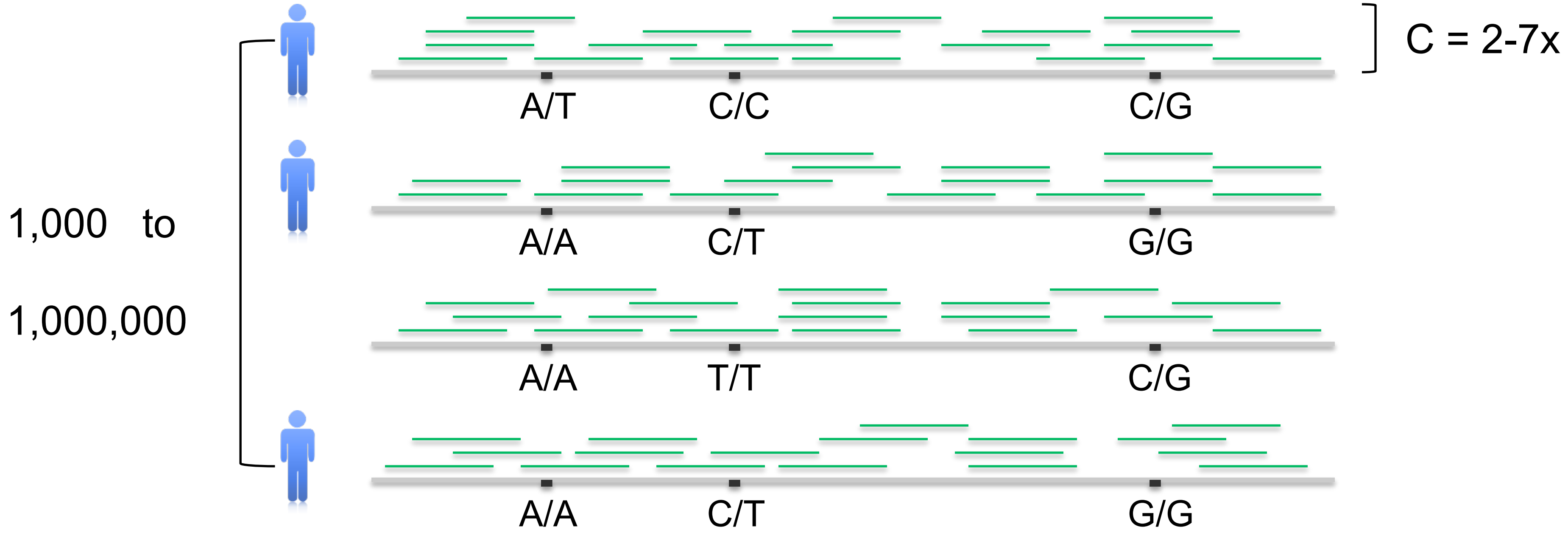




Population Sequencing



Population Sequencing



$G_1, \ldots, G_N$; $G_i = g_{i1} \ldots g_{in}$; $g_{ij} \in \{0, 1, 2\}$

$P_1, \ldots, P_N$; P_i : $[\, p_{ijg} = \text{Prob}(g_{ij} = g \mid \text{data}) \,]$

Population Sequencing

When C is high (>30x),

Prob(g_{ij} = g | data) ~

Prob(g_{ij} = g | reads mapping on (i, j))

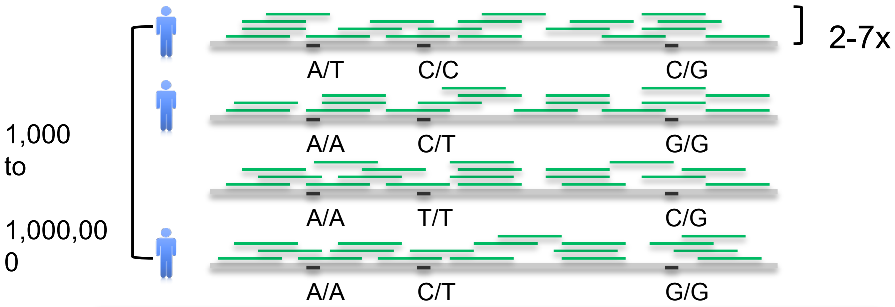
fast & easy

When C is low,

Prob(g_{ij} = g | data) needs to leverage LD:

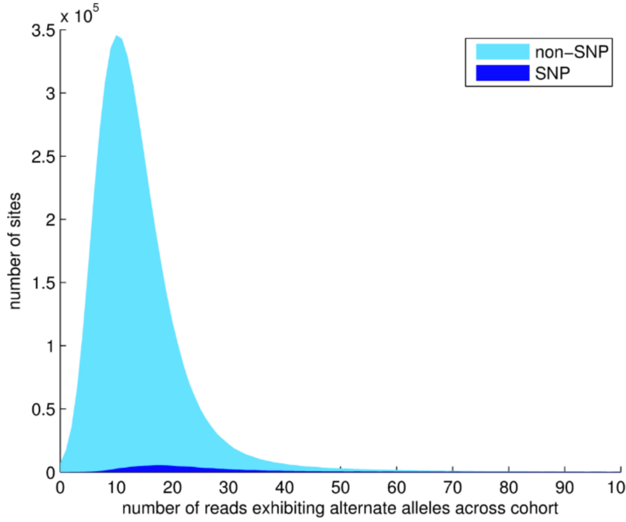
positions j’ ≠ j in all individuals

in principle, intractable



$G_1, \ldots, G_N$; $G_i = g_{i1} \ldots g_{in}$; $g_{ij} \in \{0, 1, 2\}$

$P_1, \ldots, P_N$; P_i : [p_{ijg} = Prob(g_{ij} = g | data)]



1000 Genomes Project, 2535 individuals, 7x sequencing

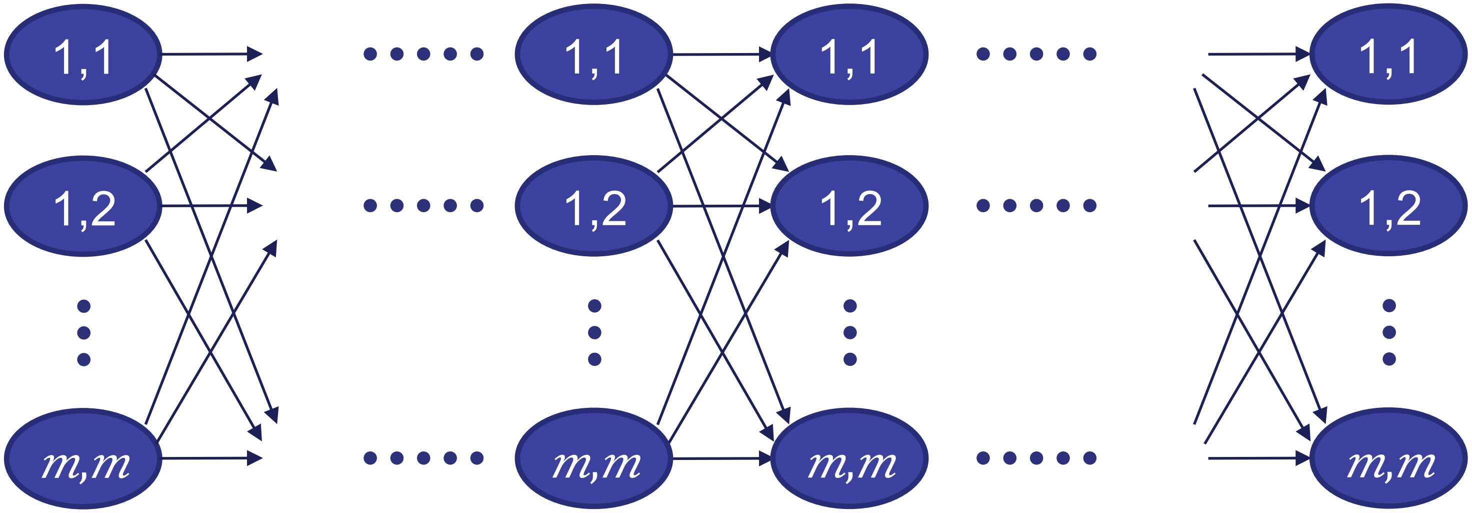
HMM-based models

- Li and Stephens 2003

Given m reference haplotypes, and a target sample,

Find the most likely path of haplotype pairs

m^2 states, m^4 transitions per position

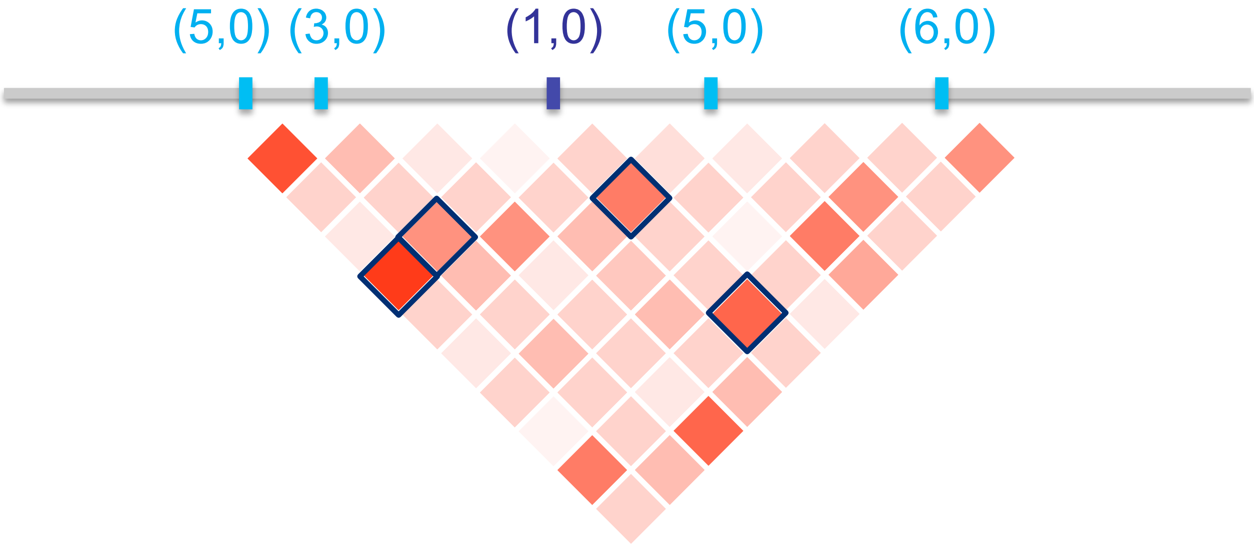


Informative Neighbors

▮ target SNP

▮ *k*-”nearest” neighbors

in terms of linkage disequilibrium



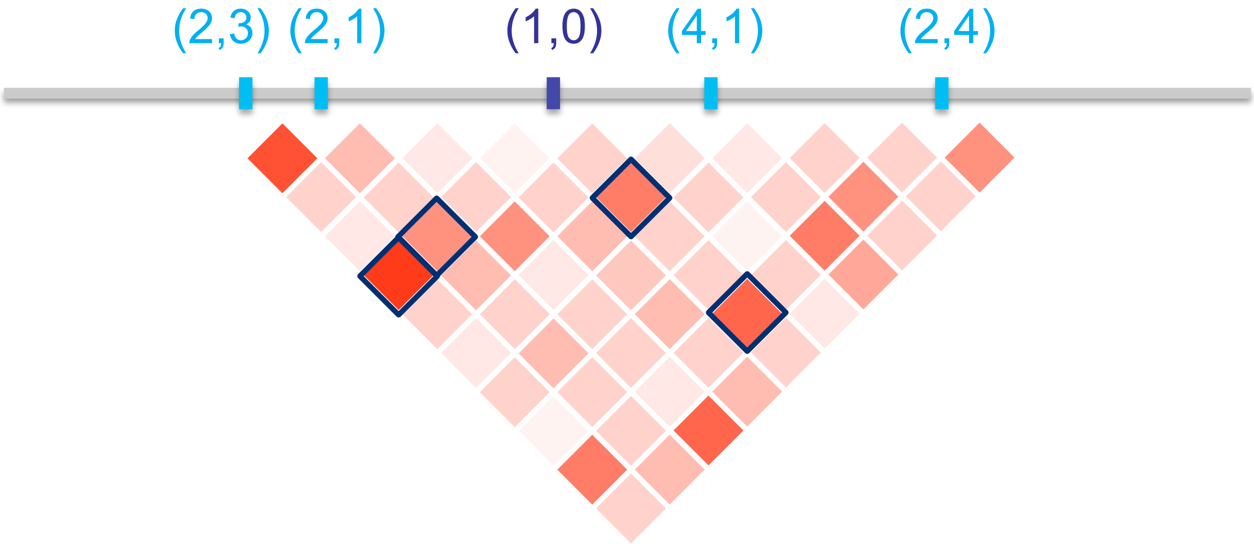
$$(R_{\text{ref}}, R_{\text{alt}}) = \Sigma_{\{\text{target, nbrs}\}}(r_{\text{ref}}, r_{\text{alt}}) = (20, 0)$$

Informative Neighbors

▮ target SNP

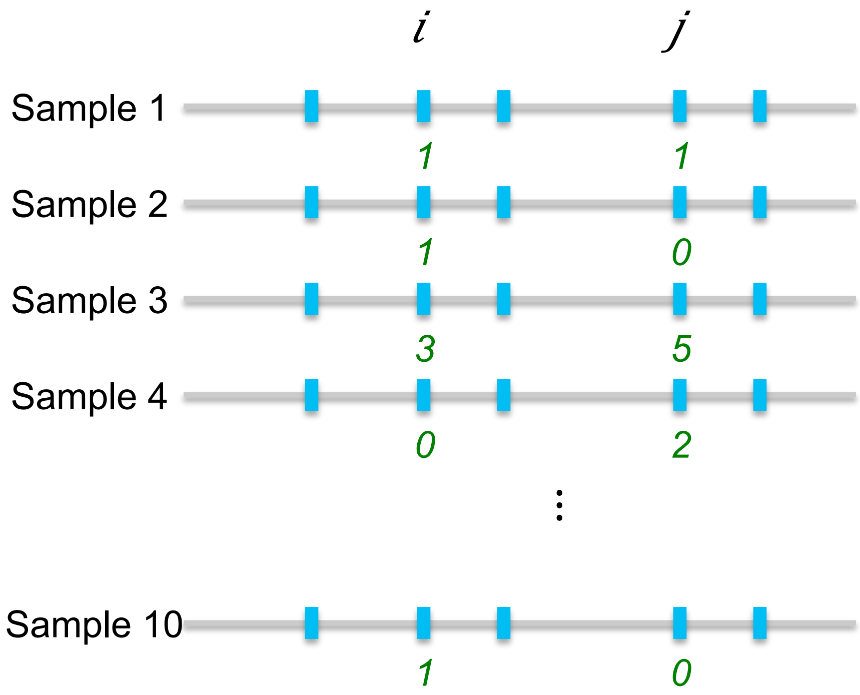
▮ *k*-”nearest” neighbors

in terms of linkage disequilibrium



$$(R_{\text{ref}}, R_{\text{alt}}) = \Sigma_{\{\text{target, nbrs}\}}(r_{\text{ref}}, r_{\text{alt}}) = (11, 9)$$

How to pick k nearest neighbors fast



Correlation Coefficient:
$r^2 = (p_{AB} - p_A p_B)^2 / p_A p_B p_a p_b$

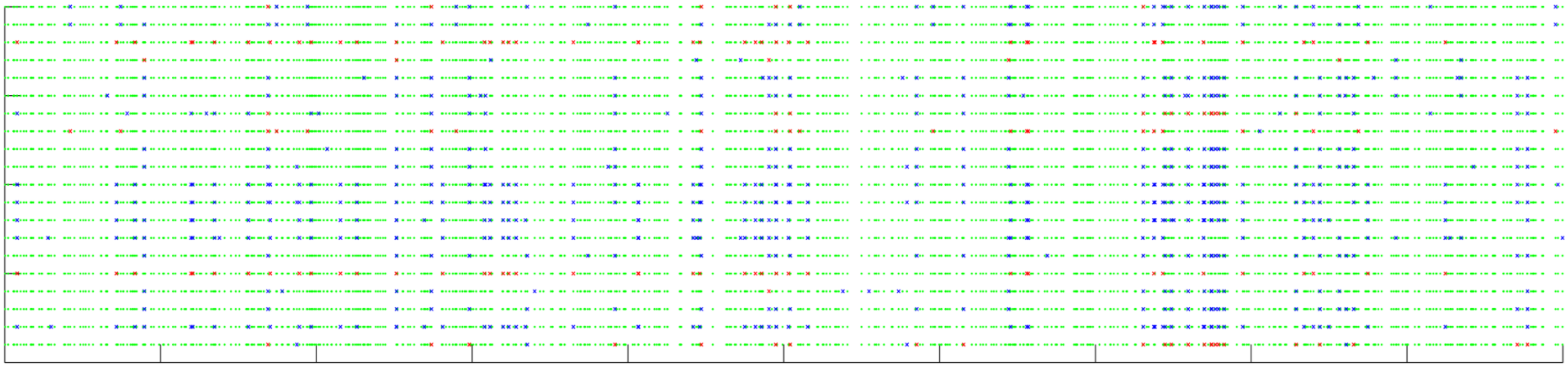
Caveat: need **genotyping**, **phasing**

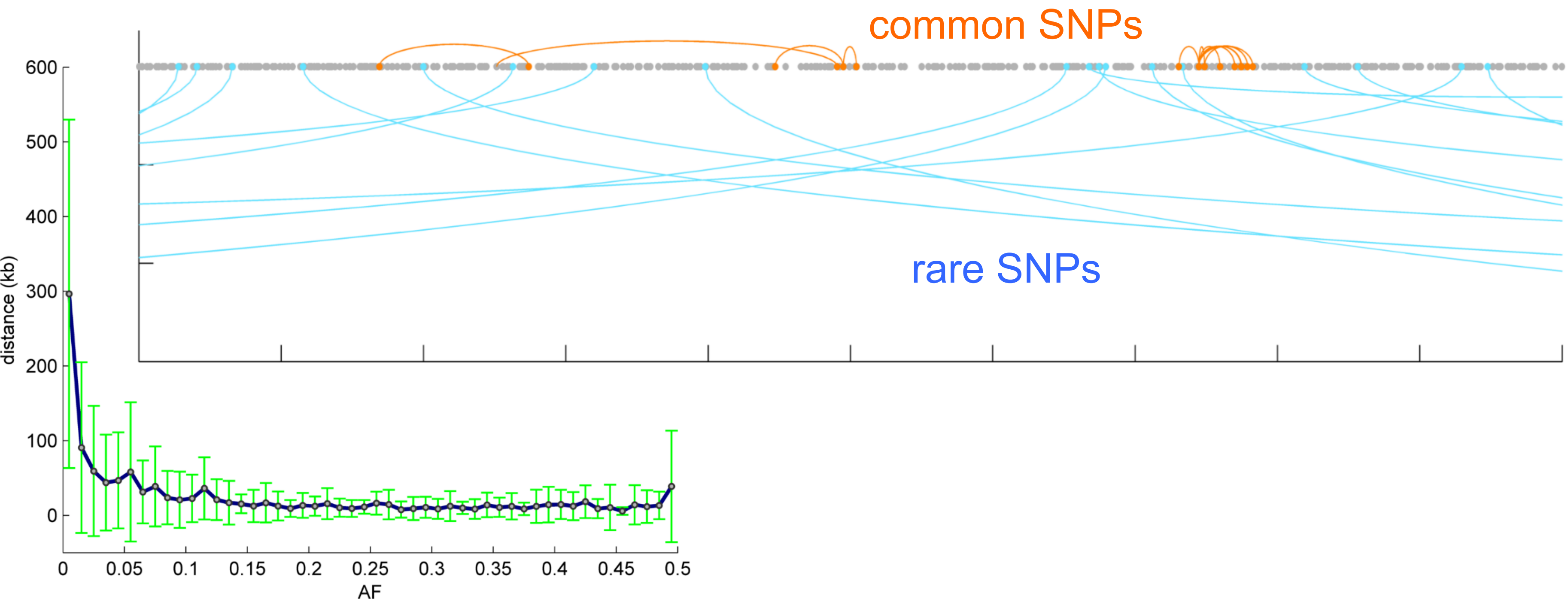
Let
S_i = { samples covering minor allele }
S_i' = { read counts of minor allele }

$S_i = \{1, 2, 3, 10\}$
$S_j = \{1, 3, 4\}$
$S_i' = \{1, 2, 3, 3, 3, 10\}$
$S_j' = \{1, 3, 3, 3, 3, 3, 4, 4\}$

$\text{Sim}_1(i, j) = (S_i \cap S_j) / (S_i \cup S_j)$
$\text{Sim}_2(i, j) = (S_i' \cap S_j') / (S_i' \cup S_j')$
$\text{Sim}_3(i, j) = ((S_i' \cap S_j') / (S_i' \cup S_j'))^2$

Genetic distance between NNs





Reveel: Variant Discovery and Imputation

Reveel:

1. Identify candidate polymorphic sites

2. Calculate k nearest neighbors
   - Jaccard indices Sim_1, Sim_2, Sim_3

3. Initialize $G^{(0)}$

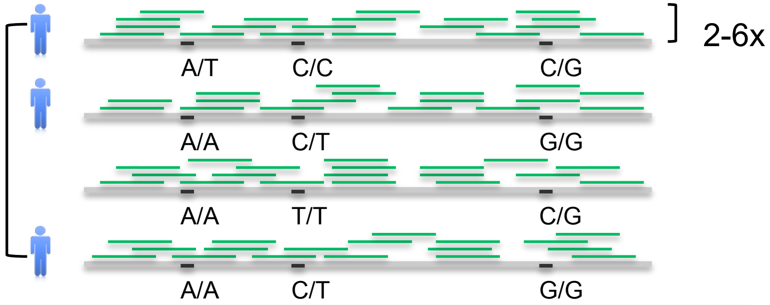
4. Summarization/Maximization
   $p^{(n+1)}_{ij\mathit{g}} = \mathrm{Prob}(g_{ij} = \mathit{g} \mid G^{(n)}, \mathrm{data})$
   $g^{(n+1)}_{ij\mathit{g}} = \mathrm{argmax}\ p^{(n+1)}_{ij\mathit{g}}$

5. Recalculate k nearest neighbors
   - Approximate Correlation Coefficient (Schaid 2004)

6. Summarization/Maximization

7. Recalculate k nearest neighbors
   - Approximate CC, Entropy

8. Summarization/Maximization



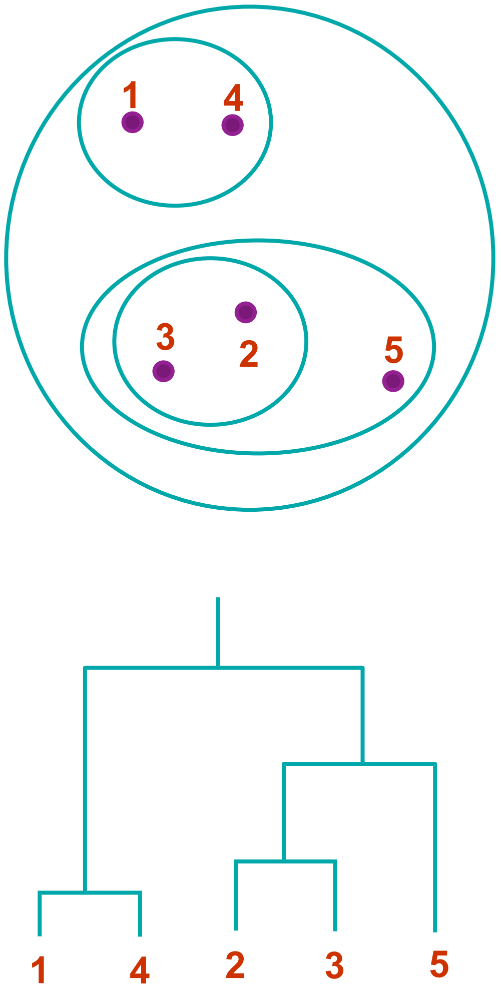
$G_1, \ldots, G_N;\quad G_i = g_{i1} \ldots g_{in};\quad g_{ij} \in \{0, 1, 2\}$

$P_1, \ldots, P_N;\quad P_i : [\, p_{ij\mathit{g}} = \mathrm{Prob}(g_{ij} = \mathit{g} \mid \mathrm{data}) \,]$

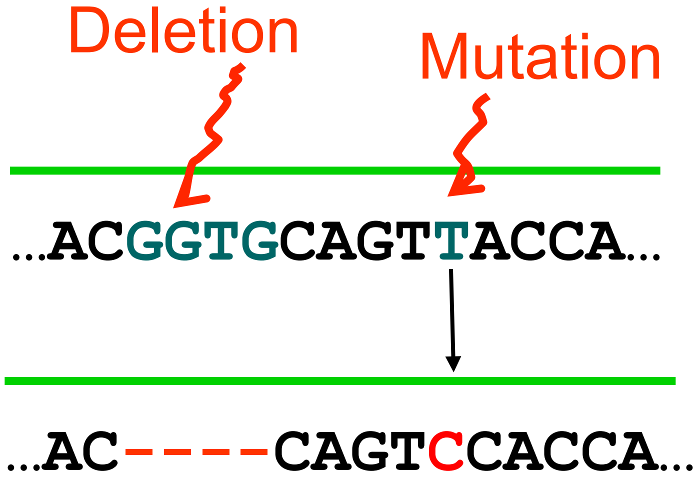
Candidate Polymorphic site

Essentially, pos'n j where some individuals have at least 2 reads with same minor allele

Molecular Evolution and Phylogenetic Tree Reconstruction



Evolution at the DNA level



Deletion

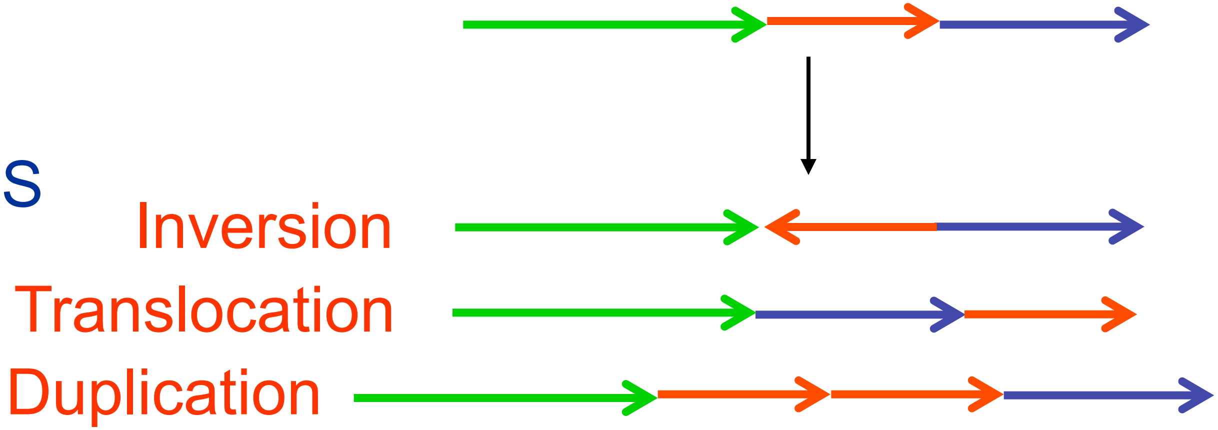
Mutation

...ACGGTGCAGTTACCA...

...AC----CAGTCCACCA...

SEQUENCE EDITS

REARRANGEMENTS



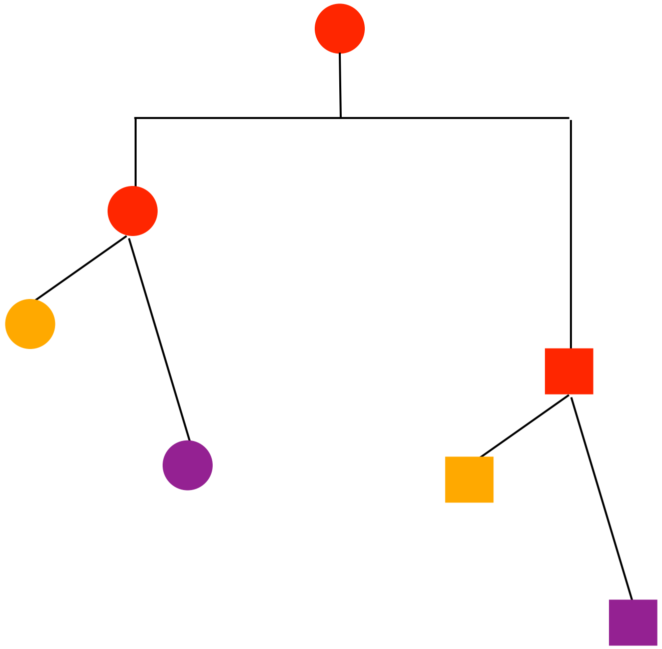
Inversion

Translocation

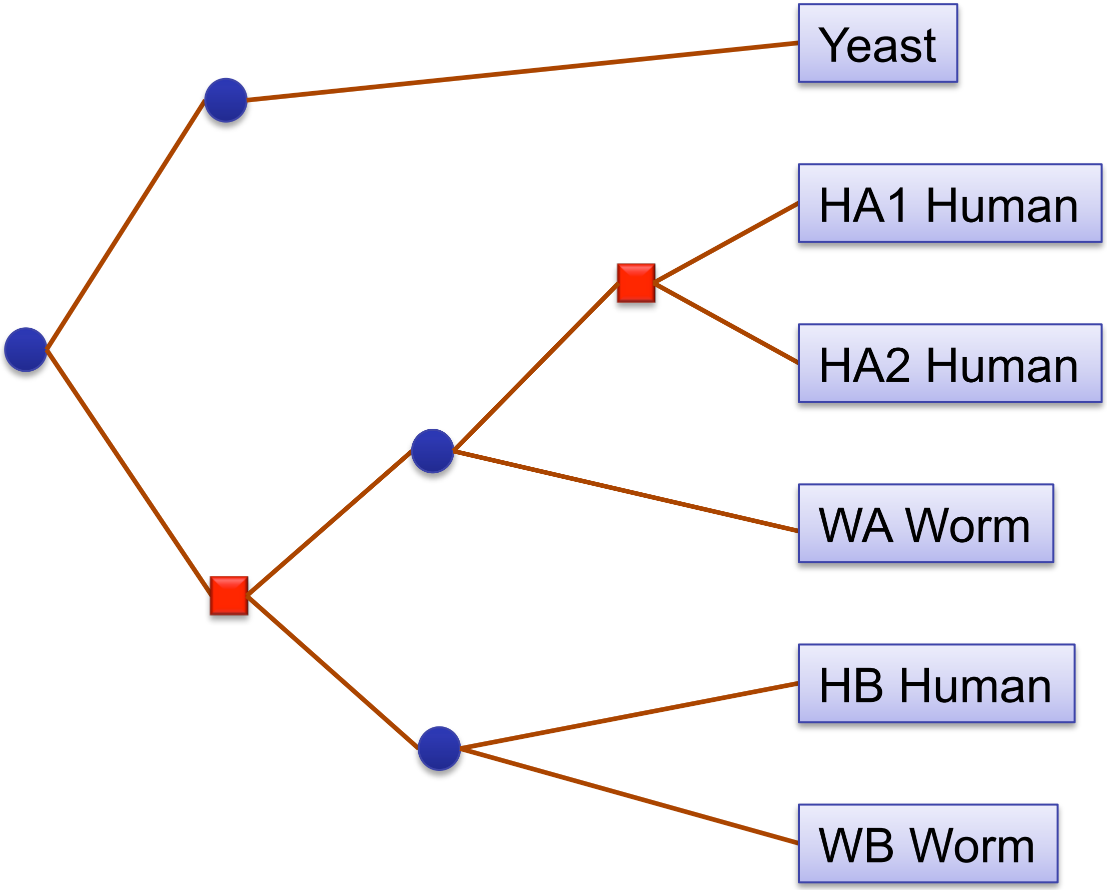
Duplication

Protein Phylogenies

- Proteins (genes) evolve by both duplication and species divergence



Orthology and Paralogy



Orthologs: *Derived by speciation*

Paralogs: *Everything else*

Orthology, Paralogy, Inparalogs, Outparalogs

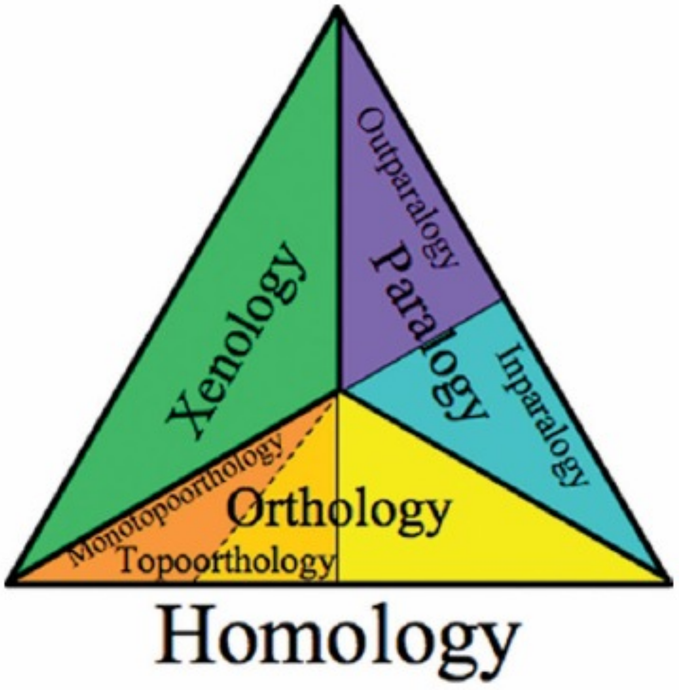


Figure 1. Refinements of homology.

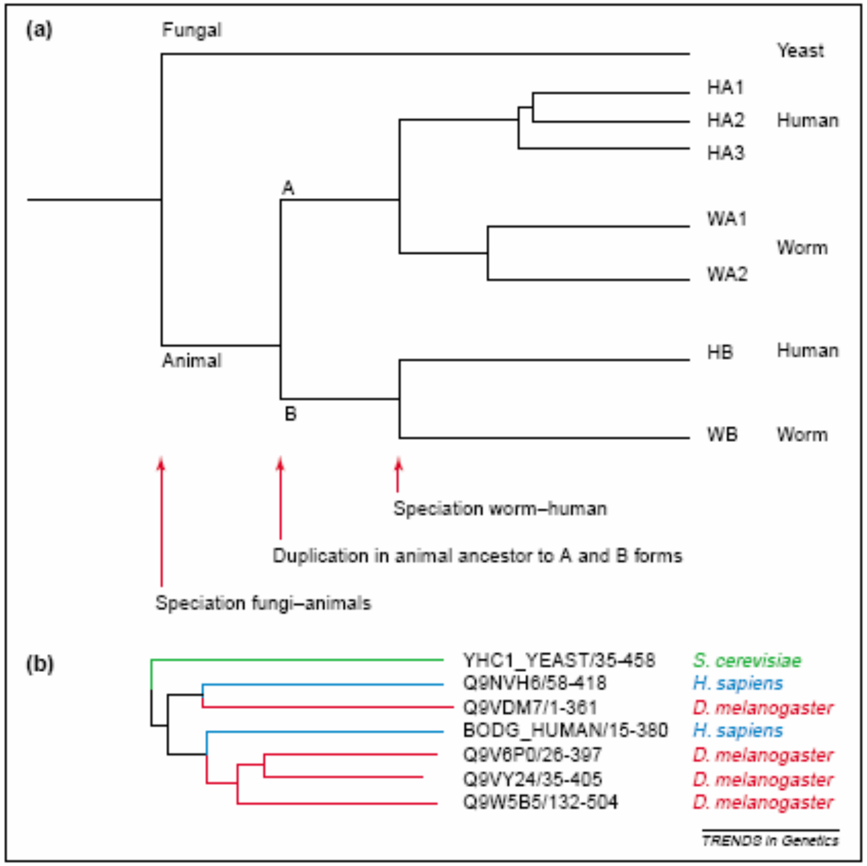
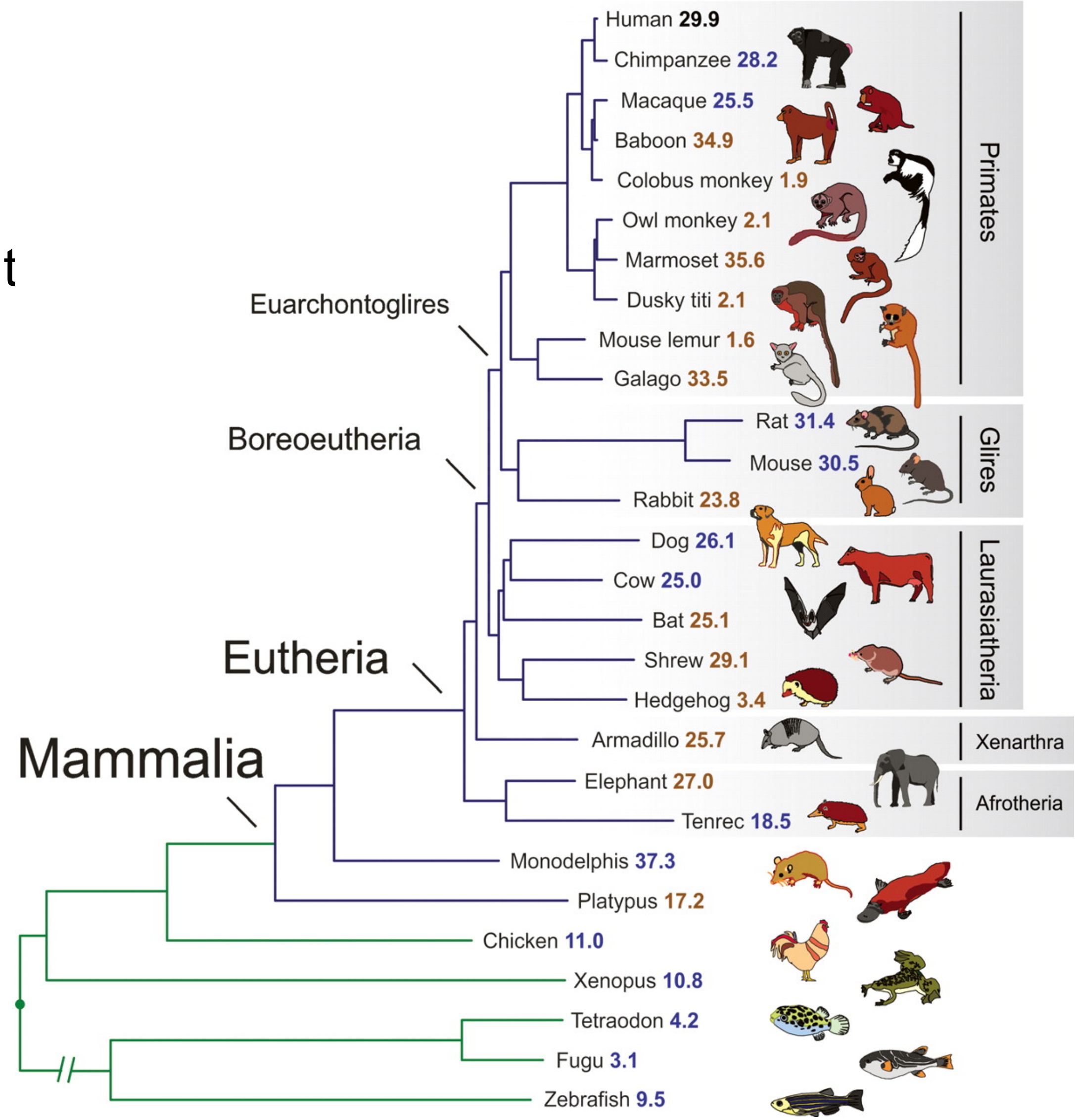


Fig. 1. The definition of inparalogs and outparalogs. (a) Consider an ancient gene inherited in the yeast, worm and human lineages. The gene was duplicated early in the animal lineage, before the human–worm split, into genes A and B. After the human–worm split, the A form was in turn duplicated independently in the human and worm lineages. In this scenario, the yeast gene is orthologous to all worm and human genes, which are all co-orthologous to the yeast gene. When comparing the human and worm genes, all genes in the HA* set are co-orthologous to all genes in the WA* set. The genes HA* are hence 'inparalogs' to each other when comparing human to worm. By contrast, the genes HB and HA* are 'outparalogs' when comparing human with worm..However, HB and HA*, and WB and WA* are inparalogs when comparing with yeast, because the animal–yeast split pre-dates the HA*–HB duplication. (b) Real-life example of inparalogs: γ-butyrobetaine hydroxylases. The points of speciation and duplication are easily identifiable. The alignment is a subset of Pfam:PF03322 and the tree was generated by neighbor-joining in Belvu. All nodes have a bootstrap support exceeding 95%.

Phylogenetic Trees

- Nodes: species
- Edges: time of independent evolution
- Edge length represents evolution time
  - AKA genetic distance
  - Not necessarily chronological time

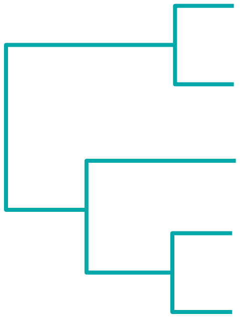


Inferring Phylogenetic Trees

Trees can be inferred by several criteria:

- Morphology of the organisms
  - *Can lead to mistakes*
- Sequence comparison

Example:



Mouse:	ACAGTGACGCCCCAAACGT
Rat:	ACAGTGACGCTACAAACGT
Baboon:	CCTGTGACGTAACAAACGA
Chimp:	CCTGTGACGTAGCAAACGA
Human:	CCTGTGACGTAGCAAACGA

Inferring Phylogenetic Trees

- Sequence-based methods
  - Deterministic (Parsimony)
  - Probabilistic (SEMPHY)

- Distance-based methods
  - UPGMA
  - Neighbor-Joining

- Can compute distances from sequences

Distance Between Two Sequences

Basic principle:

- Distance proportional to degree of independent sequence evolution

Given sequences x^i, x^j,

d_{ij} = distance between the two sequences

One possible definition:

d_{ij} = fraction f of sites u where $x^i[u] \neq x^j[u]$

Better scores are derived by modeling evolution as a continuous change process

Molecular Evolution

Modeling sequence substitution:

Consider what happens at a position for time Δt,

- $P(t)$ = vector of probabilities of $\{A,C,G,T\}$ at time t
- μ_{AC} = rate of transition from A to C per unit time
- $\mu_A = \mu_{AC} + \mu_{AG} + \mu_{AT}$ rate of transition out of A
- $p_A(t+\Delta t) = p_A(t) - p_A(t)\, \mu_A\, \Delta t + p_C(t)\, \mu_{CA}\, \Delta t + p_G(t)\, \mu_{GA}\, \Delta t + p_T(t)\, \mu_{TA}\, \Delta t$

Molecular Evolution

In matrix/vector notation, we get

$$P(t+\Delta t) = P(t) + Q\,P(t)\,\Delta t$$

where Q is the substitution rate matrix

$$Q = \begin{pmatrix} -\mu_A & \mu_{GA} & \mu_{CA} & \mu_{TA} \\ \mu_{AG} & -\mu_G & \mu_{CG} & \mu_{TG} \\ \mu_{AC} & \mu_{GC} & -\mu_C & \mu_{TC} \\ \mu_{AT} & \mu_{GT} & \mu_{CT} & -\mu_T \end{pmatrix}$$

Molecular Evolution

- This is a differential equation:

$$P'(t) = Q\,P(t)$$

- Q => prob. distribution over {A,C,G,T} at each position, stationary (equilibrium) frequencies π_A, π_C, π_G, π_T

- Each Q is an evolutionary model
  - Some work better than others

Evolutionary Models

- Jukes-Cantor

$$Q = \begin{pmatrix} * & \frac{\mu}{4} & \frac{\mu}{4} & \frac{\mu}{4} \\ \frac{\mu}{4} & * & \frac{\mu}{4} & \frac{\mu}{4} \\ \frac{\mu}{4} & \frac{\mu}{4} & * & \frac{\mu}{4} \\ \frac{\mu}{4} & \frac{\mu}{4} & \frac{\mu}{4} & * \end{pmatrix}$$

- Kimura

$$Q = \begin{pmatrix} * & \kappa & 1 & 1 \\ \kappa & * & 1 & 1 \\ 1 & 1 & * & \kappa \\ 1 & 1 & \kappa & * \end{pmatrix}$$

- Felsenstein

$$Q = \begin{pmatrix} * & \pi_T & \pi_T & \pi_T \\ \pi_C & * & \pi_C & \pi_C \\ \pi_A & \pi_A & * & \pi_A \\ \pi_G & \pi_G & \pi_G & * \end{pmatrix}$$

- HKY

$$Q = \begin{pmatrix} * & \kappa\pi_T & \pi_T & \pi_T \\ \kappa\pi_C & * & \pi_C & \pi_C \\ \pi_A & \pi_A & * & \kappa\pi_A \\ \pi_G & \pi_G & \kappa\pi_G & * \end{pmatrix}$$

Estimating Distances

- Solve the differential equation and compute expected evolutionary time given sequences

$$P'(t) = Q\,P(t)$$

Jukes-Cantor:

Let $P_{AA}(t) = P_{CC}(t) = P_{CC}(t) = P_{CC}(t) = r$

$P_{AC}(t) = \ldots = P_{TG}(t) = s$

Then,

$$r'(t) = -\tfrac{3}{4}\, r(t)\, \mu + \tfrac{3}{4}\, s(t)\, \mu$$

$$s'(t) = -\tfrac{1}{4}\, s(t)\, \mu + \tfrac{1}{4}\, r(t)\, \mu$$

Which is satisfied by

$$r(t) = \tfrac{1}{4}\,(1 + 3e^{-\mu t})$$

$$s(t) = \tfrac{1}{4}\,(1 - e^{-\mu t})$$

Estimating Distances

- Solve the differential equation and compute expected evolutionary time given sequences

$$P'(t) = Q\,P(t)$$

Jukes-Cantor:

$$P = \begin{pmatrix} \frac{1}{4} + \frac{3}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} \\ \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} + \frac{3}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} \\ \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} + \frac{3}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} \\ \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} - \frac{1}{4}e^{-t\mu} & \frac{1}{4} + \frac{3}{4}e^{-t\mu} \end{pmatrix}$$

Estimating Distances

Let p = probability a base is different between two sequences,

Solve to find **t**

- Jukes-Cantor $r(t) = 1 – p = \frac{1}{4}(1 + 3e^{-\mu t})$

$p = \frac{3}{4} – \frac{3}{4} e^{-\mu t}$

$\frac{3}{4} – p = \frac{3}{4} e^{-\mu t}$

$1 – 4p/3 = e^{-\mu t}$

Therefore,

$\mu t = - \ln(1 – 4p/3)$

Letting $d = \frac{3}{4} \mu t$, denoting substitutions per site,

$$d = -\frac{3}{4}\ln(1 - \frac{4}{3}p)$$

Estimating Distances

d: Branch length in terms of substitutions per site

- Jukes-Cantor

$$d = -\frac{3}{4}\ln(1 - \frac{4}{3}p)$$

- Kimura

$$d = -\frac{1}{2}\ln(1 - 2P - Q) - \frac{1}{4}\ln(1 - 2Q)$$

Simple method for building tree: UPGMA

UPGMA (unweighted pair group method using arithmetic averages)
Or the **Average Linkage Method**

Given two disjoint clusters C_i, C_j of sequences,

$$d_{ij} = \frac{1}{|C_i| \times |C_j|} \Sigma_{\{p \in Ci,\ q \in Cj\}} d_{pq}$$

Claim that if $C_k = C_i \cup C_j$, then distance to another cluster C_l is:

$$d_{kl} = \frac{d_{il}\ |C_i| + d_{jl}\ |C_j|}{|C_i| + |C_j|}$$

Algorithm: Average Linkage

Initialization:

Assign each x_i into its own cluster C_i

Define one leaf per sequence, height 0

Iteration:

Find two clusters C_i, C_j s.t. d_{ij} is min

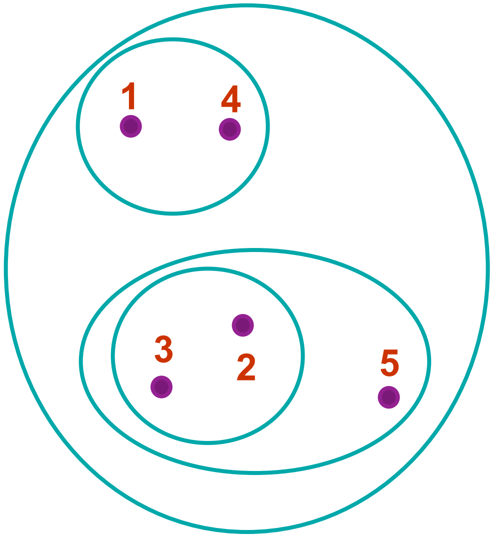
Let $C_k = C_i \cup C_j$

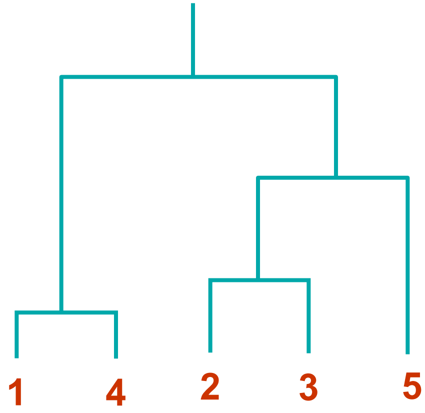
Define node connecting C_i, C_j, and place it at height $d_{ij}/2$

Delete C_i, C_j

Termination:

When two clusters i, j remain, place root at height $d_{ij}/2$





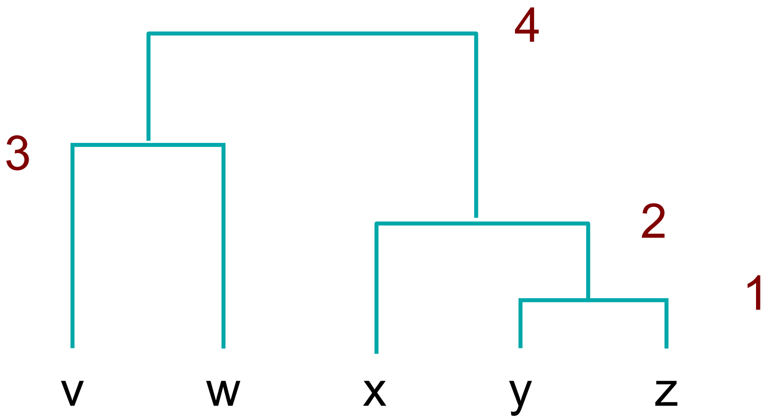
Average Linkage Example

	v	w	x	y	z
v	0	6	8	8	8
w		0	8	8	8
x			0	4	4
y				0	2
z					0

	v	w	xyz
v	0	6	8
w		0	8
xyz			0

	vw	xyz
vw	0	8
xyz		0

	v	w	x	yz
v	0	6	8	8
w		0	8	8
x			0	4
yz				0



Ultrametric Distances and Molecular Clock

Definition:

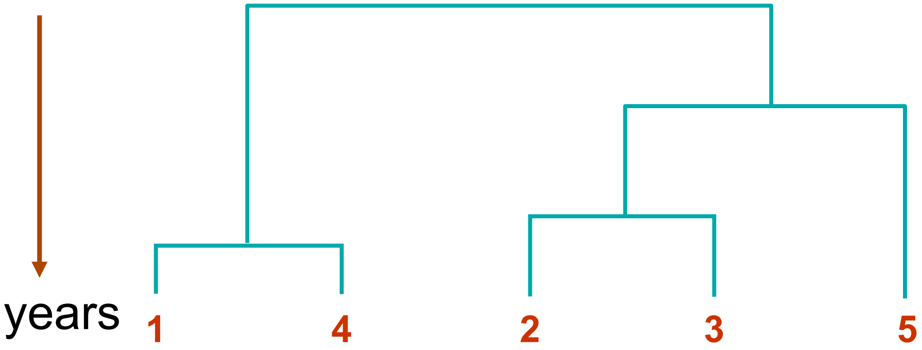
A distance function d(.,.) is ultrametric if for any three distances $d_{ij} \leq d_{ik} \leq d_{ij}$, it is true that

$$d_{ij} \leq d_{ik} = d_{jk}$$

The Molecular Clock:

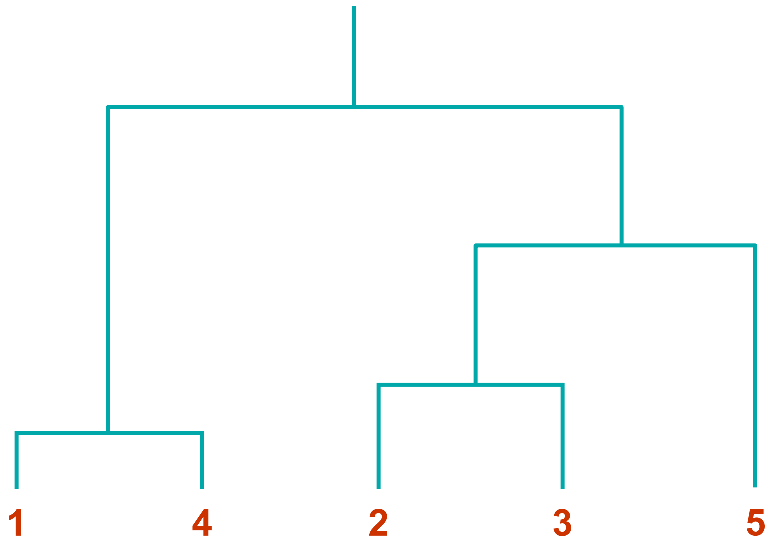
The evolutionary distance between species x and y is 2× the Earth time to reach the nearest common ancestor

That is, the molecular clock has constant rate in all species



The molecular clock results in ultrametric distances

Ultrametric Distances & Average Linkage



Average Linkage is guaranteed to reconstruct correctly a binary tree with ultrametric distances

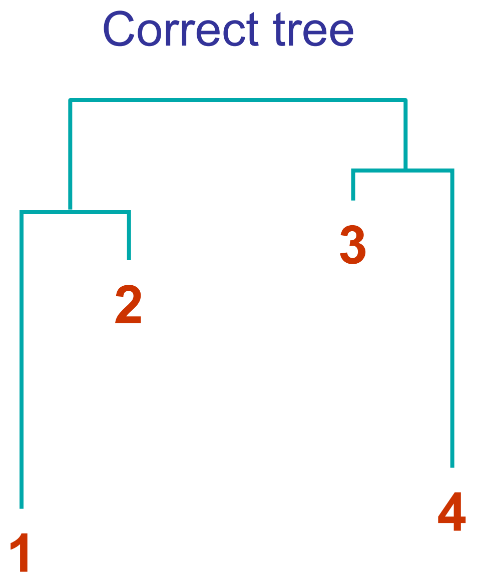
Proof: Exercise

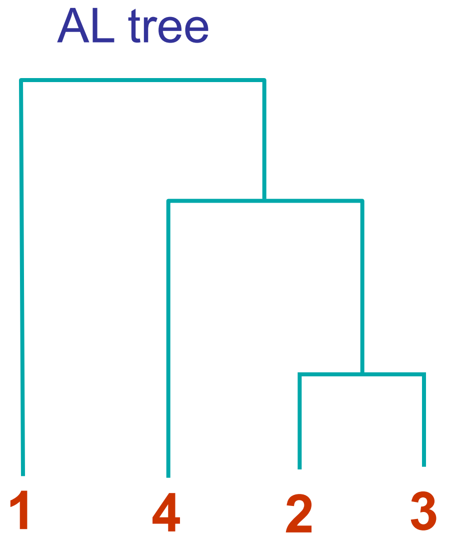
Weakness of Average Linkage

Molecular clock: all species evolve at the same rate (Earth time)

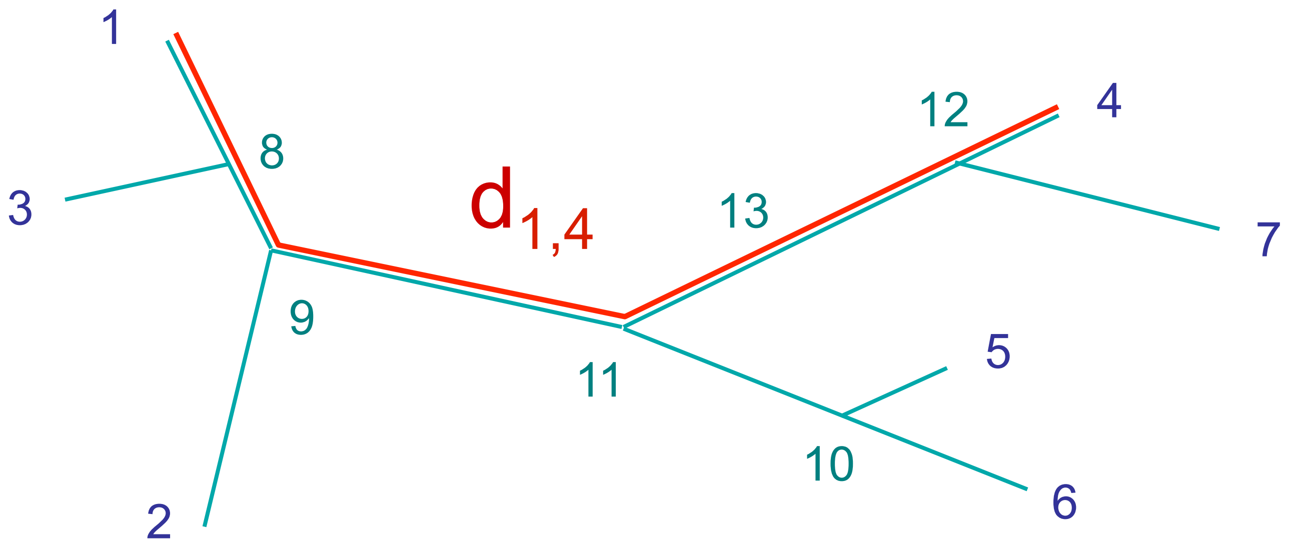
However, certain species (e.g., mouse, rat) evolve much faster

Example where UPGMA messes up:





Additive Distances

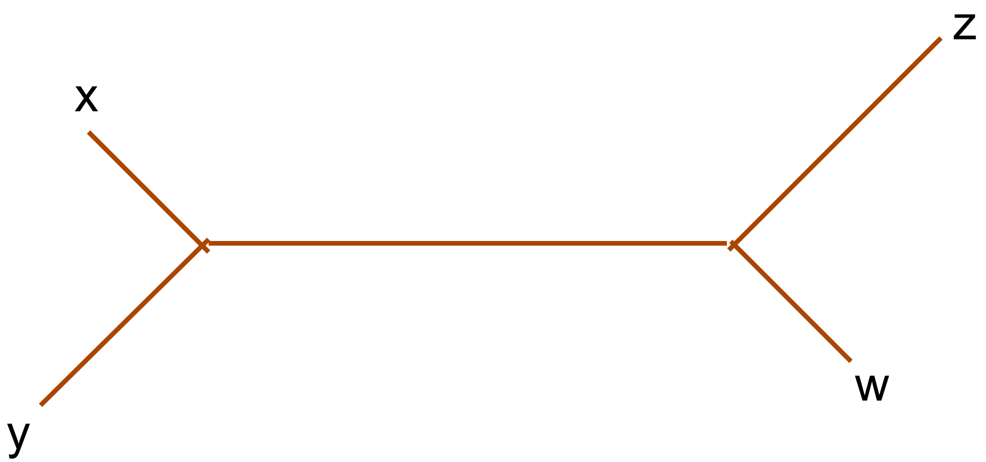


Given a tree, a distance measure is **additive** if the distance between any pair of leaves is the sum of lengths of edges connecting them

Given a tree T & additive distances d_{ij}, can uniquely reconstruct edge lengths:

- Find two neighboring leaves i, j, with common parent k
- Place parent node k at distance $d_{km} = ½ (d_{im} + d_{jm} – d_{ij})$ from any node m ≠ i, j

Additive Distances



For any four leaves x, y, z, w, consider the three sums

$$d(x, y) + d(z, w)$$
$$d(x, z) + d(y, w)$$
$$d(x, w) + d(y, z)$$

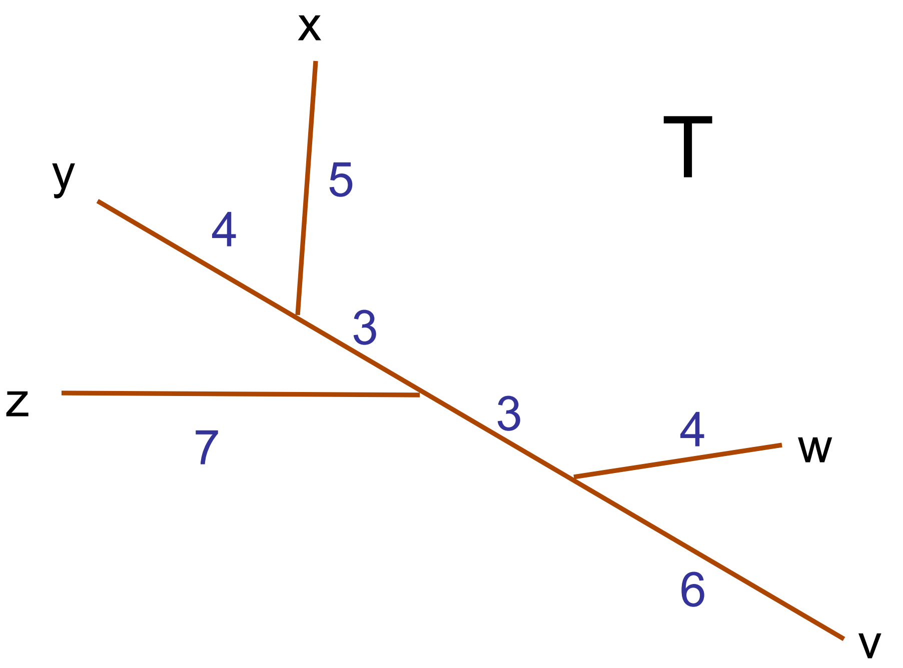
One of them is smaller than the other two, which are equal

$$d(x, y) + d(z, w) < d(x, z) + d(y, w) = d(x, w) + d(y, z)$$

Reconstructing Additive Distances Given T

D

	v	w	x	y	z
v	0	10	17	16	16
w		0	15	14	14
x			0	9	15
y				0	14
z					0

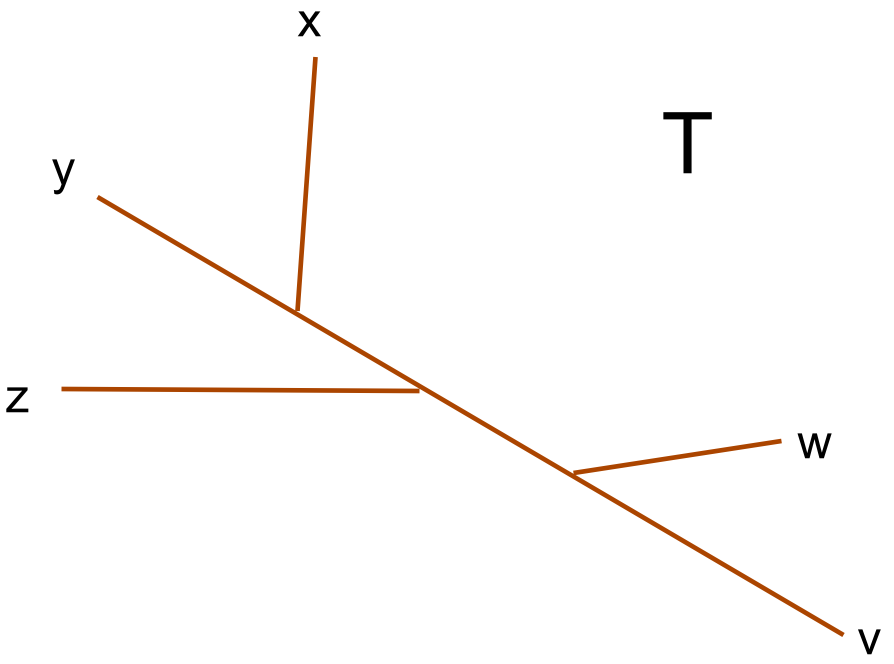


If we know T and D, but do not know the length of each leaf, we can reconstruct those lengths

Reconstructing Additive Distances Given T

D

	v	w	x	y	z
v	0	10	17	16	16
w		0	15	14	14
x			0	9	15
y				0	14
z					0



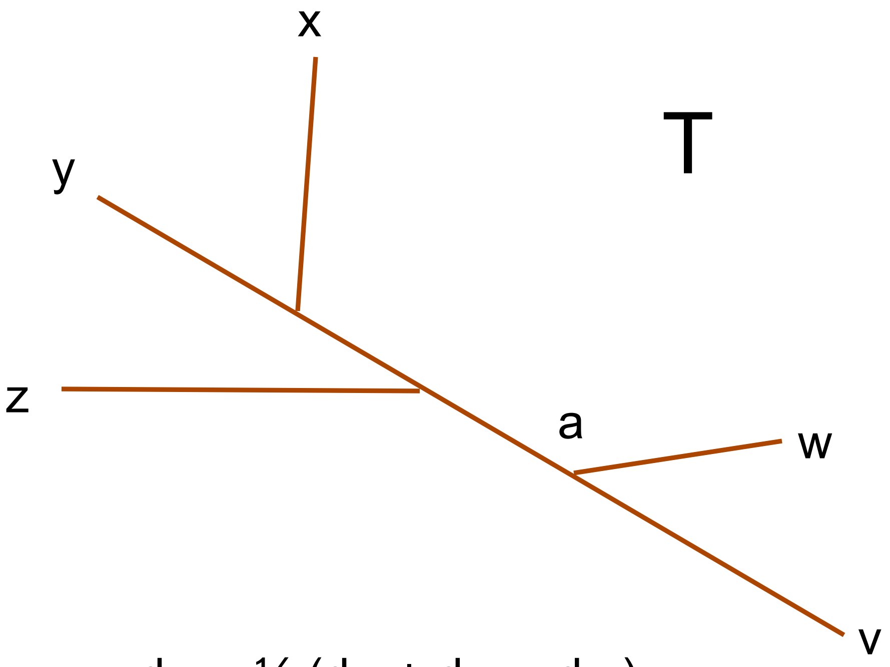
Reconstructing Additive Distances Given T

D

	v	w	x	y	z
v	0	10	17	16	16
w		0	15	14	14
x			0	9	15
y				0	14
z					0

D_1

	a	x	y	z
a	0	11	10	10
x		0	9	15
y			0	14
z				0



$$d_{ax} = ½ (d_{vx} + d_{wx} – d_{vw})$$

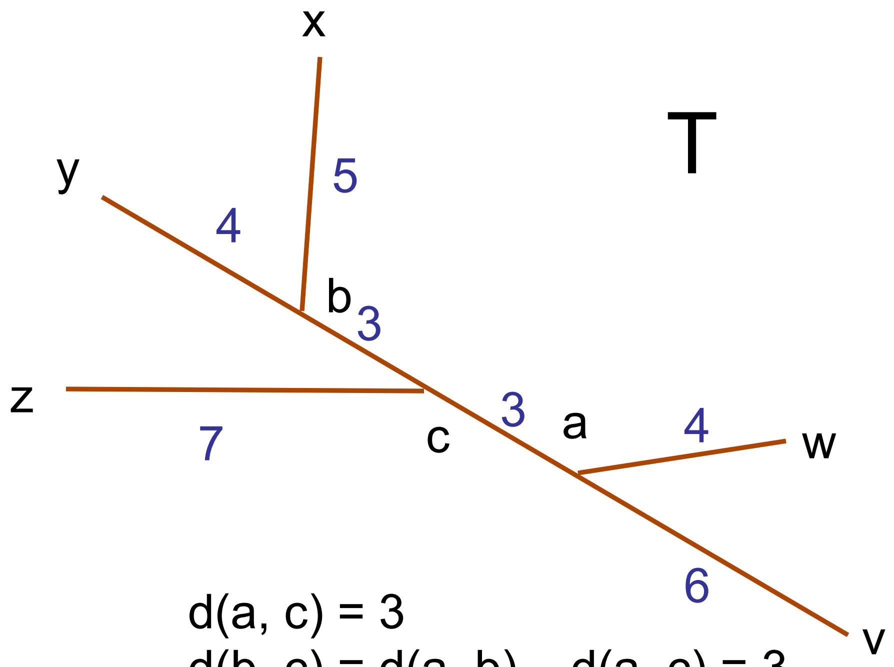
$$d_{ay} = ½ (d_{vy} + d_{wy} – d_{vw})$$

$$d_{az} = ½ (d_{vz} + d_{wz} – d_{vw})$$

Reconstructing Additive Distances Given T

D_1

	a	x	y	z
a	0	11	10	10
x		0	9	15
y			0	14
z				0



D_2

	a	b	z
a	0	6	10
b		0	10
z			0

D_3

	a	c
a	0	3
c		0

d(a, c) = 3
d(b, c) = d(a, b) – d(a, c) = 3
d(c, z) = d(a, z) – d(a, c) = 7
d(b, x) = d(a, x) – d(a, b) = 5
d(b, y) = d(a, y) – d(a, b) = 4
d(a, w) = d(z, w) – d(a, z) = 4
d(a, v) = d(z, v) – d(a, z) = 6

Correct!!!

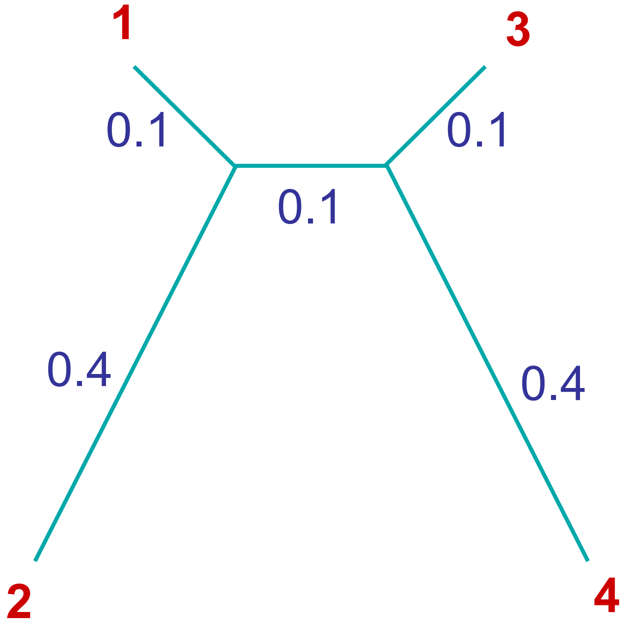
Neighbor-Joining

- Guaranteed to produce the correct tree if distance is additive
- May produce a good tree even when distance is not additive

Step 1: Finding neighboring leaves

Define

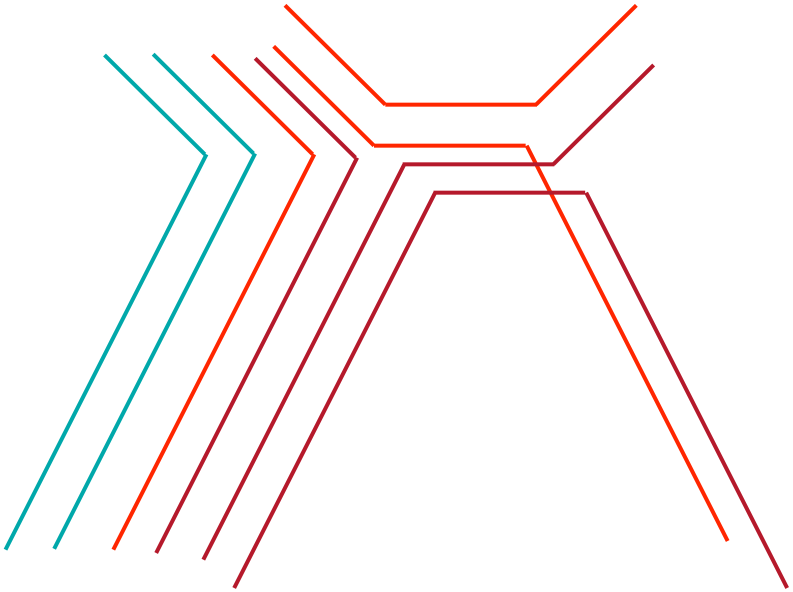
$$D_{ij} = (N - 2)\, d_{ij} - \sum_{k \neq i} d_{ik} - \sum_{k \neq j} d_{jk}$$

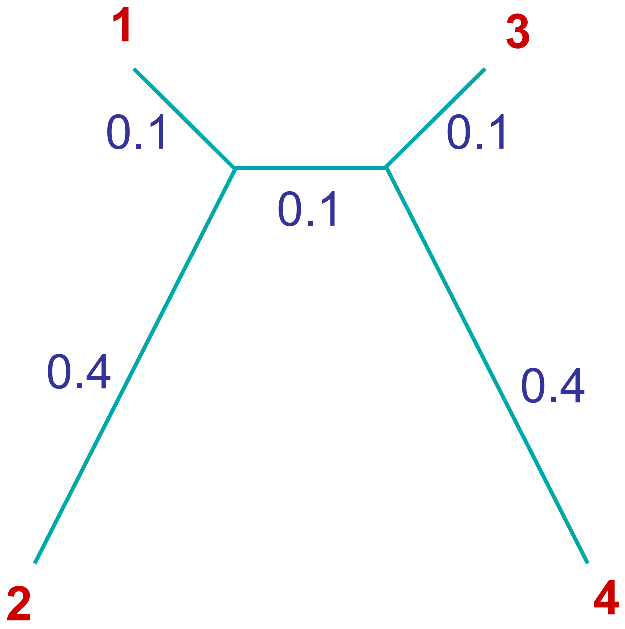


Claim: The above “magic trick” ensures that i, j are neighbors if D_{ij} is minimal

Neighbor-Joining

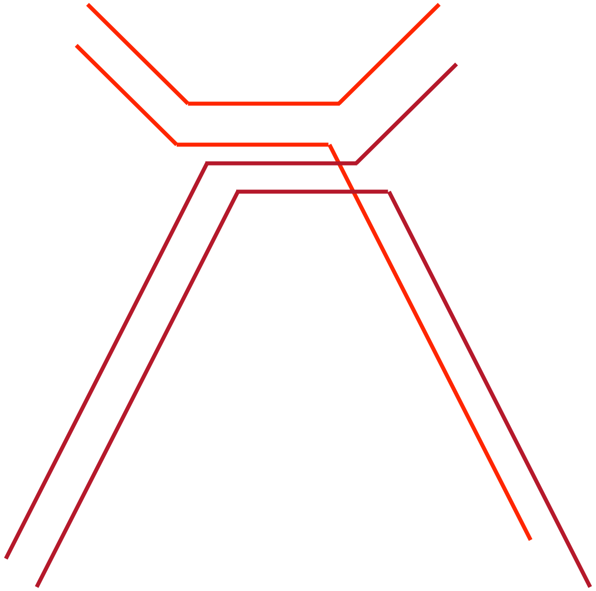
$$D_{ij} = (N - 2)\, d_{ij} - \sum_{k \neq i} d_{ik} - \sum_{k \neq j} d_{jk}$$

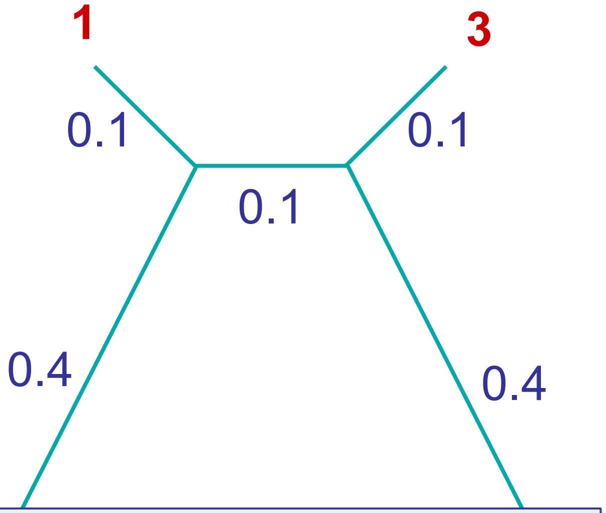




Neighbor-Joining

$$D_{ij} = (N - 2)\, d_{ij} - \sum_{k \neq i} d_{ik} - \sum_{k \neq j} d_{jk}$$





- All leaf edges appear negatively exactly twice

- All other edges appear negatively once for every path from each of the two leaves i, j, to leaves k ≠ i, j