

# Clustering Algorithms

Garrett Hellenthal  
g.hellenthal@ucl.ac.uk

University College London

**EMBO – Population Genomics: background, tools and  
programming**  
April 2, 2019

# Introduction

- ▶ this lecture/practical will cover *clustering algorithms*
- ▶ these classify individuals based on genetic similarity, so that interesting patterns emerge
- ▶ we will use SNP data, though some programs (e.g. *STRUCTURE*) should cope with microsatellites

# Outline

STRUCTURE/ADMIXTURE/FRAPPE

CHROMOPAINTER/fineSTRUCTURE

# Outline

STRUCTURE/ADMIXTURE/FRAPPE

CHROMOPAINTER/fineSTRUCTURE

# STRUCTURE

(Pritchard et al 2000, *Genetics* 155:945)

- ▶ **Aim:** classify individuals into  $K$  clusters based on genetic similarity
- ▶ let  $\{S_{i1}, \dots, S_{iL}\}$  be genetic data at locus  $l$  ( $L$  total loci) for ind  $i$  (note:  $\times 2$  for diploids)
- ▶ assume individual  $i$  is assigned to cluster  $k$
- ▶ then  $\Pr(S_{il} = j) = p_{kjl}$ , where  $p_{kjl}$  is frequency of allele  $j$  at locus  $l$  in cluster  $k$

# STRUCTURE

(Pritchard et al 2000, *Genetics* 155:945)

- ▶ **Aim:** classify individuals into  $K$  clusters based on genetic similarity
- ▶ let  $\{S_{i1}, \dots, S_{iL}\}$  be genetic data at locus  $l$  ( $L$  total loci) for ind  $i$  (note:  $\times 2$  for diploids)
- ▶ assume individual  $i$  is assigned to cluster  $k$
- ▶ then  $\Pr(S_{il} = j) = p_{kjl}$ , where  $p_{kjl}$  is frequency of allele  $j$  at locus  $l$  in cluster  $k$
- ▶ can infer  $p_{kjl}$ 's and cluster assignments of each ind by:
  - A. start with random assignment of inds to clusters  $1, \dots, K$
  - B. infer each  $p_{kjl}$  by using  $\text{freq}(j)$  at locus  $l$  among inds currently assigned to cluster  $k$
  - C. test how well (probability) each ind in  $k$  fits the inferred  $p_{kjl}$  from (B); if bad fit → move ind to different cluster
  - D. repeat (B)-(C) using Markov-Chain-Monte-Carlo (MCMC)

# STRUCTURE

(Pritchard et al 2000, *Genetics* 155:945)

**1-SNP Example:** classify 12 (haploid) individuals into  $K = 2$  clusters:

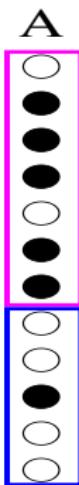


- A. start with random assignment of inds to clusters 1 and 2
- B. infer each  $p_{kjl}$  by using freq( $j$ ) at SNP  $l$  among inds currently assigned to cluster  $k$  ( $k = \square/\square$ ,  $j = \bullet/\circ$ ,  $l = 1$ )
- C. test how well (probability) each ind in  $k$  fits the inferred  $p_{kjl}$  from (B); if bad fit → move ind to different cluster
- D. repeat (B)-(C) using Markov-Chain-Monte-Carlo (MCMC)

# STRUCTURE

(Pritchard et al 2000, *Genetics* 155:945)

**1-SNP Example:** classify 12 (haploid) individuals into  $K = 2$  clusters:

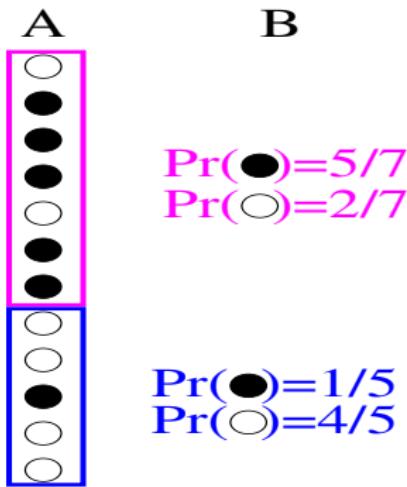


- A. start with random assignment of inds to clusters 1 and 2
- B. infer each  $p_{kjl}$  by using freq( $j$ ) at SNP  $l$  among inds currently assigned to cluster  $k$  ( $k = \text{pink}/\text{blue}$ ,  $j = \bullet/\circ$ ,  $l = 1$ )
- C. test how well (probability) each ind in  $k$  fits the inferred  $p_{kjl}$  from (B); if bad fit  $\rightarrow$  move ind to different cluster
- D. repeat (B)-(C) using Markov-Chain-Monte-Carlo (MCMC)

# STRUCTURE

(Pritchard et al 2000, *Genetics* 155:945)

**1-SNP Example:** classify 12 (haploid) individuals into  $K = 2$  clusters:

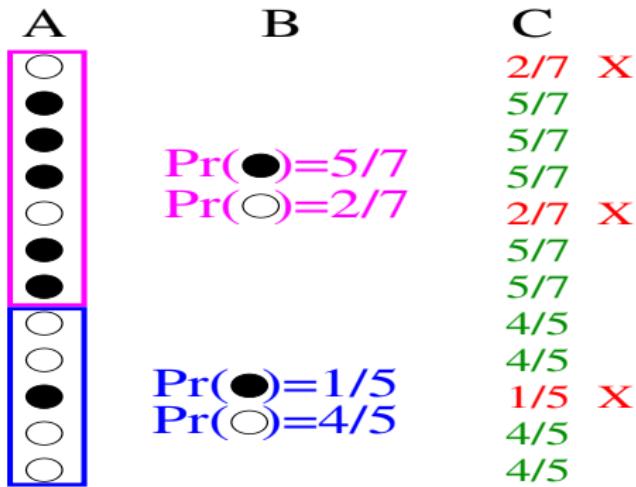


- start with random assignment of inds to clusters 1 and 2
- infer each  $p_{kjl}$  by using freq( $j$ ) at SNP  $l$  among inds currently assigned to cluster  $k$  ( $k = \bullet/\circ$ ,  $j = \bullet/\circ$ ,  $l = 1$ )
- test how well (probability) each ind in  $k$  fits the inferred  $p_{kjl}$  from (B); if bad fit  $\rightarrow$  move ind to different cluster
- repeat (B)-(C) using Markov-Chain-Monte-Carlo (MCMC)

# STRUCTURE

(Pritchard et al 2000, *Genetics* 155:945)

**1-SNP Example:** classify 12 (haploid) individuals into  $K = 2$  clusters:

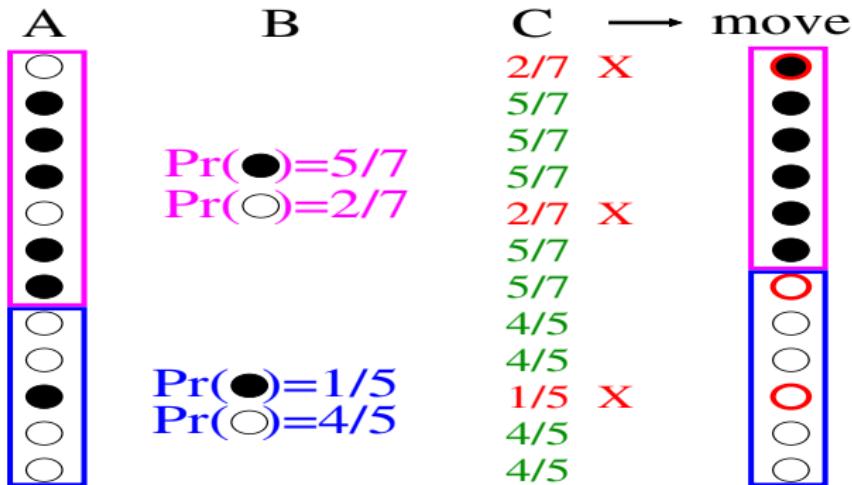


- start with random assignment of inds to clusters 1 and 2
- infer each  $p_{kjl}$  by using freq( $j$ ) at SNP  $l$  among inds currently assigned to cluster  $k$  ( $k = \square/\square$ ,  $j = \bullet/\circ$ ,  $l = 1$ )
- test how well (probability) each ind in  $k$  fits the inferred  $p_{kjl}$  from (B); if bad fit  $\rightarrow$  move ind to different cluster
- repeat (B)-(C) using Markov-Chain-Monte-Carlo (MCMC)

# STRUCTURE

(Pritchard et al 2000, *Genetics* 155:945)

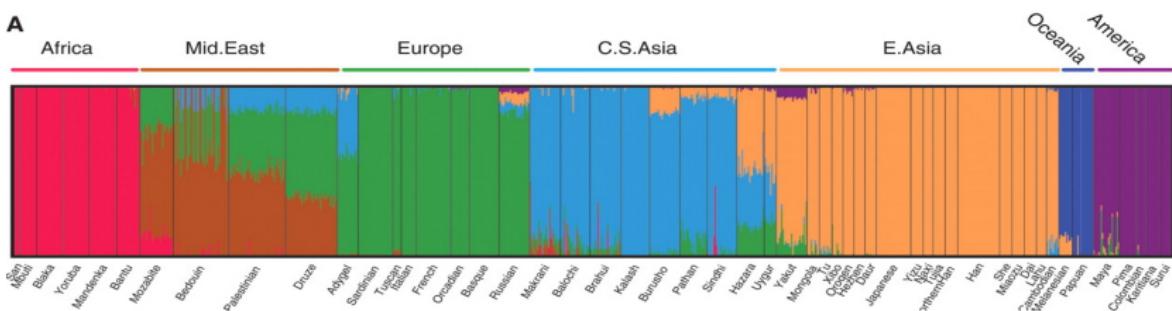
**1-SNP Example:** classify 12 (haploid) individuals into  $K = 2$  clusters:



- start with random assignment of inds to clusters 1 and 2
- infer each  $p_{kjl}$  by using freq( $j$ ) at SNP  $l$  among inds currently assigned to cluster  $k$  ( $k = \bullet/\circ$ ,  $j = \bullet/\circ$ ,  $l = 1$ )
- test how well (probability) each ind in  $k$  fits the inferred  $p_{kjl}$  from (B); if bad fit  $\rightarrow$  move ind to different cluster
- repeat (B)-(C) using Markov-Chain-Monte-Carlo (MCMC)

## STRUCTURE (Pritchard et al 2000, *Genetics* 155:945)

- ▶ There are different, much faster variants of *STRUCTURE*, which maximize the likelihood rather than sampling via MCMC:
    1. *FRAPPE* (Tang et al 2005, *Am J Hum Genet* **79**:1) – maximize likelihood using Expectation-Maximization (E-M)
    2. *ADMIXTURE* (Alexander et al 2009, *Genome Research* **19**:1655) – maximize likelihood using high-dimensional optimisation

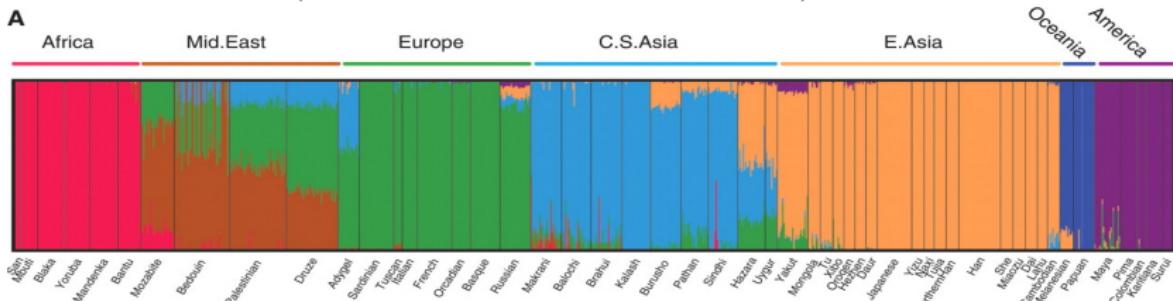


example of *FRAPPE* on HGDP (Li et al 2008, *Science* **319**:1100)

# STRUCTURE

 (Pritchard et al 2000, *Genetics* 155:945)

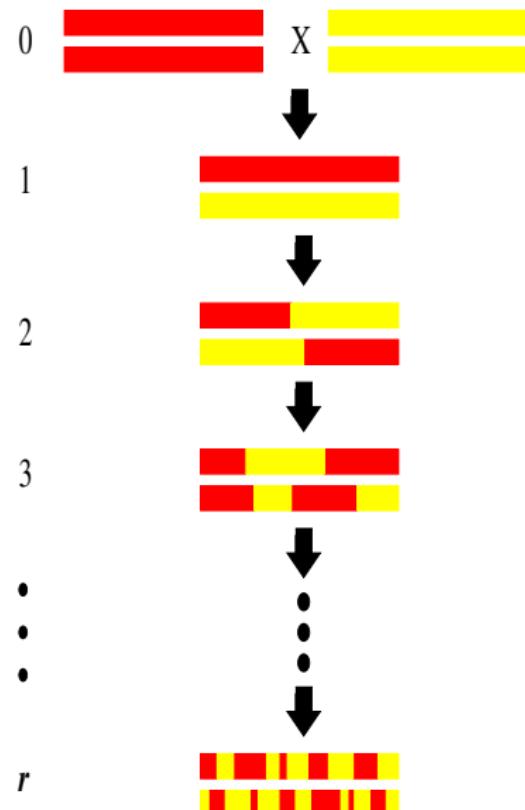
- ▶ There are different, much faster variants of *STRUCTURE*, which maximize the likelihood rather than sampling via MCMC:
  1. *FRAPPE* (Tang et al 2005, *Am J Hum Genet* 79:1) – maximize likelihood using Expectation-Maximization (E-M)
  2. *ADMIXTURE* (Alexander et al 2009, *Genome Research* 19:1655) – maximize likelihood using high-dimensional optimisation
- ▶ There are different flavors of *STRUCTURE*:
  - ▶ “no admixture model” – assign each ind  $i$  to single cluster  $k$
  - ▶ “admixture model” – assign each ind to multiple clusters (i.e. infer % of ind  $i$ 's genome assigned to clusters  $1, \dots, K$ )
  - ▶ “linkage model” – can identify regions of ind  $i$  assigned to each cluster (Falush et al 2003, *Genetics* 164:1567)



example of *FRAPPE* on HGDP (Li et al 2008, *Science* 319:1100)

## “linkage model” STRUCTURE – motivation

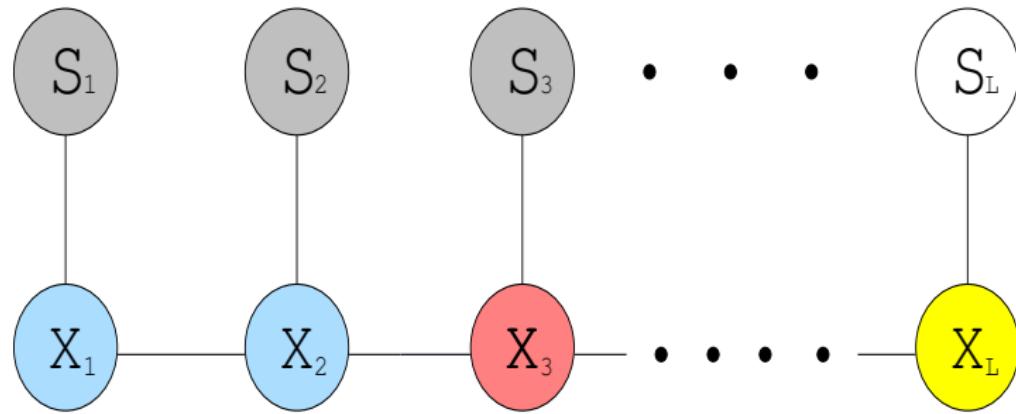
- ▶ two populations (**red**,**yellow**) admix  $r$  generations ago, followed by random mating
- ▶ genetic pieces from each population get smaller each subsequent generation due to recombination
- ▶ assuming no crossover interference, boundaries between contiguous **red** and **yellow** segments (from ancestor at time of admixture) in present-day DNA form a Poisson process with rate  $r$  per Morgan



(Falush et al 2003, *Genetics* 164:1567)

## “linkage model” STRUCTURE – Hidden Markov Model (HMM)

- ▶  $S_l$  = (observed state) SNP data at locus  $l$  of haploid  $i$
- ▶  $X_l$  = (hidden state) cluster  $1, \dots, k$  assignment at SNP  $l$



- ▶ each cluster is depicted with a unique color here
- ▶ assume “switches” in cluster assignment occur as a Poisson process of rate  $\propto r$ , the “time since admixture”
- ▶ “switches” also depend on genetic distance (cM) between SNPs

## “linkage model” STRUCTURE (Falush et al 2003, *Genetics* 164:1567)

Some mathematical details of “linkage model” STRUCTURE:

- models  $\Pr(x_{l+1}^{(i)})$ , the probability that haploid  $i$  is assigned to cluster  $k$  at SNPs( $l + 1$ ), as:

$$\Pr(x_{l+1}^{(i)} = k \mid x_l^{(i)} = k') = \begin{cases} \exp(-d_l/r) + (1 - \exp(-d_l/r))q_k^{(i)} & \text{if } k = k' \\ (1 - \exp(-d_l/r))q_k^{(i)} & \text{otherwise} \end{cases}$$

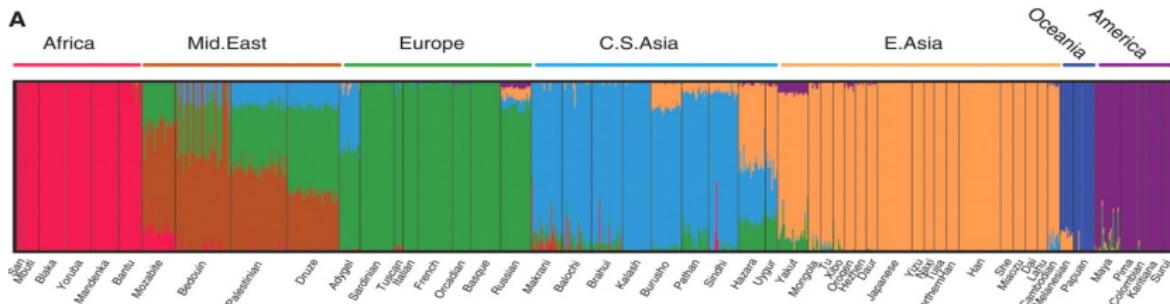
$$\Pr(s_{l+1}^{(i)} = a \mid x_{l+1}^{(i)} = k) = \Pr(\text{cluster } k \text{ carries allele } a \text{ at site } l + 1) \equiv P_{(l+1)a}^{(k)}$$

- where  $s^{(i)}$  = data,  $d_l$  = genetic distance btwn  $l$  and  $l + 1$ ,  $r$  = “time since admixture”
- estimate  $P_{(l+1)a}^{(k)}$  and  $q_k^{(i)}$ , for fixed number of clusters  $K$  using MCMC

# STRUCTURE (Pritchard et al 2000, *Genetics* 155:945), etc

## ► Advantages:

- ▶ *ADMIXTURE*, *FRAPPE* are computationally fast
- ▶ find clear structure in data (e.g. separate continents)
- ▶ can detect admixture (e.g. **Europe** in **Maya** from Mexico)



example of *FRAPPE* on HGDP (Li et al 2008, *Science* 319:1100)

## ► Disadvantages:

- ▶ how to interpret results? Drift/admixture/other can result in similar signals
- ▶ genetic loci (e.g. SNPs) assumed independent
- ▶  $K$  is fixed (though see *STRUCTURAMA* – (Huelsenbeck & Andolfatto 2007, *Genetics* 175:1787))

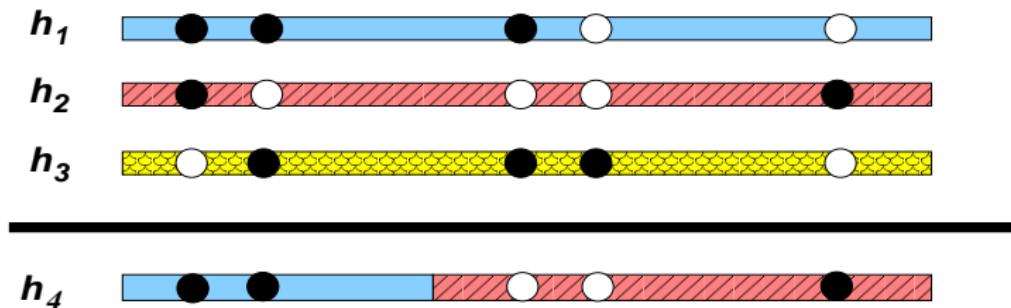
# Outline

STRUCTURE/ADMIXTURE/FRAPPE

CHROMOPAINTER/fineSTRUCTURE

# Incorporating haplotype information: chromosome painting

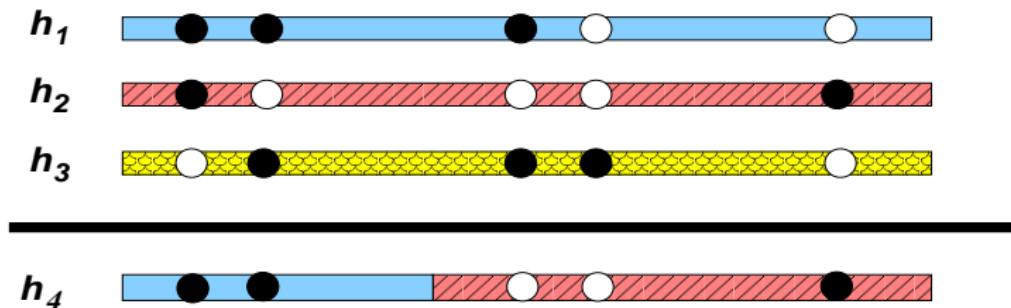
(Lawson et al 2012, *PLoS Genet* 8:e1002453)



- ▶ use some sampled chromosomes (e.g.  $h_1, h_2, h_3$ ) as “donors”
- ▶ match (or “paint”) other chromosomes (e.g.  $h_4$ ) to donors’ DNA
- ▶ → cluster based on who shares many **blocks of SNPs** rather than who shares similar **SNP frequencies**

# Incorporating haplotype information: chromosome painting

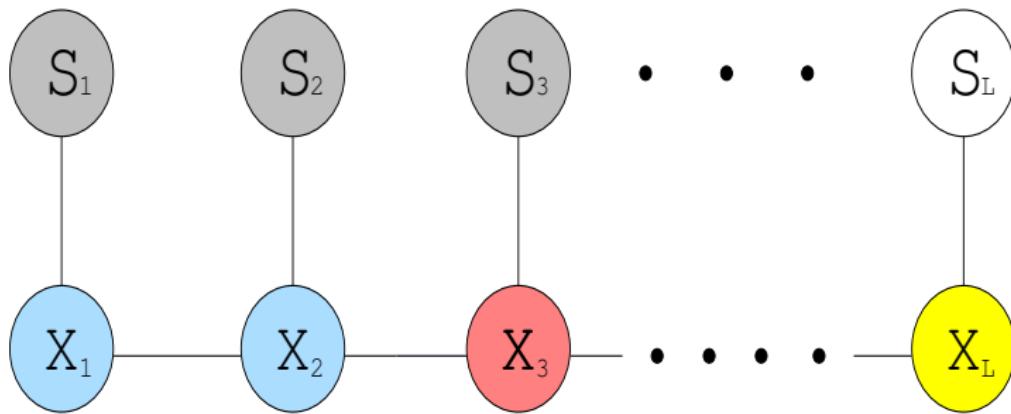
(Lawson et al 2012, *PLoS Genet* 8:e1002453)



- ▶ use some sampled chromosomes (e.g.  $h_1$ ,  $h_2$ ,  $h_3$ ) as “donors”
- ▶ match (or “paint”) other chromosomes (e.g.  $h_4$ ) to donors’ DNA
- ▶ → cluster based on who shares many **blocks of SNPs** rather than who shares similar **SNP frequencies**
- ▶ can do “painting” using many approaches, e.g:
  1. *CHROMOPAINTER* (Lawson et al 2012, *PLoS Genet* 8:e1002453)
  2. *HAPMIX* (Price et al 2009, *PLoS Genet* 5:e1000519)
  3. *RFMIX* (Maples et al 2013, *AJHG* 93:278)
  4. *MULTIMIX* (Churchhouse & Marchini 2013, *Genet Epidemiol* 37:1)

## CHROMOPAINTER – Hidden Markov Model (HMM)

- ▶  $S_l$  = (observed state) SNP data at locus  $l$  of haploid  $i$
- ▶  $X_l$  = (hidden state) donor haploid  $1, \dots, d$  copied at SNP  $l$



- ▶ each donor haploid is depicted with a unique color here
- ▶ assume “switches” in donor copied occur as a Poisson process of rate  $\propto g_l$ , the cM distance between SNPs  $l$  and  $l + 1$

# Copying Model (HMM) (Lawson et al 2012, PLoS Genet 8:e1002453)

(based on Li & Stephens 2003, Genetics 165:2213)

$X_l$  = (unknown) “donor” haplotype copied at SNP  $l$

$S_l$  = observed data at SNP  $l$

$$\Pr(X_{l+1} = d \mid X_l = d') = \begin{cases} \exp(-g_l N_e) + (1 - \exp(-g_l N_e)) q_d & \text{if } d = d' \\ (1 - \exp(-g_l N_e)) q_d & \text{otherwise} \end{cases}$$

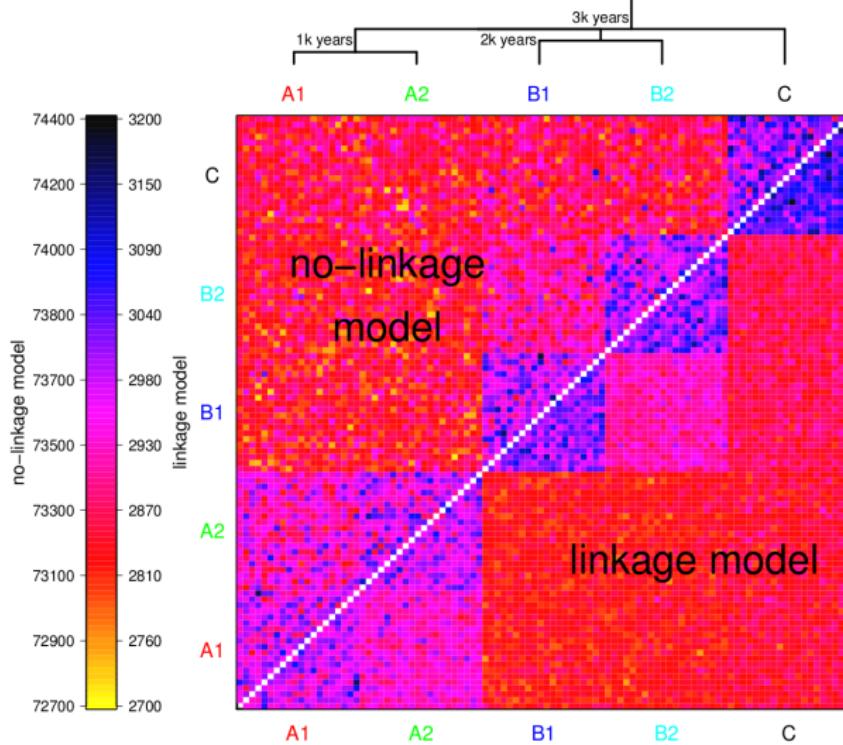
$$\Pr(S_{l+1} = s \mid S_{X_{l+1}} = s_X) = \begin{cases} 1 - \theta & s = s_X \\ \theta & s \neq s_X \end{cases}$$

- ▶ where  $g_l$  = genet distance btwn SNPs  $l$  and  $l + 1$ ,  $N_e$  = “scaling term”,  $\theta$  = mutation (emission) rate,  $q_d = \Pr(\text{copy } d)$
- ▶ estimates proportion copied from donor  $d$  conditional on data
- ▶ can sample  $X_l$  along genome

## Summarizing *CHROMOPAINTER* painting: Heatmaps

- ▶ allow each individual  $i$  to copy from every other individual  $j \neq i$  using the copying model
- ▶ calculate  $y_{ij}$  – expected number of “chunks” ind  $i$  copies from ind  $j$ 
  - ▶ if you assume each SNP is a “chunk” (i.e. “no-linkage” model) you capture information equivalent to PCA of genetic data
  - ▶ considerably more power if you use haplotype-based model

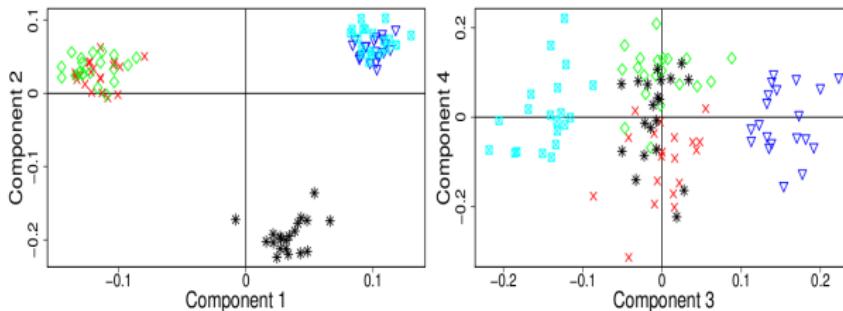
# Simulated Example – 5 pops (150 5-Mb regions)



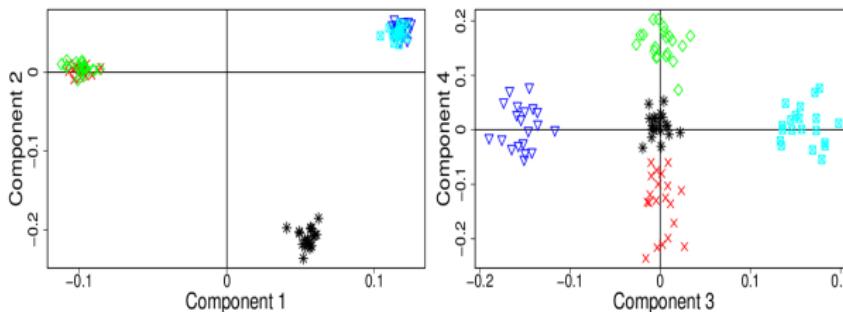
- ▶ **Heatmap:** each square is the number of DNA segments that each row (recipient) copies from (i.e. is painted by) each column (donor)
- ▶ **upper left triangle:** ignoring haplotypes
- ▶ **lower right triangle:** using haplotypes

# Simulated Example – 5 pops (PCA of heatmap)

Unlinked PCA



Linked PCA



top row: ignoring haplotypes (equivalent to regular PCA)

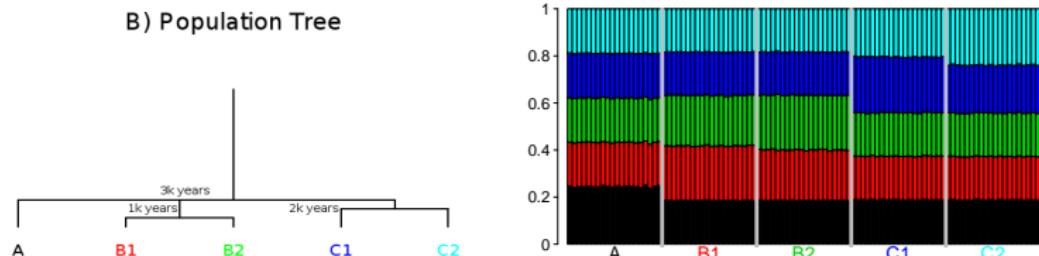
bottom row: using haplotypes

# *fineSTRUCTURE*: cluster using *CHROMOPAINTER* paintings

## B) Population Tree

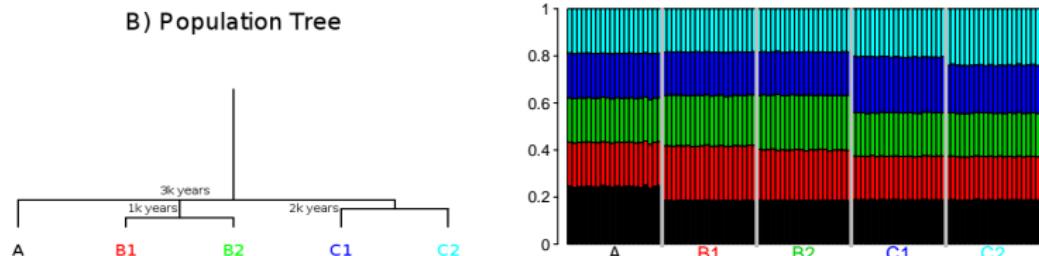


## *fineSTRUCTURE*: cluster using *CHROMOPAINTER* paintings

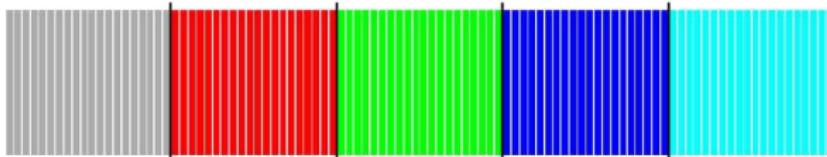


- ▶ paint each individual  $i$  using all other inds as “donors”
- ▶ calculate  $y_{ij}$  – number of DNA segments of ind  $i$  painted by ind  $j$

## *fineSTRUCTURE*: cluster using *CHROMOPAINTER* paintings

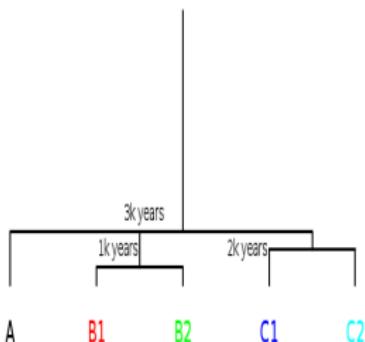


- ▶ paint each individual  $i$  using all other inds as “donors”
- ▶ calculate  $y_{ij}$  – number of DNA segments of ind  $i$  painted by ind  $j$
- ▶ cluster individuals who have similar painting patterns (MCMC):
  1. start with random assignment of inds to clusters  $1, \dots, K$
  2.  $(y_{i1}, \dots, y_{iK}) \sim \text{Mult}(P_{A1}, \dots, P_{AK})$  for ind  $i$  assigned to cluster  $A$
  3. infer  $P_{Ak}$  by using number of segments by which inds in cluster  $A$  are painted by inds in cluster  $k$
  4. if  $\Pr(y_{i1}, \dots, y_{iK})$  low —> move ind  $i$  to different cluster

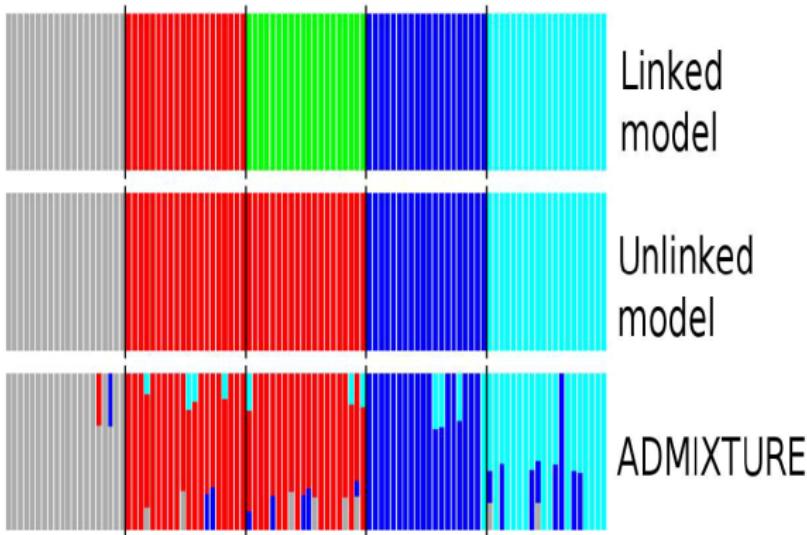


# Simulated Example – 5 pops (classification)

B) Population Tree

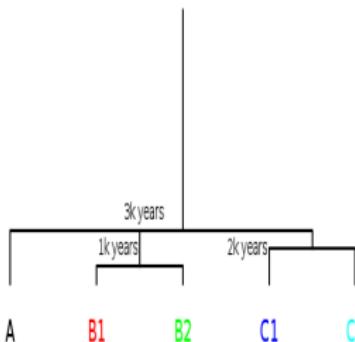


B) Barplot

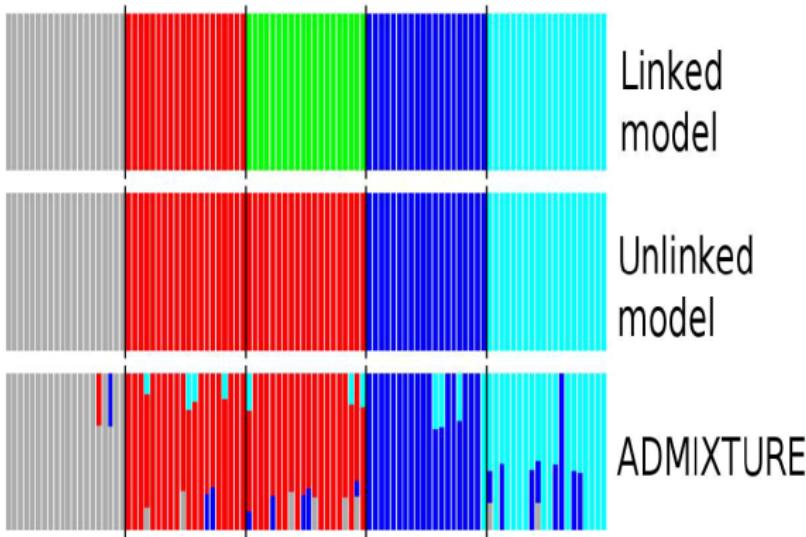


## Simulated Example – 5 pops (classification)

B) Population Tree

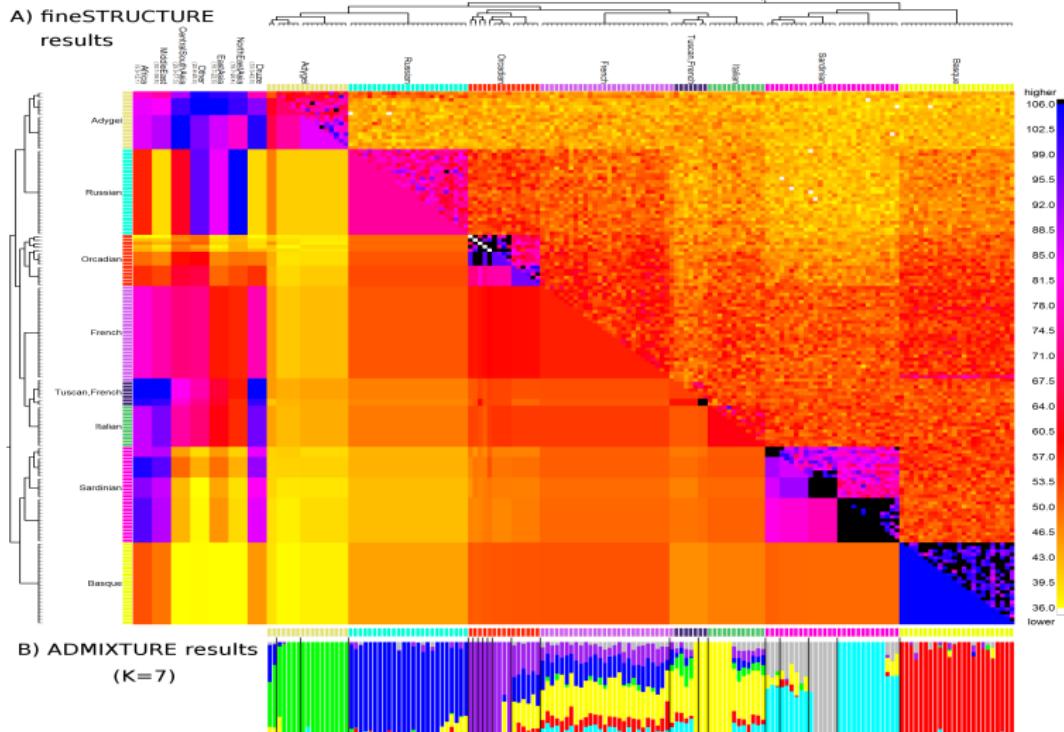


B) Barplot



- ▶ can build tree by greedily merging pairs of clusters, two-at-a-time, until all are merged

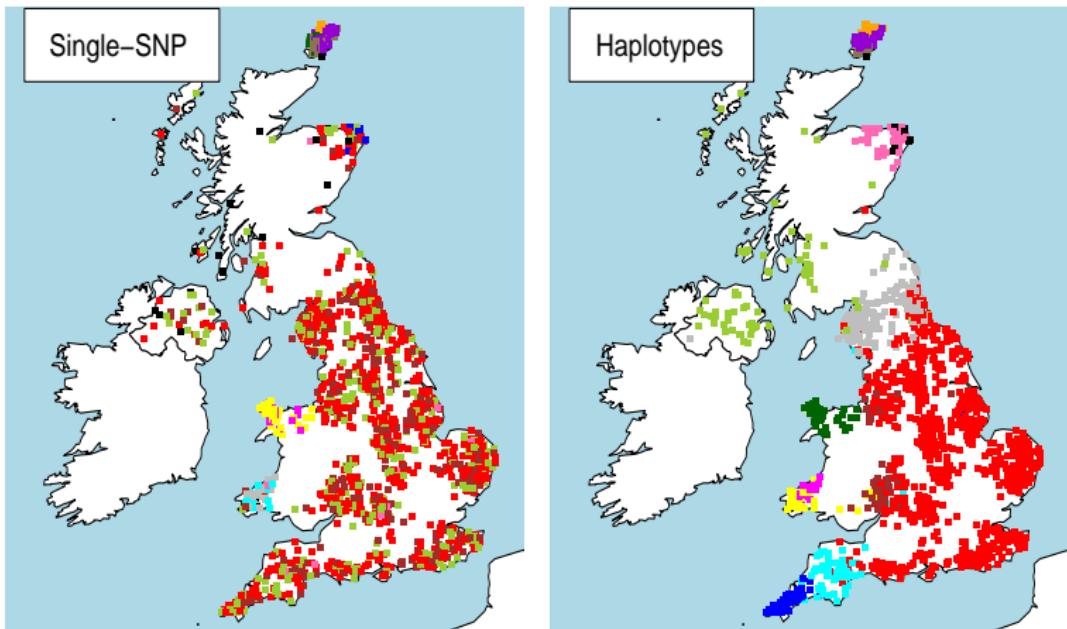
# European populations (Lawson et al 2012, PLoS Genet 8:e1002453)



**upper right triangle:** # of segments each row (recipient) copies from each column (donor)

**lower left triangle:** averages across all inds within same cluster

# United Kingdom – clustering using haplotype info



(Leslie et al 2015, *Nature* 519:309)

- ▶ dots = individuals / colors = clusters
- ▶ **left:** ignore haplotypes (as in PCA, STRUCTURE/ADMIXTURE)
- ▶ **right:** using haplotypes shows more localised (though subtle!) correspondence of genetics and geography

## ► Advantages:

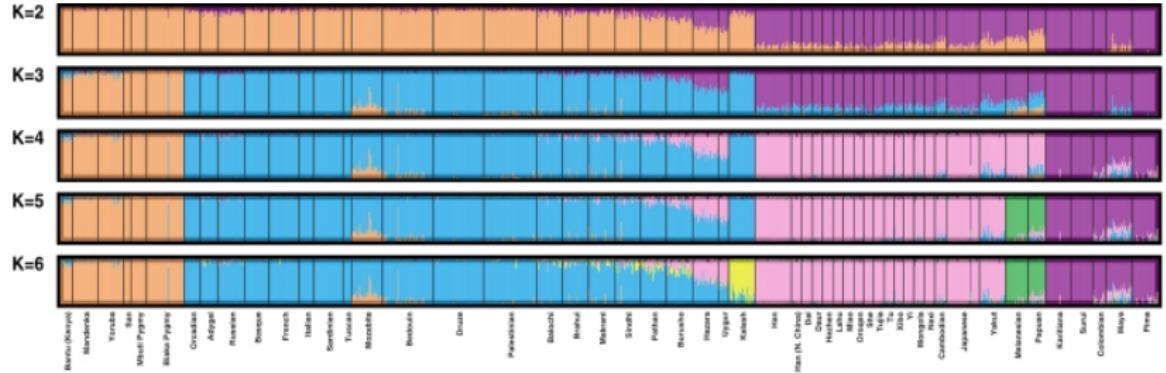
- ▶ increased power – finds more subtle structure in data missed by *ADMIXTURE* (e.g. Europe, United Kingdom)
- ▶ heatmaps can show clear structure/admixture
- ▶ current implementation infers number of clusters  $K$  automatically, and builds “tree” relating these clusters

## ► Disadvantages:

- ▶ how to interpret results? Drift/admixture/other can result in similar signals
- ▶ requires phased data (e.g. using *SHAPEIT*)
- ▶ currently only has “no admixture” model only
- ▶ computationally slow (relative to *ADMIXTURE*), more complicated to use

# Caution: Clustering individuals

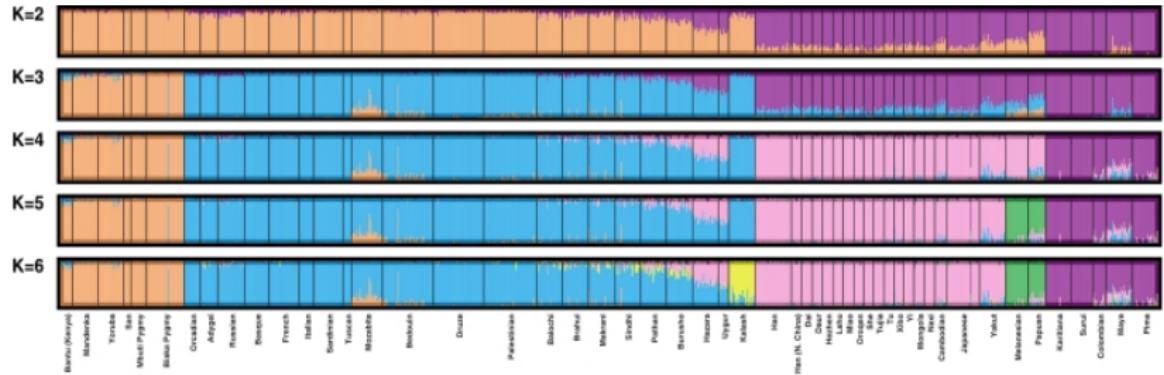
- ▶ can be difficult to interpret → drift, admixture, ancient relatedness can give similar signals



example of *STRUCTURE* (Rosenberg et al 2002, *Science* 298:2381)

## Caution: Clustering individuals

- ▶ can be difficult to interpret → drift, admixture, ancient relatedness can give similar signals



- ▶ with  $K = 6$  clusters, the *Kalash* of Pakistan assigned to own **yellow** cluster because they are relatively drifted  
(live within mountainous region → genetically isolated)
- ▶ has been interpreted as the Kalash being an ancient genetic isolate (Ayub et al 2015, *AJHG* 96:775)
- ▶ but recent work has shown ancestors of Kalash have recently intermixed with outside groups within last  $\approx 3k$  years (Hellenthal et al 2014)

## Summary

- ▶ clustering algorithms highlight genetic differences/similarities among groups (e.g. correlations between genetics and geography)
- ▶ more challenging to get at factors driving genetic differences
- ▶ *STRUCTURE/ADMIXTURE/FRAPPE* – assumes SNPs are independent, computationally quick (the latter two methods)
- ▶ *CHROMOPAINTER + fineSTRUCTURE* – uses correlations among SNPs (haplotype information) to increase power, computationally slower, individuals cannot be mixtures of multiple clusters