RC24677 (W0810-112) October 17, 2008
Computer Science

IBM Research Report

A Case for Recombinomics

Laxmi Parida[1], Asif Javed[1,2], Marta Melé[1,2,3,] Jaume Bertranpetit[3]

[1]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
USA

[2]Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY
USA

[3]Biologia Evolutiva
Universitat Pompeu Fabra
Barcelona, Catalonia
Spain

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

TECH REPORT

A Case for Recombinomics

Laxmi Parida[*†] Asif Javed[†‡§] Marta Melé [†‡¶] Jaume Bertranpetit[¶]

Abstract

In this report we present the results of the recombinational analysis based on our model IRiS. We investigate HapMap III database with 11 populations and over 1000 samples: we picked this testbed primarily due to the choice of SNPs in the database. In an effort to reduce the effects of compounding errors due to limitations of current technologies and techniques, we focus on the recombining X Chromosome. In our preliminary analysis, our results are two-fold. Firstly, we demonstrate the presence of recombinations-based evidence in short segments of the genome to detect subcontinental divide in the populations. We observe this in both populations-centered as well as recombinations-centered analysis. Secondly, we make the surprising observation that the effect of the population dynamics that shapes the allele-frequency variations between populations is also reflected in the purely recombination-based variations. We conclude that our recombinational-based exploration has the potential to go well beyond the known into non-traditional territories.

Contents

[*]Corresponding author: parida@us.ibm.com
[†]Computational Biology Center, IBM T J Watson Research, Yorktown, USA.
[‡]Department of Computer Science, Rensselaer Polytechnic Institute, New York, USA
[§]Work done during an internship at IBM T J Watson Research Center.
[¶]Biologia Evolutiva, Universitat Pompeu Fabra, Barcelona, Catalonia, Spain.

1 Exploring Human X Chromosome

We glean regions of X-Chromosome for our recombinational analysis. At this stage HapMap III is the most extensive in terms of population coverage as well as choice of SNPs. Although HGDP (Human Genome Diversity Panel: `http://www.stanford.edu/group/morrinst/hgdp.html`) database has data from very large number of ethnic groups, we found that the SNP density was not adequate for our analysis. We use the HapMap III database: this includes samples from eleven populations with the subcontinental divide as follows:

1. Four African (Af) populations: LWK, MKK, YRI and ASW.
2. One South-Asian (sA) population: GIH.
3. Three East-Asian (eA) populations: CHB, CHD, and JPT.
4. One American (Am) population: MEX.
5. Two European (Eu) populations: CEU and TSI.

This is summarized in the following table.

label	population sample	number of samples
LWK	Luhya in Webuye, Kenya	100
MKK	Maasai in Kinyawa, Kenya	180
YRI	Yoruba in Ibadan, Nigeria	180
ASW	African ancestry in Southwest USA	90
GIH	Gujarati Indians in Houston, Texas	100
CHB	Han Chinese in Beijing, China	90
CHD	Chinese in Metropolitan Denver, Colorado	100
JPT	Japanese in Tokyo, Japan	91
MEX	Mexican ancestry in Los Angeles, California	90
CEU	Utah residents with Northern and Western European ancestry from the CEUPH collection	180
TSI	Toscans in Italy	100

Some details on the SNP data. The SNP genotype data was generated from 1115 samples, collected using two platforms: the Illumina Human1M (by the Wellcome Trust Sanger Institute) and the Affymetrix SNP 6.0 (by the Broad Institute). Data from the two platforms have been merged for this release. The Illumina Human 1M Beadchip is focused on tagSNPs, SNPs in genes, and SNPs and non-polymorhic markers in known and novel copy number variation regions. There are $\approx$ 950,000 tag SNPs and $\approx$ 100,000 non-HapMap SNPs. There are 565,000 SNPs in and near coding regions and for the CNVs identification there are $\approx$ 260,000 markers. The Affymetrix SNP 6.0 chip includes more than 906,600 SNPs containing an unbiased selection of 482,000 SNPs from the Array 5,0, a selection of additional 424,000 tag SNPs, new SNPs added in the dbSNP database and SNPs in recombination hotspots. On the X-chromosome, there are approximately 16,500 SNPs in HGDP and 31,000 SNPs in HapMap III.

1.1 Screening Criteria

I. Identifying Potential Pitfalls. It is important to recognize the irrecoverable errors that any analysis may produce to avoid possible misinterpretation of the results. We identify two primary sources that a

recombinational-analysis must be wary of: presence of copy number variations (CNV) and segmental duplication (SD) in the input data. Our focus is on phylogeographic studies and thus to avoid the interplay of any potential selection and the recombinational landscape, we avoid the putative gene regions on the chromosome. Note that most phasing techniques are challenged by regions of low LD and by low frequency haplotypes. Thus to avoid phantom recombinations due to possible phasing errors, in the female samples, we pick only those that are homozygous in the screened region.

II. Enhancing signal-to-noise ratio. Note that not all SNPs are typed in all the populations in the database: we pick only those that are typed in *all* the eleven populations. Further, we eliminate those regions that do not have a sufficient number of SNPs (we use a cut-off limit of 80).

The details of our screening of the database, accounting for I and II above, is summarized below.

1. ChrX Regions: X chromosome of HapMap III (`www.hapmap.org`) data.

(a) at least 50 Kb away from genes. This information comes from Ensembl v37 (Feb 06) which is the last version of Build 35, retrieved using BioMart (`http://feb2006.archive.ensembl.org/Homo-sapiens/martview`).

(b) at least 50 Kb away from copy number variation (CNV) regions. This was done using Database of Genomic Variants (`http://projects.tcag.ca/variation/`), Build 35.

(c) at least 50 Kb away from segmental duplication (SD) regions. This was done using Segmental Duplications Database (`http://humanparalogy.gs.washington.edu`), Build 35.

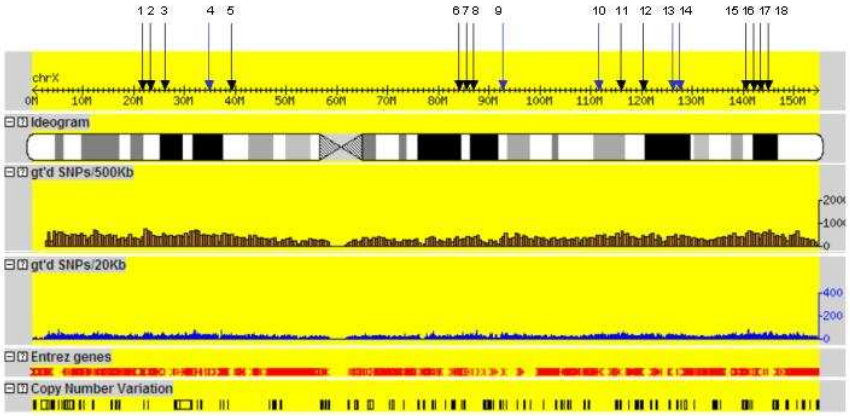
2. Samples: In each selected ChrX region, males or homozygous females are selected.

3. SNPs: In each selected ChrX region of the samples, we picked the SNPs by applying the following rules (in the order they are listed):

(a) SNP typed in all 11 populations,

(b) a total of at least 80 SNPs in the region,

(c) minor allele frequency (MAF) > 0.1 of total population.



				SNPs			Recombinations	
Label	Start	End	size (bp)	No.	density	samples	ρ	Hotspots
1	22505979	22728622	222643	95	1/2394	477	3.608	5
2	23071760	23213016	141256	96	1/1456	504	3.355	2
3	25715611	26016381	300770	83	1/3957	505	0.660	4
4 (Cg)	35038017	35504132	466115	81	1/5755	560	0.603	4
5	38875482	39480082	604607	179	1/3378	490	1.237	9
6	84704863	84952842	247979	80	1/3306	579	0.255	2
7	86338463	86609425	270962	90	1/3188	452	1.604	2
8	87288915	87838907	549992	204	1/2736	406	1.691	8
9 (Eg)	93522874	94555707	1032833	180	1/5738	392	0.895	6
10 (Dg)	112181012	112602418	421406	92	1/4581	514	0.375	2
11	116631417	116865805	234388	82	1/2894	527	1.849	2
12	120875730	121450338	574608	157	1/3831	431	1.335	7
13 (Bg)	125833172	126301999	468827	81	1/5787	492	0.526	3
14 (NH)	126499106	126892013	392907	72	1/5457	669	0.182	0
15	140883556	141050268	166712	99	1/1755	433	5.122	3
16	141376625	141647366	270741	88	1/3148	460	3.847	4
17	143563468	143896320	332852	96	1/3579	509	1.533	6
18	144769060	145266667	497607	162	1/3110	446	2.106	4
			7197205					

Figure 1: The 18 viable ChrX regions. The recombination rate ρ and the number of hotspots are computed from HapMap II data: NH is the only region without hotspots by this computation.

1.2 The 18 Viable ChrX Regions

This screening gives us 18 viable regions on the human X Chromosome for recombinational exploration. The distribution of these regions on the chromosome is shown in the ideogram. Further details on these regions are shown in Fig 1.

Recombination rates and the location and the number of recombination hotspots were estimated from Phase II HapMap data, release 21 (`www.hapmap.org`) using methods described in [MMH+04]. In Section 4.1 we discuss the possible interplay of hotspots with our analysis. We characterize the viable regions of the human ChrX: Fig 2 gives the breakdown of the samples by population and the value of minimum number of recombinations, R_M, for each population (see [HSW05] for R_M).

2 Method

Each region, which is a contiguous segment on the chromosome, is handled independently.

Biological Insights. The authors in [GSN+02] established the existence of haplotype blocks in humans. Exploiting this characteristic that appropriately chosen SNP's in a neighborhood on the chromosome exhibit linkage disequilibrium (or the lack of independence), we segment the input haplotypes into some g-sized chunks of adjacent SNPs. g is called the *grain size*. The analysis flow is summarized in the box.

ChrX Regions		Sample Break-down by Ethnicity										
		LWK	MKK	YRI	ASW	GIH	CHB	CHD	JPT	MEX	CEU	TSI
Cg	samples	42	69	90	29	45	39	39	46	28	80	53
SNPs=81	R_M	14	14	16	12	10	10	5	5	5	12	10
Region 6	samples	41	84	87	33	42	45	38	43	46	80	40
SNPs=75	R_M	11	11	11	9	6	5	4	2	5	11	7
Eg	samples	24	65	69	24	35	30	20	27	13	51	32
SNPs=180	R_M	39	53	57	44	26	27	19	21	14	32	24
Dg	samples	37	78	80	32	43	38	30	34	24	78	40
SNPs=92	R_M	25	26	24	21	13	11	10	8	12	12	12
Bg	samples	32	64	78	28	38	41	41	40	22	69	39
SNPs=81	R_M	11	15	15	16	15	11	7	10	13	17	16
NH	samples	41	69	89	29	59	67	62	73	30	92	58
SNPs=72	R_M	12	12	12	12	5	3	2	1	8	10	8

Figure 2: The R_M estimates of the populations in the different ChrX regions.

ANALYSIS FLOW:

(Statistical Analysis of Input)
LOOP
  Choose block size g
  Is this choice statistically sound? (Section 2.1)
    YES: Exit LOOP
END LOOP

(Combinatorial Analysis)
Proceed to use **IRiS** with block size g (Section 2.2)

(Statistical Analysis of Output)
Analyze shared recombinations (Section 3.1)
Analyze recombination per population/sample (Section 3.2)

2.1 Statistical Analysis (using p-value estimations)

Are we justified in using g SNP's as a block (or pattern)? Using k, the number of distinct patterns of the g SNPs across the samples, as a proxy for the extent of LD in this block, we estimate the p-value of k. Loosely speaking, when these g SNPs are in linkage equilibrium (or independent), k should be much larger than when they are in LD. An alternative view is that k is an estimate of the number of *lineages* for the g sized segment of the chromosome.

Let the number of samples be n and let the number of SNPs be N. Further, let V be a column vector of size n. Since the SNPs are assumed to be bi-allelic, V which represents the value of a SNP in the n samples is binary. We use two schemes, based on the mode of definition of the N vectors, to estimates the p-value.

The range of values of k seen in our data is $2 \leq k \leq 16$ and we study the p-value estimates in this range.

1. **RandV:** In this scheme, V_1, V_2, ..., V_N are defined randomly. In other words, each entry of each V is picked independently and uniformly from a set of two alleles. We use 10000 replicates and the distribution of the number of g-sized patterns is shown in Fig 3. The p-values estimated based on this scheme is shown in the table below. The p-values are ≈ 0.0 for every value of k.

2. **PermV:** While the RandV Scheme is not incorrect, we make some domain-dependent modifications to design another scheme. In the PermV scheme we

   (i) mimic the allele frequencies seen in the input data and

   (ii) use the population distribution (by ethnicity) of the screened samples in the chromosomal region.

   This is done as follows: the individual V vectors are plucked from the X-Chromosome of the HapMap III data, (but the SNPs span the entire chromosome) and any untyped SNP (i.e., N in the database) in the vector is given a value in agreement with the allele frequency of that column. Further, we use only those V's that have RAF ≥ 0.1, as is done in the screening process. We again use 10000 replicates and for each replicate, we randomly permute the N vectors. The distribution of the number of g-sized patterns is shown in Fig 3 (b). The p-values estimated based on this is shown in the table below.

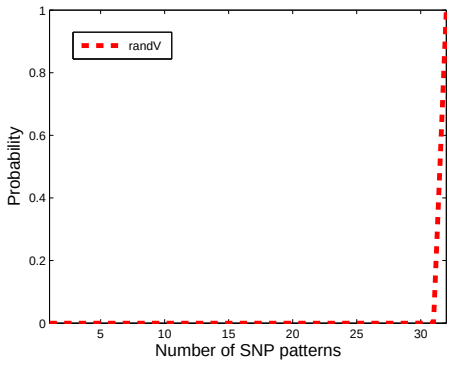
k	2..9	10	11	12	13	14	15	16	17
randV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
permV	0.0	4.8e-7	4.8e-7	9.5e-7	2.4e-6	2.9e-6	1.3e-5	7.8e-5	1.4e-4

k	18	19	20	21	22	23	24	25	26
randV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
permV	1.9e-4	2.3e-4	2.8e-4	4.5e-4	8.9e-4	2.1e-3	5.3e-3	1.0e-2	2.0e-2

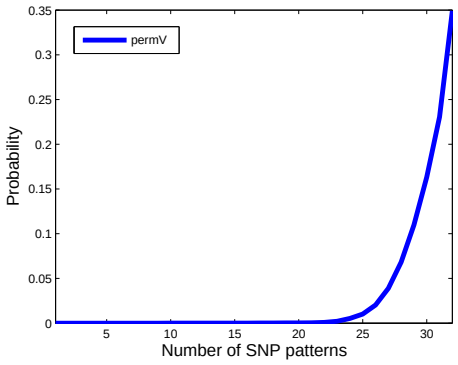
If for a block, k has an insignificant p-value, then the subsequent analysis risks becoming unreliable. We then reduce the grain size. An alternative is to discard the offending SNPs of the block, thus fragmenting the region. In our experiments we used a grain size $g = 5$ and the p-values obtained for this on all the regions were acceptable. The haplotypes are re-coded as sequence of these SNP patterns for the combinatorial analysis.

2.2 Combinatorial Analysis (using IRiS)

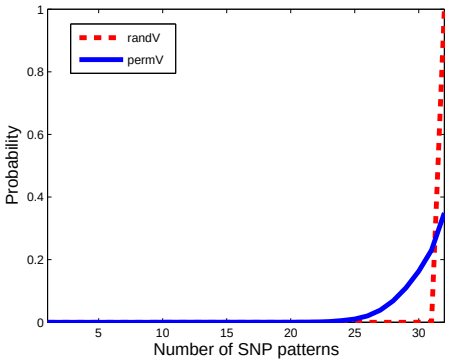
Computational Insights. Note that the general task of computing a phylogenetic network is computationally hard. Striking a balance between computational hardness and biological reality, we use a compatible model that makes our problem tractable. Our approach can be briefly summarized as follows. We first cluster, possibly overlapping, haplotypes that display no evidence of recombinations and a representative haplotype of each cluster is extracted for the next phase. Then exploiting the coherence seen in such data, each haplotype is recoded using blocks of SNPs (patterns seen across different haplotypes) of granularity g. Finally, a network is constructed from the recoded representative haplotypes. Using a divide-and-conquer paradigm, the haplotype is segmented to give simple structures and then these individual structures are merged to give a unified topology using a DSR (Dominant Subdominant Recombinant) Scheme. This gives a plausible explanation of the data through recombinations. The underlying mathematical model along with some tests on simulated data had been earlier presented in [PMC$^+$08]. In a subsequent work, we give a mathematical proof of the effectiveness of our algorithm in terms of distance from the ideal optimal along with results on a three-population HapMap II data [PJM$^+$09]. To avoid digression, some relevant details are presented in the Appendix for the interested reader.



(a) randV Scheme

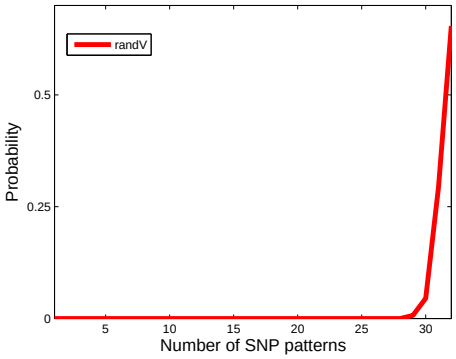


(b) permV Scheme

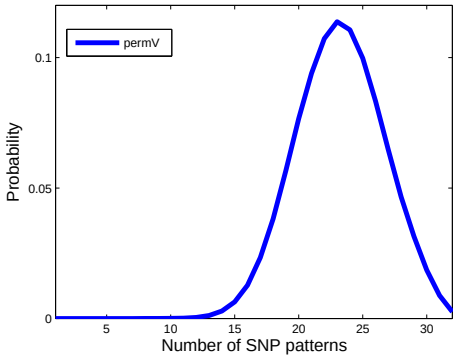


(c) randV and PermV Schemes

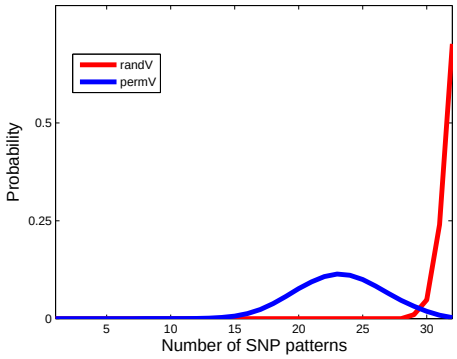
Figure 3: Distribution of k for $g = 5$ for the ChrX region Eg. Recall that the randV Scheme is independent of the region but the permV Scheme is not, since it uses the population distribution of the region (and allele frequency distribution of the entire ChrX) for a more realistic estimation.



(a) randV Scheme



(b) permV Scheme



(c) randV and PermV Schemes

Figure 4: Using HapMap II data.

3 Results

As expected, the application of our method on the X Chromosome regions gives extremely complex network of recombinations on samples. Note that potentially we are exploring L lineages *per sample* where L could be as large as 100 or more. Compare this with *one lineage per sample* in a phylogenetic tree analysis. Due to the enormity and the depth of the available information, we systematically analyze the results at different levels with different perspectives, using well-established and accepted methods (such as MDS, PCA, Mantel Test). Our analysis can be categorized as follows.

1. Populations-specific. We estimate pairwise population distances using recombinations (details in Section 3.1). We use two methods to interpret these distances: (1) visualization of the pairwise distances of the eleven populations using multidimensional scaling (MDS) and (2) compare with traditional F_{ST} distances using Mantel Test. In this context our results are two-fold. Firstly, we demonstrate the presence of recombinations-based evidence in short segments of the genome to detect subcontinental divide in the populations. Secondly, we make the surprising observation that the effect of the population dynamics that shapes the allele-frequency variations between populations is also reflected in the purely recombination-based variations.

2. Recombinations-specific. Using a 11-dimensional vector for each recombination where each dimension denotes a population and each value indicates the number of support of individuals from that population, we do a PCA analysis of the data for each region. Here we again observe the subcontinental separation of the samples within the short segments.

   A next step is to use a M-dimensional vector per recombination where each dimension corresponds to a sample. The results of this analysis is not ready at this time.

Further details on these is described below.

3.1 Recombinational-Distances of Populations

Shared Recombinations. The networks generated by IRiS are analyzed to associate each recombination event with participating populations. If in the set of haplotypes with a recombination event x, there are at least l members of population Z_1 and l of populations Z_2, then recombination event x is said to be *shared* between populations Z_1 and Z_2. In our analysis we set $l = 1$. Similarly, we can extend this notion to sharing of a recombination event between three or more populations.

Recombinational-Distance Matrix D_r Computation. We use the number of recombinations shared by two populations Z_1 and Z_2 as a measure of similarity between Z_1 and Z_2. Let S be the similarity matrix with each element written as s_{ij} denoting the number of shared recombinations between population Z_i and Z_j. Then matrix S is converted to a normalized distance matrix D_r where each entry is written as d_{ij}. Firstly, the similarity is converted to a distance by taking the reciprocal of the value. Roughly speaking, this indicates that larger the number of shared recombination events smaller the distance and vice versa. Note that s_{ii} denotes the number of population-specific recombinations, but this is not used in the definition of d_{ij}. Hence, s_{ii} is set to 0 for subsequent computations. This is then normalized (scaled) with the the minimum of the total number recombinations seen in population Z_i and in population Z_j. Precisely speaking,

$$d_{ij} = \frac{\min\left(\sum_i s_{ij}, \sum_j s_{ij}\right)}{\max(1, s_{ij})}, \qquad \text{for all } i < j. \tag{1}$$

The distance between a population and itself is 0. Thus for all i, $d_{ii} = 0$. The distance is symmetric, i.e. for all i, j, $d_{ij} = d_{ji}$ holds. We give one complete example below for the reader. Since the matrices are symmetric, we show only the upper diagonals of the matrices. This is the distance matrix for the ChrX region Eg.

$$S = \left[\begin{array}{cccccccccccc} \text{LWK} & \text{MKK} & \text{YRI} & \text{ASW} & \text{GIH} & \text{CHB} & \text{CHD} & \text{JPT} & \text{MEX} & \text{CEU} & \text{TSI} & \\ \hline - & 80 & 93 & 60 & 27 & 21 & 15 & 18 & 14 & 28 & 16 & \text{LWK} \\ & - & \mathbf{116} & 77 & 44 & 41 & 31 & 35 & 28 & 47 & 38 & \text{MKK} \\ & & - & 85 & 41 & 42 & 31 & 34 & 26 & 37 & 28 & \text{YRI} \\ & & & - & 44 & 33 & 29 & 29 & 26 & 41 & 32 & \text{ASW} \\ & & & & - & 43 & 41 & 39 & 26 & 40 & 40 & \text{GIH} \\ & & & & & - & 45 & 49 & 22 & 33 & 31 & \text{CHB} \\ & & & & & & - & 39 & 19 & 21 & 23 & \text{CHD} \\ & & & & & & & - & 22 & 29 & 28 & \text{JPT} \\ & & & & & & & & - & 26 & 27 & \text{MEX} \\ & & & & & & & & & - & 45 & \text{CEU} \\ & & & & & & & & & & - & \text{TSI} \end{array}\right].$$

As a concrete example we compute the distance between populations MKK and YRI d_{23} using s_{23}. Note that S is a symmetric matrix although we do not display all the values here. The diagonal entries for the calculation purposes are set to zero. Then

$$\sum_i s_{i2} = 537, \quad \sum_i s_{i3} = 533,$$

$$d_{23} = \frac{\min(537, 533)}{116} = 4.5948 \approx 4.60$$

Recombinational-distance matrix D_r for the similarity matrix S is shown below.

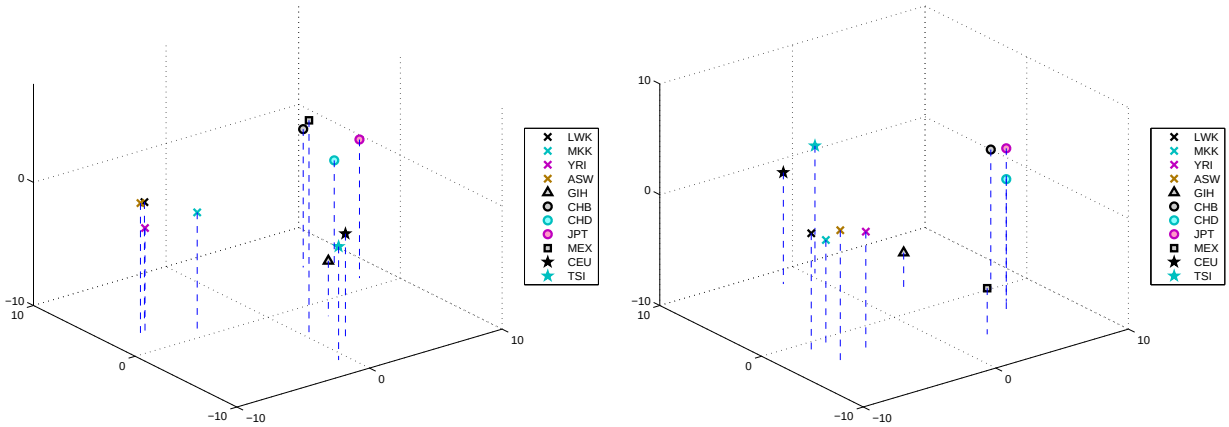
$$D_r = \left[\begin{array}{cccccccccccc} 0 & 4.65 & 4 & 6.2 & 13.78 & 17.14 & 19.6 & 17.89 & 16.86 & 12.40 & 19.25 \\ & 0 & \mathbf{4.60} & 5.92 & 8.75 & 8.78 & 9.49 & 9.2 & 8.43 & 7.38 & 8.11 \\ & & 0 & 5.36 & 9.39 & 8.57 & 9.48 & 9.47 & 9.08 & 9.38 & 11.00 \\ & & & 0 & 8.75 & 10.91 & 10.14 & 11.10 & 9.08 & 8.46 & 9.63 \\ & & & & 0 & 8.37 & 7.17 & 8.26 & 9.08 & 8.68 & 7.7 \\ & & & & & 0 & 6.53 & 6.57 & 10.73 & 10.52 & 9.94 \\ & & & & & & 0 & 7.54 & 12.42 & 14 & 12.78 \\ & & & & & & & 0 & 10.73 & 11.10 & 11.00 \\ & & & & & & & & 0 & 9.08 & 8.74 \\ & & & & & & & & & 0 & 6.84 \\ & & & & & & & & & & 0 \end{array}\right].$$

3.1.1 Multidimensional Scaling (MDS) Visualization

Figs 5, 6, 7 give the Multidimensional Scaling (MDS) analysis done using MATLAB. The figures show the subcontinental separation in the regions.

3.1.2 Comparison with F_{ST} Distances (Mantel Test)

Next for each region, we compare distance matrix D_r (defined in Section 3.1) with the F_{ST} measure that *Arlequin* software (`http://cmpg.unibe.ch/software/arlequin3/`) computes on the 11 populations, using Mantel test [SR95] with 10000 replicates. The results are shown in Fig 8.



(1) Region 1: stress 3.7%

(2) Region 2: stress 6.4%

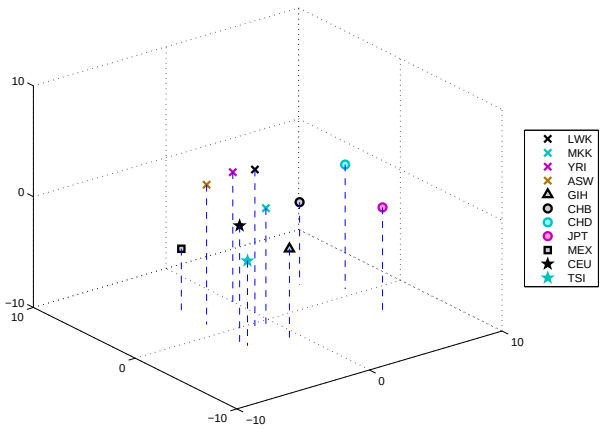
(3) Region 3: stress 6.1%

(4) Region 4 (Cg): stress 3.5%

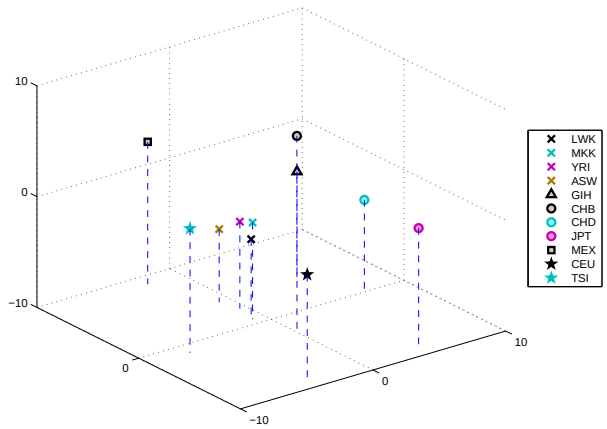
(5) Region 5: stress 7.6%

(6) Region 6: stress 2.6%

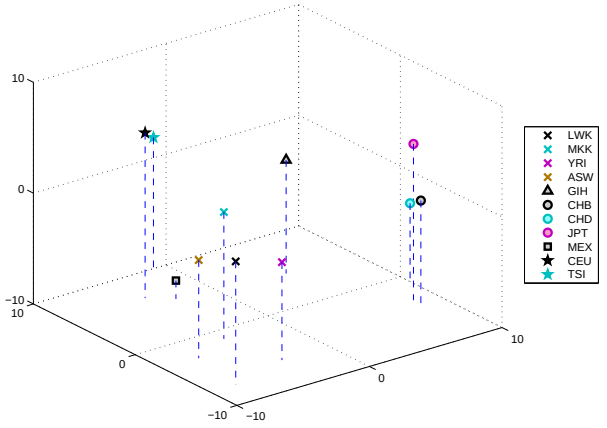
Figure 5: MDS plots of ChrX regions 1-6.



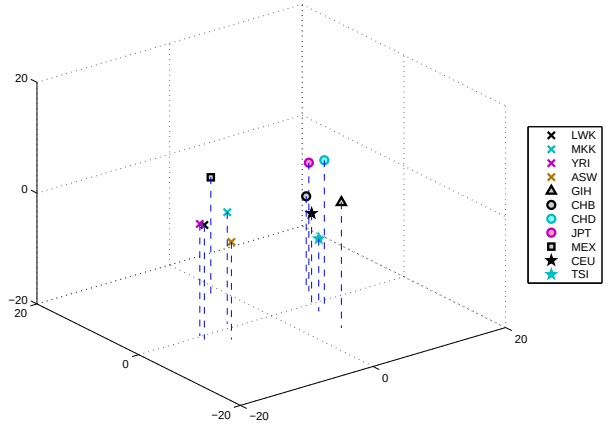
(1) Region 7: stress 4.8%



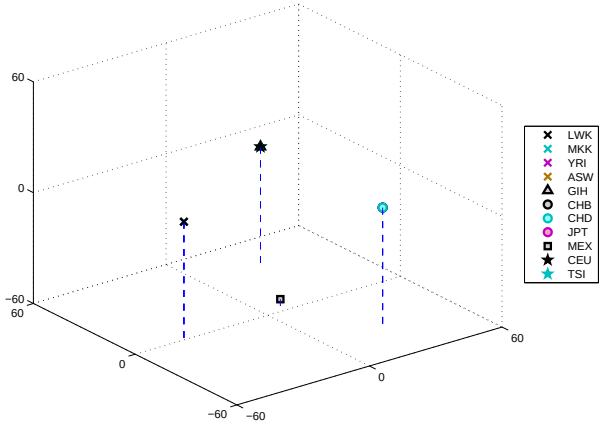
(2) Region 8: stress 5.6%



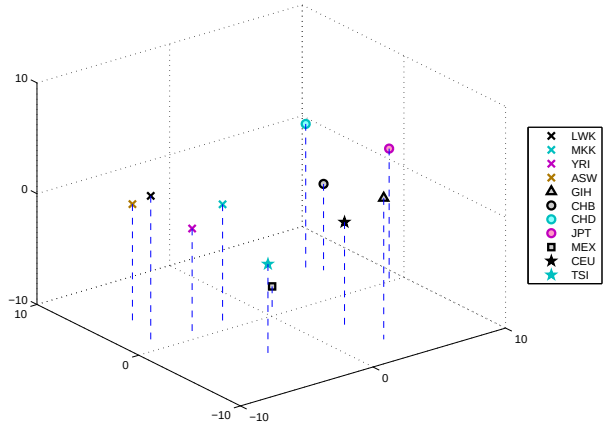
(3) Region 9 (Eg): stress 5.5%



(4) Region 10 (Dg): stress 3.2%

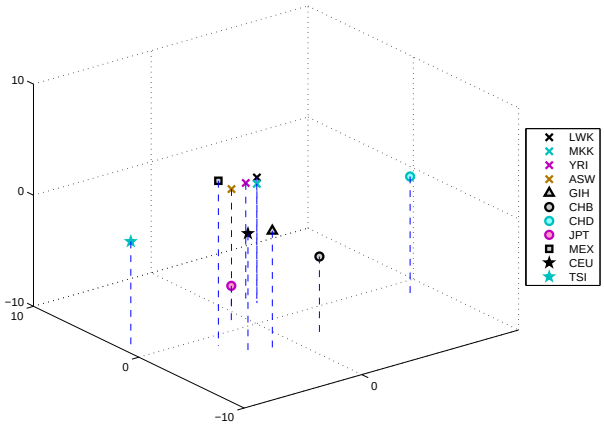


(5) Region 11: stress 0.014%
(some numerical instability in the MDS software)

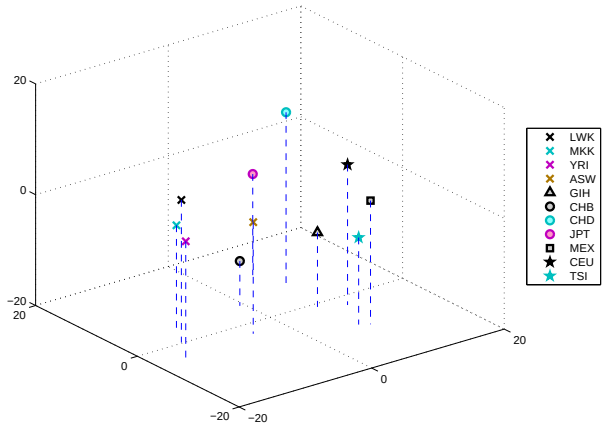


(6) Region 12: stress 4.7%

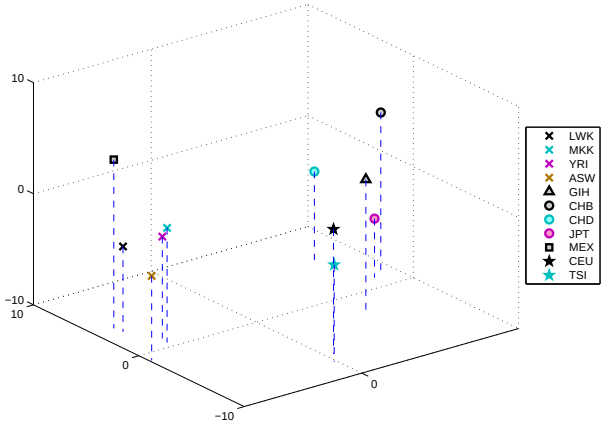
Figure 6: MDS plots of ChrX regions 7-12.



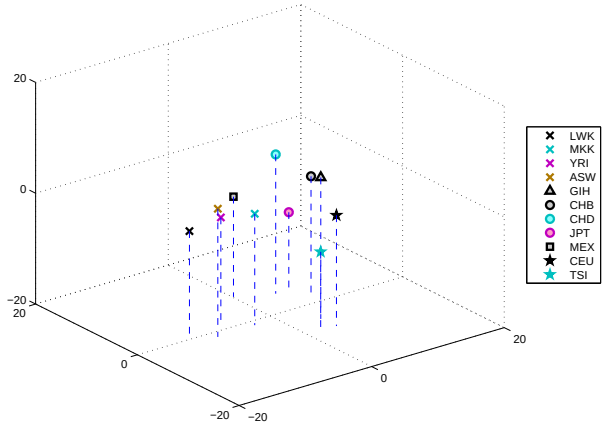
(1) Region 13 (Bg): stress 8.6%



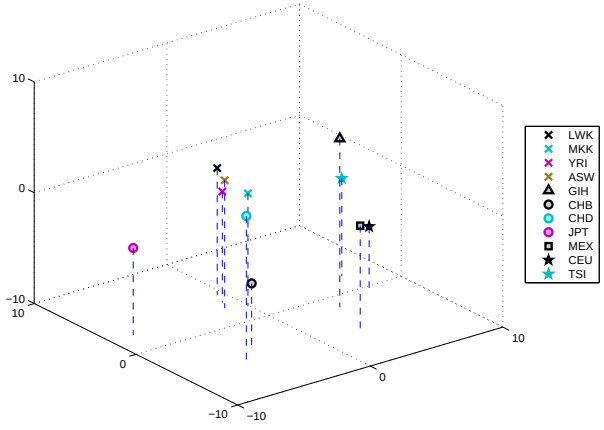
(2) Region 14 (NH): stress 4.7%



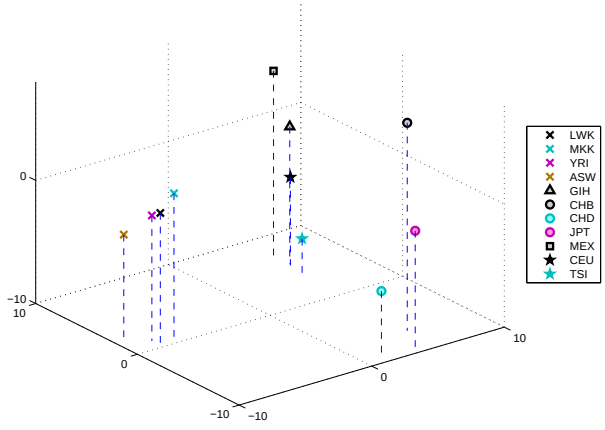
(3) Region 15: stress 6.1%



(4) Region 16: stress 6.0%



(5) Region 17: stress 6.5%



(6) Region 18: stress 6.3%

Figure 7: MDS plots of ChrX regions 13-18.

		SNPs			Recombs		IRiS	Mantel Test		MDS
Label	size (bp)	No.	density	Smp	ρ	H	#	r	p-value	stress
1	222643	95	1/2394	477	3.608	5	365	0.721	10^{-4}	3.7
2	141256	96	1/1456	504	3.355	2	243	0.645	10^{-4}	6.4
3	300770	83	1/3957	505	0.660	4	67	0.455	2.8×10^{-3}	6.1
4(Cg)	466115	81	1/5755	560	0.603	4	94	0.566	5×10^{-4}	3.5
5	604607	179	1/3378	490	1.237	9	225	0.720	$< 10^{-6}$	7.6
6	247979	80	1/3306	579	0.255	2	24	$-.044$	0.5733	2.6
7	270962	90	1/3188	452	1.604	2	104	0.741	$< 10^{-6}$	4.8
8	549992	204	1/2736	406	1.691	8	171	0.368	8.8×10^{-3}	5.6
9(Eg)	1032833	180	1/5738	392	0.895	6	254	0.542	3×10^{-4}	5.5
10(Dg)	421406	92	1/4581	514	0.375	2	116	0.630	1.1×10^{-3}	3.2
11	234388	82	1/2894	527	1.849	2	128	0.146	0.158	.014
12	574608	157	1/3831	431	1.335	7	258	0.611	10^{-3}	4.7
13(Bg)	468827	81	1/5787	492	0.526	3	75	0.566	$< 10^{-6}$	8.6
14(NH)	392907	72	1/5457	669	0.182	0	32	0.480	2.3×10^{-3}	4.7
15	166712	99	1/1755	433	5.122	3	243	0.568	6×10^{-4}	6.1
16	270741	88	1/3148	460	3.847	4	214	0.576	2×10^{-4}	6.0
17	332852	96	1/3579	509	1.533	6	202	0.531	$< 10^{-6}$	6.5
18	497607	162	1/3110	446	2.106	4	261	0.435	8×10^{-4}	6.3
	7197205						3076			

Figure 8: Brief descriptions of the ChrX regions and evaluation of our method. Recombination rate ρ and the number of hotspots H is computed using LDhat. IRiS # is the number of recombinations detected by IRiS. The two evaluations are: (1) Mantel Test with the recombination matrix D_r (correlation coefficient is shown as r) and (2) stress factor in the MDS analysis.

3.2 Specificity Analysis

Recall that we have recovered, through IRiS estimation of recombinations, a population structure that fits what is known through traditional population genetic analysis. Now we go further in recovering the specificity of each population given by the repertoires of chromosomes (samples) harboring particular sets of recombinations. There are several ways to organize the output of IRiS. In any case, the output is organized in the two formats below.

- Format 1: A 11-dimensional vector for each recombination where each dimension denotes a population and each value indicates the number of support of individuals from that population. As the number of samples varies among the populations, a direct comparison may be misleading. We experiment with three ways of accounting for this non-uniformity in the input data. Each entry (i, j) in the matrix is scaled by $1/x_j$. In Scheme I, x_j is the number of samples in population j. In Scheme II, x_j is the sum of the entries of column j in the matrix. In Scheme III, x_j is the number of non-zero entries in column j of the matrix. We found that Scheme II and III work well and give similar results. We adopted Scheme II for the analysis shown in supplement.

- Format 2: An M-dimensional vector per recombination where each dimension corresponds to a sample. This format is more exhaustive and will have a wider interest in the complete analysis of historical recombinations.

It is important to note that due to the limited number and the sampling process, the presence/absence of recombination events in the population may not be interpreted as deterministic markers.

Population-Overlap Study. Here we use heuristics to ascertain the significance of a recombination and display the distribution of these significant recombinations in the populations in a few regions. This is done primarily to gain an understanding of the overlaps of the populations in terms of recombination events. The table confirms the inherent complexity of the recombination networks. The subcontinental-population labels of Section 1 are used here.

ChrX Regions→	1	2	3	4	5	9	10	12	18	total
Af	65	69	12	26	70	63	40	60	70	475
sA									1	1
eA	2	3	2	5	1	3	1	1	1	19
Am							1		1	2
Eu		4				2		3	2	11
Af & sA	1	0	1		1				1	4
Af & eA	3	2	1		1		8	1	3	19
Af & Eu	4	6	3		7	3	1	2	2	28
sA & eA	1					1		2		4
sA & Am	1							2		3
sA & Eu				1		2		1	1	5
eA & Eur	1					1				2
eA & Am							1	2		3
Am & Eu	1									1
Af & eA & sA	2	4	1		1	3			1	12
Af & Eu & sA	2	1		1	3	2		3	3	15
Af & eA & Am		1	1		1			1	1	5
Af & eA & Eu	5	3			4			2	2	16
Af & Am & Eu	10				2			1		13
sA & Eu & eA	2						1	1	1	5
Af & eA & sA & Eu	2	2			6	3	3		5	21
sA & Eu & eA & Am	1		1		2			1		5
sA & Eu & Af & Am			1	1	1					3
sA & Eu & Af & Am & Eu	8	3	3	5	12	13	4	4	8	60

Multivariate Analysis. Here there are several possibilities: principal component analysis (PCA), correspondence analysis (as the data is a frequency table), discriminant analysis or more genetic-centered like STRUCTURE (`http://pritch.bsd.uchicago.edu/structure.html`) . The last two require data in format 2.

We show the results for regions 1 and 2: see figures at the end of the document. The details of Region 1 is discussed here and the same follows for Region 2. PCA gives gives us powerful results, with very strong discrimination both at continental as well as population level. The first principal component clearly separates Africans from non-Africans, with 27% of the variance explained while the second principal component, with 16% makes a good separation of MKK from the other three African populations. This result has also been found in traditional genetic analysis based on allele frequency or those based on phylogeography. The third component, with 13% of variance, separates at the same time MKK from other Africans and Asian populations from the rest on non-Africans. There is low discrimination between TSI (Italy) and GIH (India) and between CEU (North Europe) and MEX (Mexico).

Population specificity can be achieved, nonetheless just by taking the next components. Considering those that are significant in the PCA, shown in the table, it is possible to see how particularly each population is shaded by the most important contributions to the component. Thus the frequencies that each recombination is found in each population gives a clear picture of population differentiation.

To see at a glance both effects, of population differentiation and the contribution of each of the detected

recombination, a correspondence analysis has also been done and the plotted results can be interpreted in a similar way. In the representation of dimensions 1, 2 and 3 (see figures), the clouds of recombination events show how they pull the populations to segregate among them.

To conclude, results shown graphically are nothing but a simplification of the numerical analysis in which all the significant factors are considered simultaneously.

(Results from the analysis of data in Format 2 are not ready)

4 General Discussion

Here we comment on existing notions and concepts along with our observations based on insights gained through this work.

4.1 Recombination Hotspots: Friend or Foe?

The occurrences of meiotic recombinations in the human genome (and some other genomes) is not uniform, but rather there are regions called *hotspots* (usually 1-2 kb in width) where the frequency of recombination is 10 to several thousand times higher than the average in the background, and almost all recombination events happen within them [LZZ06]. Recent studies have shown that hotspots are a ubiquitous feature of the human genome, and recombination hotspots are also the main contributor of the block-like pattern of haplotypes. So the pattern (blocks) that we exploit in the IRiS model owes its existence to hotspots.

In Fig 9 we compare two existing popular methods of computing recombination rates. We selected a fixed region in ChrX. In the first, we transposed the results from HapMap II from the website onto our ChrX regions in the HapMap III database. In the second we used the PHASE algorithm:
`http://stephenslab.uchicago.edu/software.html`.

The common intuition is that recombination hotspots must confound any recombinational analysis. At this stage of our work, it is not clear to us that it has affected our analysis- we do recognize that the number of recombination events estimated by our analysis may actually be a gross underestimate around the hotspots. Further we expect high p value estimates of Section 2.1 of the g blocks if a hotspot region displays a likelihood of very many lineages.

4.2 F_{ST} Adjustment under LD (dependence) Assumptions

The F_{ST} distance measure comes in various forms [JHTS04]. Basically the difference in computation lies in the details of the measure of allele frequency differences in two populations. There is an underlying assumption that, any pair of alleles is independent and thus a simple aggregation over all the alleles suffice.

The very basic premise of our model is that there is a fair chance of two alleles being dependent (having the same lineage or identical genealogical history), at least in a population. This is also seen through our R_M analysis of the individual populations in Fig 2. It is unclear whether this F_{ST} estimate is sufficiently accurate for this analysis.

Our suggestion for modifying the to F'_{ST} is as follows. Let $b_1, b_2, \ldots, b_K$ be blocks of alleles such that each block b_i has a high LD (or estimated to have the same genealogical history, say through a T_M analysis) in at least one of the populations. For example for the two populations CHB and CHD, the blocks in each population are as follows:

CHB: 3 20 26 27 43 45 51 66 80 83 91

CHD: 3 6 10 45 47 54 66 83 85 91

Here the numbers refer to the position of the cluster boundary in terms of SNP blocks. Thus the first cluster of three blocks occurs in both the populations. The next two cluster boundaries are at positions 6 and 10 in

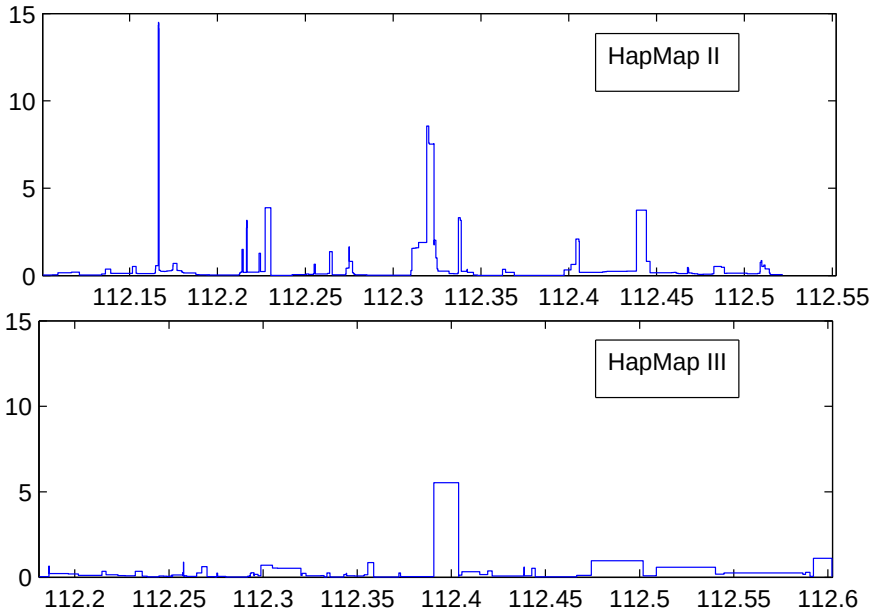


Figure 9: Comparison of the recombination rate ρ for the region Dg using two sources. HapMap II result is the one posted on HapMap website where software LDhat has been used. HapMap III results are computed using the software Phase. HapMap II uses Build 35 and HapMap III uses Build 36, hence the positions marked along the x-axix in the two plots do not match exactly but the SNPs have been vertically aligned (using the RS ids).

the second population followed by position 20 in the first population. Proceeding in this manner, the blocks for the two populations are:

CHB or CHD: 3 6 10 20 26 27 43 45 47 51 54 66 80 83 85 91

In this case $K = 16$. Then the F_{ST} distance is adjusted as follows:

$$F'_{ST} = \frac{\sum_i F_{ST}(b_i)}{K},$$

where $F_{ST}(b_i)$ is the usual F_{ST} distance between the two populations restricted to the alleles in the block b_i. Roughly speaking, this will adjust the biases due to strong LD segments (or blocks).

Acknowledgments We thank Hafid Laayouni for his efforts on the PCA analysis. We gratefully acknowledge Ajay Royyuru's enthusiastic support, and more, of the work.

References

[FSF+04] Roubinet F, Despiau S, Calafell F, Jin F, Bertranpetit J, Saitou N, and Blancher A. Evolution of the O alleles of the human ABO blood group gene. *Transfusion*, 44(5):707–15, 2004.

[GSN+02] Stacey B. Gabriel, Stephen F. Schaffner, Huy Nguyen, Jamie M. Moore, Jessica Roy, Brendan Blumenstiel, John Higgins, Matthew DeFelice, Amy Lochner, Maura Faggart, Shau Neen Liu-Cordero, Charles Rotimi, Adebowale Adeyemo, Richard Cooper, Ryk Ward, Eric S. Lander,

Mark J. Daly, and David Altshuler. The structure of haplotype blocks in the human genome. *Science*, 296(5576):2225 – 2229, 2002.

[HSW05] Jotun Hein, Mikkel H. Schierup, and Carsten Wiuf. *Gene Genealogies, Variation and Evolution: A Primer in Coalescent Theory*. Oxford Press, 2005.

[JHTS04] M.A. Jobling, M. Hurles, and C. Tyler-Smith. *Human Evolutionary Genetics: Origins, Peoples and Disease*. Mathematical and Computaional Biology Series. Garland Publishing, 2004.

[LZZ06] Jun Li, Michael Q. Zhang, and Xuegong Zhang. A new method for detecting human recombination hotspots and its applications to the HapMap ENCODE data. *Am. J. Hum. Genet*, 79:628, 2006.

[MMH+04] Gilean A. T. McVean, Simon R. Myers, Sarah Hunt, Panos Deloukas, David R, Bentley, and Peter Donnelly. The fine-scale structure of recombination rate variation in the human genome. *Science*, 304(5670):581–584, 2004.

[PJM+09] Laxmi Parida, Asif Javed, Marta Melé, Francesc Calafell, Jaume Bertranpetit, and Genographic Consortium. Minimizing recombinations in consensus networks for phylogeographic studies. *to appear in BMC Bioinformatics*, 2009.

[PMC+08] Laxmi Parida, Marta Melé, Francesc Calafell, Jaume Bertranpetit, and Genographic Consortium. Estimating the ancestral recombinations graph (ARG) as compatible networks of SNP patterns. *Journal of Computational Biology*, 15(9):1–22, 2008.

[SR95] R. R. Sokal and F. J. Rohlf. *Biometry*. Mathematical and Computaional Biology Series. New York: Freeman, 1995.

Appendix

Our Underlying Mathematical Model

We propose a model, that exploits the contiguous patterns in the extant sequences to construct a phylogenetic network. A striking difference from the other network models is that ours permits multiple roots. Further, a recombination is modeled as a hybrid of stretches (or *segments*) of parental sequences. We do this by extending the notion of *compatible trees* to networks that models recombinations in a fairly generalized form. The network is defined in terms of a segmentation S of the aligned extant sequences into say K segments. Every edge in the network is associated with at least one segment (from the K segments). When the network is restricted only to the edges of fixed segment, the resulting topology is a phylogenetic tree of only that segment from all the extant sequences.

We formally define the compatible network here. Let I be the given input matrix with n rows and m columns. Each row corresponds to an extant sequence or haplotype. The matrix I has two kinds of elements, *solid characters* and dont-care. As our model will not only consider single character (where it would be normal to use the four nucleotides `A`, `C`, `G`, `T`) but combinations of them (see *grain* below) the nomenclature is not restricted to four states. A dont-care is written as '-' and its semantics will be discussed later. Further, for ease of exposition, let all occurrences of a character c be within a single column of I.

The *segmentation* S of the m columns of input I, written as the closed interval $[1, m]$, is a collection of non-overlapping intervals such that each column j is in at most one segment. For example, given 5 columns, a possible segmentation of $[1, 5]$ is:

$$S = \{[1, 2], [3, 4], [5, 5]\}$$

For convenience, the three segments are denoted simply by integer labels 1, 2 and 3.

A *compatible network* N is a directed acyclic graph (DAG) that explains input I with a segmentation S. N is defined as follows: It has three kinds of nodes. A node with no incoming edge is a *root* node and N may have multiple root nodes. A node with no outgoing edges is a *leaf* node and there can be no more than n leaf nodes in N. Every other node is an *internal* node. An internal node has at most two incoming edges. When a node has exactly one incoming edge, it is called a *mutation node* and the incoming edge is called a *mutation edge*. When the node has two incoming edges, the node is called a *recombination* or a *hybrid* node and the incoming edges are called *recombination edges*. A mutation edge is labeled with element(s) from the matrix I. A recombination edge is labeled with segment(s) of S.

Each node in N is labeled by a sequence of length m. Let e be the mutation edge coming into node v with labels $c_1, c_2, \ldots c_l$. Let c_i occur in column j_i of I. Then the label of v is obtained from the label of its only parent, by replacing positions $j_1, j_2, \ldots, j_l$ with values $c_1, c_2, \ldots, c_l$ respectively.

Let e_1 and e_2 be two recombination edges coming into a node v with segment labels $s_1, s_2, \ldots s_{l_1}$ and $r_1, r_2, \ldots r_{l_2}$ of parents 1 and 2 respectively. The label of node v is a hybrid sequence with segments $s_1, s_2, \ldots s_{l_1}$ from the label of the first parent and segments $r_1, r_2, \ldots r_{l_2}$ from the label of the second parent. Each position in the missing segments, i.e. neither from parent 1 nor from parent 2, in the label is written as ϕ. The interpretation of ϕ is that it is a sort of a filler, that is not reflected in any way in the extant sequences and could be ignored for all practical purposes. In genetic terms these are partial sequences that have been lost and their existence is known because only part of it (a recombinant fragment) has reached the present. An interesting example is found in the ABO sequence variations [FSF+04].

Next, the network N is compatible with input I with segmentation S if the two conditions hold:

(1) the label of a leaf node corresponds to a row in matrix I and every row in matrix I is the label of some leaf node in N.

(2) For each column j in I, every solid character (of j) occurs exactly once in the label of some mutation edge in N.

The Compatible Network Construction Problem

For a segment $s \in S$, $Restricted(N, s)$ is the network obtained by doing the following two operations. (1) Removing all recombination edges that do not have the label s. (2) Let character c occur in column j and $j \notin s$, then the label c can be removed from the mutation edge label.

Fact 1 *For each segment $s \in S$, $Restricted(N, s)$ is a forest, i.e., each connected component is a tree.*

$L(N, s, c)$ is the collection of rows corresponding to leafnodes reachable from node v in $Restricted(N, s)$ where v has an incoming mutation edge with the label c. Note that for a fixed c, the node v is unique in a compatible network. We pose the following optimization problems.

Problem 1 *(Minimal Segmentation) Given I, the task is to compute a compatible network N with minimum number of elements in segmentation S of I.*

Problem 2 *(Minimal Recombination) Given I, the task is to compute a compatible network N with minimum number of recombinations in N.*

Note that the number of recombinations in a compatible network is at least $K - 1$, where K is the number of segments in the segmentation S.

Fact 2 *(Nonuniqueness) Two distinct segmentations $S \neq S'$ can give distinct compatible networks N and N'. Moreover, it is possible that a segmentation S, can give two distinct compatible networks N_1 and N_2.*

Fig **??** shows a character matrix I and two distinct compatible networks for the same segmentation S (the interested reader can see an example in Fig **??**, where two distinct segmentations of the same size ($K = 2$) is possible for the same character matrix).

We propose to tackle Problem 1 in a three step process: In the first step we transform the input haplotypes into a character matrix; in the second step we split this character matrix into segments where a phylogenetic forest can explain each segment and then in the final step construct the compatible network from the forests. The details follow.

(Step 1) Staging the Input: Haplotypes to 'Character' Matrix

The input is a collection of haplotypes, in the form of a matrix I' where each row is an ordered vector of SNP values as they appear along a chromosome. A running example of 25 haplotypes each with 85 SNP's is shown in in Fig 10. A reference sequence is shown at the top. The asterisk in the haplotype denotes agreement with the reference sequence.

We process this data into a smaller matrix (I) of blocks of SNPs that we call *characters* (not unlike taxonomic characters) since each column may now take on multiple values and further, the order is unspecified. This processing is carried out in the following three stages.

1. Removing redundancies: Without loss of generality, no two rows are identical in I'. However, since our interest is in recombinations, identical columns are not considered redundant for the purpose of extracting the topology of the phylogenetic network. Note, however that the information regarding the *redundant* rows is retained for the final analysis.

	12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
Ref	GCCCCCACGGAGGCGGTGCCGGCGCCTGCTAGCAGACCTGAAGGGTCAGAGACCGATACGGCAGCCGCTAGCGCGCTCATGCCGG
1	CT*G***T*A*****ACCGGTC***G****CT**************GA************************A********AA
2	CT*G***T*A*****ACCGGTC***G*****T**************GA************************A********AA
3	CT*G***T*A*****A*CGGTC***G*****T**************GA************************A********AA
4	CT*G***T*A*****ACCGGTC***G*****T**************GA************************A********AA
5	CT*G***T*A*****ACCGGTC***G****CT**************G************************A********AA
6	CT*G***T*A*****ACCGGTC***G*****T**************G************************A********AA
7	***A******GA*****TGGCC***GC******GA***C**************T*********A******A**********AA
8	***A*******A*T***TGGCT**GGC***AA
9	***A*******A*T***TGGCT**GGC********************A********************************AA
10	***A*******A*T***TGGCT**GGC********************A********************************AA
11	***A*******A*T***TGGCT**GGC***************CA****A********************************AA
12	***A*******A*T***TGGCT**GCC***************CA****A********************************AA
13	***A*******A*T***TCCCT**GCC*************************************G*****************AA
14	***A*******A*T***TCCCT**GGC******C*************************G*****************AA
15	***A*******A*T***TGGCTT*GGC******C*************************G***************CCC*A*
16	***A*******A*T***TGGCT**GGC***A*
17	***A*******A*T***TGGCT**GGC***AA
18	***A*******A*T***TGGCT**GGC***AA
19	***A*******A*T***TGGCT**GGC***********A********************A*****C*CG****A***G*****A
20	***A*******A*T***TGGCT**GGC***********A********************A*****C*CG****A***G*****A
21	***G*******AA*A**TGGCC***GC******G*****A********************A*****C*CG****A***G*****A
22	***A*******A*T***TGGCC**GGCA**********C**G*****************A****TC*C*****A*********A
23	***A*******A*T***TGGCC**GGCA**********C**G*****************A****TC*C*****A*********A
24	***A*******A*T***TGGCC**GGCA*****CA***C**G*****************A****TC*C*****A*********A
25	***A*******A*T***TGGCC**GGCA*****CA***C**G*****************A****TC*C*****A*********A

(1) I': Input matrix of SNPs.

	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	0	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
7	2	2	2	0	1	3	2	1	2	1	3	0	2	1	1	2	2	0	1	1	1	0	1	0	2	1	1	1	1
8	2	2	2	3	2	3	2	2	3	1	3	1	1	1	1	2	2	1	1	1	1	1	1	1	2	1	1	1	1
9	2	2	2	3	2	3	2	2	3	1	3	1	1	1	1	2	1	1	1	1	1	1	1	1	2	1	1	1	1
10	2	2	2	3	2	3	2	2	3	1	3	1	1	1	1	2	1	1	1	1	1	1	1	1	2	1	1	1	1
11	2	2	2	3	2	3	2	2	3	1	3	1	1	1	2	2	1	1	1	1	1	1	1	1	2	1	1	1	1
12	2	2	2	3	2	3	2	2	4	1	3	1	1	1	2	2	1	1	1	1	1	1	1	1	2	1	1	1	1
13	2	2	2	3	2	3	3	2	4	1	3	1	1	1	1	2	2	1	1	1	2	1	1	1	2	1	1	1	1
14	2	2	2	3	2	3	3	2	3	1	3	3	1	1	1	2	2	1	1	1	2	1	1	1	2	1	1	1	1
15	2	2	2	3	2	3	2	0	3	1	3	3	1	1	1	2	2	1	1	1	2	1	1	1	2	1	0	0	2
16	2	2	2	3	2	3	2	2	3	1	3	1	1	1	1	2	2	1	1	1	1	1	1	1	2	1	1	1	2
17	2	2	2	3	2	3	2	2	3	1	3	1	1	1	1	2	2	1	1	1	1	1	1	1	2	1	1	1	1
18	2	2	2	3	2	3	2	2	3	1	3	1	1	1	1	2	2	1	1	1	1	1	1	1	2	1	1	1	1
19	2	2	2	3	2	3	2	2	3	1	3	1	1	2	1	2	2	1	1	1	3	1	2	3	1	1	3	3	1
20	2	2	2	3	2	3	2	2	3	1	3	1	1	2	1	2	2	1	1	1	3	1	2	3	1	1	3	3	1
21	2	1	2	3	0	3	2	1	2	1	3	0	1	2	1	2	2	1	1	1	3	1	2	3	1	1	3	3	1
22	2	2	2	3	2	3	2	1	3	2	3	1	2	3	1	2	2	1	1	1	3	3	2	1	1	1	1	3	1
23	2	2	2	3	2	3	2	1	3	2	3	1	2	3	1	2	2	1	1	1	3	3	2	1	1	1	1	3	1
24	2	2	2	3	2	3	2	1	3	2	3	5	2	3	1	2	2	1	1	1	3	3	2	1	1	1	1	3	1
25	2	2	2	3	2	3	2	1	3	2	3	5	2	3	1	2	2	1	1	1	3	3	2	1	1	1	1	3	1

(2) I'': Blocking $g = 3$ SNPs of I'.

cluster	1	2	3	4	5	6	7	8	910	1	2	3	4	5	6	7	8	920	1	2	3	4	5	6	7	8	9	(rows of I'')
1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	(2 3 4)
2	2	2	2	3	2	3	2	1	3	0	3	1	2	0	1	2	2	1	1	1	3	0	2	1	1	1	1	3 1 (22 23 24 25)
3	2	2	2	3	2	3	2	2	3	1	3	1	1	1	1	2	1	1	1	1	1	1	1	1	2	1	1	1 1 (9 10)
4	2	2	2	3	2	3	2	2	3	1	3	1	1	1	1	2	2	1	1	1	1	1	1	1	2	1	1	1 1 (8 17 18)
5	2	2	2	3	2	3	2	2	3	1	3	1	1	2	1	2	2	1	1	1	3	1	2	3	1	1	3	3 1 (19 20)
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1 (1)
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1 (5)
8	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1 (6)
9	2	2	2	0	1	3	2	1	2	1	3	0	2	1	1	2	2	0	1	1	1	0	1	0	2	1	1	1 1 (7)
10	2	2	2	3	2	3	2	2	3	1	3	1	1	1	2	2	1	1	1	1	1	1	1	1	2	1	1	1 1 (11)
11	2	2	2	3	2	3	2	2	4	1	3	1	1	1	2	2	1	1	1	1	1	1	1	1	2	1	1	1 1 (12)
12	2	2	2	3	2	3	3	2	4	1	3	1	1	1	1	2	2	1	1	1	2	1	1	1	2	1	1	1 1 (13)
13	2	2	2	3	2	3	3	2	3	1	3	3	1	1	1	2	2	1	1	1	2	1	1	1	2	1	1	1 1 (14)
14	2	2	2	3	2	3	2	0	3	1	3	3	1	1	1	2	2	1	1	1	2	1	1	1	2	1	0	0 2 (15)
15	2	2	2	3	2	3	2	2	3	1	3	1	1	1	1	2	2	1	1	1	1	1	1	1	2	1	1	1 2 (16)
16	2	1	2	3	0	3	2	1	2	1	3	0	1	2	1	2	2	1	1	1	3	1	2	3	1	1	3	3 1 (21)

(3) I: Clustering of the rows of I''.

Figure 10: The steps involved in processing input SNP matrix I' to obtain the character matrix I''. See text for details and Fig 11 for the explanation of one of the clusters represented in I.

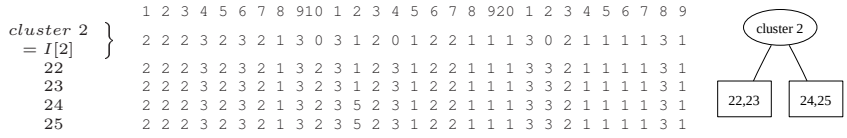


Figure 11: Possible evolutionary history of cluster 2. The putative root node (vector) for this history is row 2 of I of Fig 10.

2. Grouping the columns (grain *g*): A fixed number, g, of consecutive columns of I' are blocked together to obtain I''. Thus, each column of I'' represents a block of g consecutive SNPs from the input matrix I'. Each distinct pattern in I'' is assigned a distinct integer label. Note that for the purposes of extracting the topology, it is adequate to distinguish different patterns in a single column. Thus a label '2' in column one may represent an entirely different pattern from label '2' in column four. (Note that we used uppercase alphabets for these labels, instead of integers, in the previous section for ease of exposition.)

In Fig 10 a grain size of $g = 3$ is used on I' to give a 25×29 matrix I''. Each column represents a block of three consecutive SNPs (except the very last column which has only one SNP). Each distinct pattern is assigned a distinct integer. For example, in column 1, pattern `CTC` (or `CT`$\star$ in I') is assigned integer 1; pattern `GCC` (or $\star\star\star$ in I') is assigned integer 2.

Wild character `0` *in* I'. A unique pattern in a column is assigned 0. Note that a single column may have multiple 0's. For example, in Fig 10 (2), column 12 has a value 0 at rows 7 and 21. The pattern `GAA` appears only in row 7 and the pattern `GGA` appears only once in row 21 in that column.

3. Clustering the rows: In this step, we reduce the number of rows of I'' by identifying clusters of rows that (possibly) do not have any recombinations in their evolutionary history.

> Each cluster is represented by a single vector that is the putative root node of the evolutionary history (tree) of the cluster elements.

We use a greedy iterative algorithm to compute the clusters. This is based on uninterrupted patterns across the rows of the matrix and the process is summarized below.

Wrapper Procedure:

(Step 1) In each column of I'', replace a unique entry by 0.

(Step 2) Collapse I'' to $\breve{I}$, where a row in $\breve{I}$ is the putative root of a cluster of rows of I'' (see below).

(Step 3) IF I'' is identical to $\breve{I}$, terminate the process

ELSE Update $\breve{I}$ to I'' and go to Step 1.

The pseudcode for *Collapse*, the process of clustering multiple rows into one putative root vector, is given below. This is a recursive process and the very first call is made with $c = 1$; r_{max} is the number of rows and c_{max} is the number of columns in I; and pat is initialized to the empty string ϕ. The concatenation of string pat followed by string p is written as $pat \oplus p$: this is the putative root vector that is built incrementally in the successive calls of the routine. The calls terminate either when singleton sets ($\mathcal{S}$) are reached or the rightmost column c_{max} is reached.

Collapse(I, c, c_{max},r_{max}, pat)
(1) IF $(c > c_{max})$ THEN rows of I is a cluster and pat is the putative root
(2) ELSE {
(3) $\mathcal{S}_{zero} \leftarrow \{i \mid I[i, c]$ is '0'$\}$
(4) FOR each $p \neq$ '0' in column c of I {
(5) $\mathcal{S}_p \leftarrow \{i \mid I[i, c]$ is $p\} \cup \mathcal{S}_{zero}$
(6) Restrict I only to the r $(= |\mathcal{S}_p|)$ rows of $\mathcal{S}_p$ to get $\tilde{I}$
(7) IF $(r > 1)$ THEN Collapse($\tilde{I}$, $c + 1$, c_{max}, r, $pat \oplus p$)
(8) }
(9) }

The structure of the tree (plausible evolutionary history) of a cluster with the putative root is embedded in the run-time history of the wrapper procedure and the recursive calls of Collapse(). We omit the details here to avoid digression. It suffices to assume that this tree can be constructed with ease. Note that it is possible to get overlapping clusters using Collapse(), due to the use of $\mathcal{S}_{zero}$ in the procedure: see line (5) above. When a sample, say i, appears in multiple clusters, the result is to be interpreted as follows: for the granularity used, the analysis is unchanged whether i is in one cluster or the other(s). The strength of this approach is in the flexibility it provides in terms of alternative plausible hypotheses for a data set.

Fig 10 (3) shows a clustering of the samples of I''. Fig 11 describes cluster 2 from the example and shows its possible evolutionary history as a tree. Note that the four rows 22, 23, 24, 25 of I'' are identical except in column 12 where rows 22 and 23 have a value 1 and rows 24 and 25 have a value 5. Thus these two groups are split in the tree shown in Fig 11. The vector representing the putative root of this cluster has a value of 1 in column 12. This is because the value 5 does not appear in any of the remaining rows (other than rows 22, 23, 24, 25) in I' at column 12; however value 1 does appear in some of the others. Were it the case, that value 1 did not appear in any of the remaining rows, then the vector would have had a value 0 in this column.

(Step 2) Computing Segmentation *S*

In this step we segment the matrix I into as small a number of segments as possible, such that for each segment there exists a *compatible forest* (described in Problem 3 at the end of the section). However, we must first introduce some terminology to understand this problem setting and its proposed solution.

Let S be a segmentation with K segments of the input $n \times m$ character matrix I. For each $1 \leq k \leq K$, let the kth segment be $s^k = [j_1^k, j_2^k]$. For convenience, the size of the kth segment is $sz_k = j_2^k - j_1^k + 1$. Then I^k is the $n \times sz_k$ sub-matrix obtained by extracting the columns $j_1^k \leq j \leq j_2^k$ of I.

The segmentation S is such that for each (sub)matrix I^k there exists a compatible network N^k with no recombination nodes and a single segment in its segmentation. In other words, N^k is a forest, i.e., every connected component is a tree.

Character array I *to multisets* C*'s.* Let column j of the matrix I have L^j distinct characters, where none of the characters is a dont care ('-'). Let these characters of column j be $c_1^j, c_2^j, \ldots, c_l^j, \ldots, c_{L^j}^j$. For each distinct character c_l^j of column j, define a set of rows (or sample numbers) as follows:

$$C_l^j = \{i \mid I[i, j] \text{ is } c_l^j\}.$$

Thus each column j can be written as a multiset (or, set of sets):

$$\mathcal{C}^j = \{C_1^j, C_2^j, \ldots, C_l^j, \ldots C_{L^j}^j\}.$$

Let C_0^j is the set of 0's or wild cards in column j and is called the *zeroset* of column j.

Two sets C_1 and C_2 *straddle* if both the conditions hold: (1) the intersection of C_1 and C_2, $C_1 \cap C_2$, is not empty, and, (2) both the set differences $C_1 \setminus (C_1 \cap C_2)$ and $C_2 \setminus (C_1 \cap C_2)$ are non empty. As an example, let $C_1 = \{1, 2, 3\}$, $C_2 = \{2, 3, 4\}$, $C_3 = \{3, 4\}$ and $C_4 = \{5, 6\}$. Then C_4 does not straddle with any of the other sets and C_3 does not straddle with C_2. However, C_1 and C_2 straddle; C_1 and C_3 straddle.

Two nodes v_1 and v_2 are *incomparable* if there exists no segment label s such that there is a s-path from a root to a leaf node with both v_1 and v_2 in the path.

Fact 3 *(incomparable mutation nodes) Let v_1 and v_2 be two mutation nodes with incoming edges e_1 and e_2 respectively with solid character labels c_1 and c_2 at column j. Then v_1 and v_2 must be incomparable in the compatible network.*

Problem 3 *(Compatible Forest Problem) Given a matrix I, the task is to find if there exists some partitioning of the zeroset C_0^j of each j such that any pair of columns j and k are compatible using the partitions of the zerosets for the given A.*

It is possible to ignore the wild characters by replacing each by a non-zero (unique) integer. Further, when the character matrix I is binary and the wild characters are ignored, this problem has been called *perfect phylogeny* in literature. Even this restricted problem is known to be NP-hard.

We use a greedy algorithm to segment the matrix I into as small a number of segments as possible, such that for each segment there exists a compatible forest. Fig **??** shows a example of I and three possible segmentations using $A = 1$, i.e., at most one character per column is designated as a dont care. The first one has the minimum number of segmentations, i.e, 1. Nevertheless, we study two more segmentations for illustrative purposes. For partitioning of zerosets, consider the multisets for columns 5, 6 and 7 of I of Fig **??** (1):

$\mathcal{C}^5$	$\mathcal{C}^6$	$\mathcal{C}^7$
$C_0^5 = \{1\}$, $C_1^5 = \{3, 6\}$, $C_2^5 = \{2, 4, 5\}$.	$C_0^6 = \{1, 2, 4, 6\}$, $C_1^6 = \{3, 5\}$.	$C_1^7 = \{1, 2, 3, 6\}$, $C_2^7 = \{4, 5\}$.

Pattern '`1`' of column 5 and pattern '`1`' of column 7 are designated dont care characters ('−') in the respective columns in Fig **??** (3). Next only C_1^6 is augmented with elements of the zeroset C_0^6 to obtain $\{1, 2, 3, 4, 5, 6\}$.

$\mathcal{C}^5$	$\mathcal{C}^6$	$\mathcal{C}^7$
$C_0^5 = \{1\}$, $C_2^5 = \{2, 4, 5\}$.	$C_1^{6'} = \{1, 2, 3, 4, 5, 6\}$.	$C_2^7 = \{4, 5\}$.

Now, it can be easily verified that each pair of columns (5, 6, 7) is compatible and a compatible tree is shown in Fig **??** (3b).

(Step 3) Forests to Networks: The DSR Algorithm

Segmentation suggests a method to compute a compatible network: Given I, a possible segmentation and the corresponding forests are first constructed and then the compatible network is constructed from the forests. Thus we set the stage to solve the following problem.

Problem 4 *(Consensus Compatible Network Problem) Given two networks N_1 on a vertex set U and N_2 on a vertex set V, defined on the same set of samples, and with no common mutation edge labels, the task is to compute N_3 on some vertex set W, with two segments s_1 and s_2 such that for each edge label c_1 in N_1 and each edge label c_2 in N_2, the following holds:*

$$\begin{aligned} L(N_3, s_1, c_1) &= L(N_1, s_1, c_1), \\ L(N_3, s_2, c_2) &= L(N_2, s_2, c_2). \end{aligned}$$

Overview of the approach. We solve this problem using a topology based method. Our approach is iterative, bottom-up working at one level of N_1 and N_2 at a time. The method gets its name from the need to give one of three possible "colors" (Dominant or Subdominant or Recombinant) assignment to nodes at each stage, which is central to this approach. Roughly speaking, a *dominant* node in W uses the edge labels of N_1 and N_2; a *subdominant* uses one of the edge labels of N_1 and N_2 (but not both); and a *recombinant* uses neither of the edge labels of N_1 and N_2 and is indeed a recombinant node in N_3. In the iterative procedure, the "color" of a dominant or a subdominant node may change to recombinant. For the DSR algorithm, the label of the leafnode is the set of samples or rows represented by that node. We begin by considering the bottommost level in both N_1 and N_2 and computing the intersection matrix X. Let P_u be the leaf nodes in N_1 and P_v be lefnodes in N_2. For convenience, the P's of iteration i are written as P_u^i and P_v^i. P_u^i and P_v^i are updated through the iterations as follows:

$$\begin{aligned} P_u^i &= \{u \in U \mid \text{for all descendants } x \text{ of } u \text{ in } N_1,\, x \in P_u^{i'} \text{ holds for some } i' < i\}, \\ P_v^i &= \{v \in V \mid \text{for all descendants } x \text{ of } v \text{ in } N_2,\, x \in P_v^{i'} \text{ holds for some } i' < i\}. \end{aligned}$$

Compatible DSR assignment of X. An appropriate assignment of DSR is required so that a network N_3 can actually be built. The DSR assignment of matrix X is compatible if it satisfies the following two conditions:

1. Each row and each column in matrix X has at most one dominant. If there is no dominant, then it has at most one subdominant.

2. A non-recombinant element can have another non-recombinant in its row or its column but not both.

It can be verified that if any one of the above conditions is violated, it would be impossible to construct a compatible network N_3. Although in the discussion above N_1 and N_2 are forests, the same method can be extended to the case when one or both are not forests (see the original papers).

The correctness of the DSR algorithm follows from the two observations which can be verified.

Fact 4 *(Incomparable P's) If $u_1, u_2 \in P_u^i$, at some iteration i, then u_1 and u_2 are incomparable in N_1. Similarly for $v_1, v_2 \in P_v^i$.*

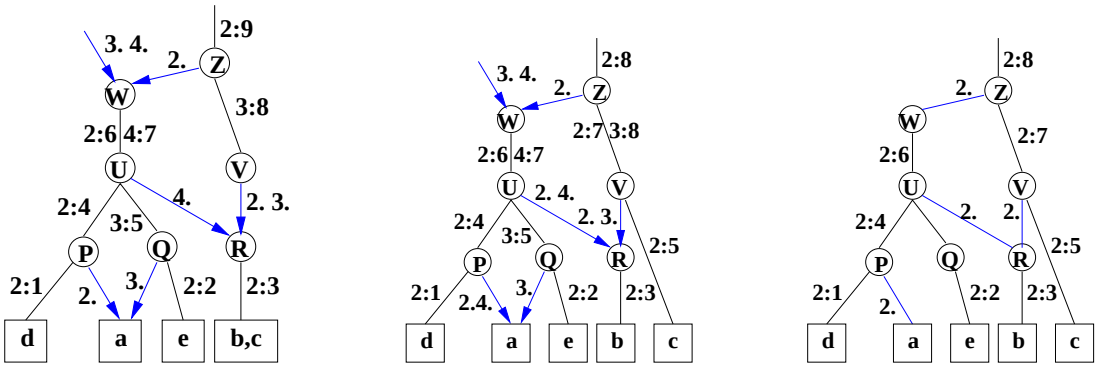
Fact 5 *(Incomparable w's) If two solid characters c_1 and c_2 are edge labels at position j, incident on nodes w_1 and w_2 in N_3, then w_1 and w_2 are incomparable in N_3.*

Approximation Factor of the Greedy DSR Scheme

In this section, we compute the approximation factor of the greedy version of the DSR Scheme. Let the number of new recombination events produced by the DSR algorithm in G_3 be N_{DSR}. Let the optimal number of new recombinations be N_{opt}. We use the following definition of the true approximation factor:

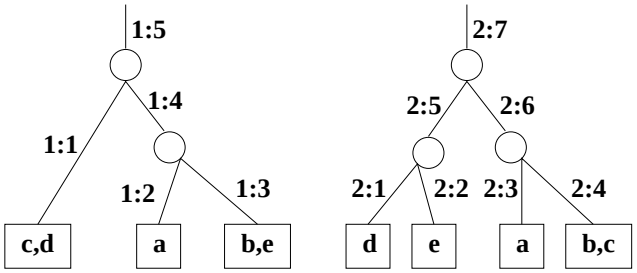
$$approx_{true} = \frac{N_{\text{DSR}} - N_{\text{opt}}}{N_{\text{opt}}}. \tag{2}$$

For given graphs G_1 and G_2 let $z_l = \max(n_l, m_l)$ where $n_l > 0$ and $m_l > 0$ are the number of nodes at level l in G_1 and G_2 respectively. Further, let Z be the sum of all z_l over all the levels (excluding the leaf level). Let $L_v(G)$ be all the leafnodes (extant units) reachable from node v in G. For each level, $l > 0$, i.e. excluding the leafnodes, consider $L_{v_i}(G_1)$, $1 \leq i \leq n_l$, where each v_i is at level l in G_1. Similarly consider $L_{u_i}(G_2)$, $1 \leq i \leq m_l$, where each u_i is at level l in G_2. Let x_l be the number of non-empty intersections between the two collection of sets and let Y be the sum of x_l over all the levels (excluding leaf level). Note that if G_1 and G_2 are the same (isomorphic) graphs then $Y = Z$ and $N_{\text{opt}} = 0$.



(a) Network (G_1, S) (b) Incompatible Network (G_2, S) (c) $Restricted(G_2, 2)$

Figure 12: In (a) & (b) G_1 and G_2 have segmentation $S = \{2, 3, 4\}$. (b) The two parents of node 'R' have labels $\{4, 2\}$ and $\{3, 2\}$. Thus, the network restricted to segment label 2, shown in (c), has a closed path defined by the nodes labeled 'Z', 'W', 'U', 'R' and 'V'. Hence the network in (b) is not compatible.



(a) Tree T_1 (b) Tree T_2

	$2:1$	$2:2$	$2:3$	$2:4$	$-\phi-$
$1:1$	$\{d\}$	$\emptyset$	$\emptyset$	$\{c\}$	$\emptyset$
$1:2$	$\emptyset$	$\emptyset$	$\{a\}$	$\emptyset$	$\emptyset$
$1:3$	$\emptyset$	$\{e\}$	$\emptyset$	$\{b\}$	$\emptyset$
$-\phi-$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	

(1) X_1

	$2:5$	$2:6$	$-\phi-$
$1:4$	$\{\cdot\}$	$\{\cdot\}$	?
$-\phi-$	?	?	

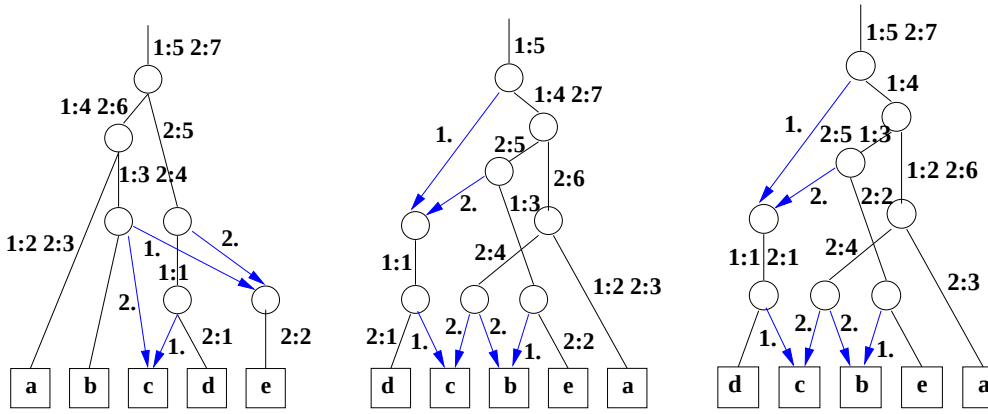
(2) X_2

	$2:7$	$-\phi-$
$1:5$	$\{\cdot\}$	?
$-\phi-$	?	

(3) X_3

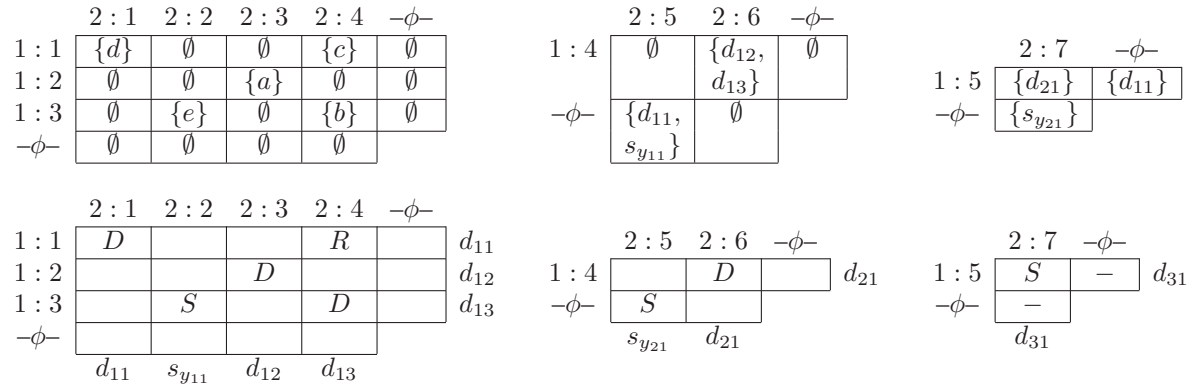
(c) Sequence of X-matrices of T_1 and T_2

Figure 13: Given trees T_1 in (a) and T_2 in (b), each of height 3. (c) These two trees define X_l, $1 \leq l \leq 3$, for each level l.Note that the entries in X_l, $l > 1$ differ in details depending on the choices the DSR algorithm makes. While '$\emptyset$' denotes an empty set, '?' (including '$\{\cdot\}$') could be either empty or non-empty, again depending on the choices the DSR Scheme makes.



(a) Network G_1 (b) Network G_2 (c) $G_2' \equiv G_2$

Figure 14: (a) & (b) Two possible consensus networks G_1 and G_2 for two input trees T_1 and T_2 of Fig 13. (c) The edge labels of G_2 have been locally shuffled keeping the exact same topology.



(a) Level 1: X_1 (b) Level 2: X_2 (c) Level 3: X_3

Figure 15: X-matrices of Network G_1 of Fig 14 (a). The X_l matrix is shown on the top and the DSR assignment shown in the bottom row for each l, $1 \leq l \leq 3$.

Theorem 1 .
$$approx_{true} \leq \frac{Z}{\max(1, Y - Z)}. \tag{3}$$

Proof: Let $N_{\max}$ ($N_{\min}$) be the maximum (minimum) number of new recombinations produced by the DSR scheme over all possible DSR assignments. Then we first show the following:

$$N_{\min} \leq N_{\text{opt}} \leq N_{\text{DSR}} \leq N_{\max}. \tag{4}$$

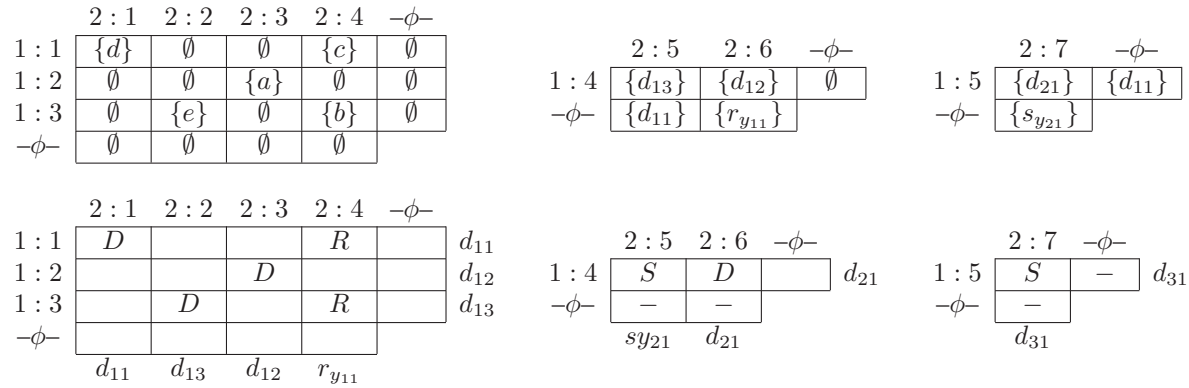
Clearly $N_{\text{opt}} \leq N_{\max}$ holds (else it contradicts the optimality of N_{opt}). Next we have to show that $N_{\min} \leq N_{\text{opt}}$ holds as well. For this we need a few more characterizations of the network.

Recombination Node Descriptor $F_1|F_2$: Let Y be the set all given haplotypes (or taxa). A *split* or *bipartition* is written as $Z_1|Z_2$ where Z_1 and Z_2 are nonoverlapping subsets of Y with $Y = Z_1 \cup Z_2$. A *tripartition* $Z_1|Z_2|Z_3$ is defined similarly. In earlier works a mutation event has been associated with a bipartition of Y and a recombination event with a tripartition. However, the latter requires certain restrictions in the form of network G, i.e., a recombination node cannot be a direct descendent of another recombination node. Here we define recombination nodes as a bipartition of an appropriate subset of features.

For a fixed segment s, let s-path be a path in the graph with mutation edge(s) and recombinant edge(s) with s in its label. For any v, note that there is a unique s-path from a root to v. Further, let v be a recombination node and lbl_1 and lbl_2 be the labels of the two incoming (recombination) edges u_1v and u_2v respectively. For $s_1 \in lbl_1$ but $s_1 \notin lbl_2$, let feature f_1 be such that $s_1 : f_1$ is in the label of the closest mutation edge on the s_1-path from v. Then F_1 is the set of all such features. F_2 is defined similarly. For example consider in G_1 of Fig 12(a), consider the recombination leafnode labeled with haplotype a. Here $lbl_1 = \{2\}$, $lbl_2 = \{3\}$ and the descriptor for this node is $F_1|F_2 = \{2{:}4\}|\{3{:}5\}$. For the recombination node labeled 'R', $lbl_1 = \{4\}$, $lbl_2 = \{2, 3\}$ and the descriptor is $F_1|F_2 = \{4{:}7\} \mid \{2{:}9, 3{:}8\}$.

Isomorphism ($G_1 \equiv G_2$): Let $L_v(G)$ be all the leafnodes (extant units) reachable from node v. Let $s{:}f$ be in the label of the unique incoming edge on mutation node v and then let $L_{s:f}(G)$ be the same as L_v. Two compatible networks G_1 and G_2 on the same segmentation S are *isomorphic* (or identical), written as $G_1 \equiv G_2$, if the following two conditions hold: (1) For each element $s{:}f$ in G_1, $L_{s:f}(G_1) = L_{s:f}(G_2)$ and viceversa, and, (2) For each recombination node v in G_1 with descriptor $F_1|F_2$, there exists a recombination node in G_2 with the same descriptor and viceversa.

Canonical Form: It is possible to bubble *up* or *down* an element in the mutation edge label to obtain G' such that $G' \equiv G$. Our convention will be to bubble *down* the element of the mutation edge label, towards a leafnode. A network G is in the *canonical form* (1) if no node has only one outgoing edge and (2) if no element of any mutation edge label can be bubbled down to obtain G' with $G' \equiv G$. For example see Fig 14.



(a) Level 1: X_1 (b) Level 2: X_2 (c) Level 3: X_3

Figure 16: X-matrices of Network G_2 of Fig 14 (b). Also see Fig 15 for a description of the matrices.

Since the levels of nodes in a canonical network are unique, the following can be readily verified (see also concrete examples in Figs 13 and 19).

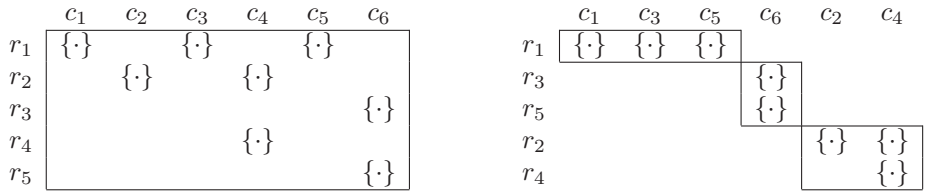
Lemma 1 *Let G_3 be the consensus of G_1 and G_2 which are in canonical forms, with $l_{\max}$ ($l_{\min}$) as the maximum (minimum) of the heights of G_1 and G_2. Then there exist some X-matrices, X_1, X_2, ..., $X_{l_{\max}}$ whose DSR assignments produce G_3. This is written as $G_3 \cong X_1, X_2, \ldots, X_{l_{\max}}$.*

Back to the proof: We have to show that $N_{\min} \leq N_{\text{opt}}$ holds. Assume the contrary, i.e., $N_{\text{opt}} < N_{\min}$. In other words, the optimal number of new recombinations is even lower than the minimum produced by the algorithm over all possible choices. Then consider this network G'_3 with N_{opt} new recombinations. Then by Lemma 1, there exist a sequence of X-matrices $G'_3 \cong X_1, X_2, \ldots, X_{l_{\max}}$ with some DSR assignments for each X_l. Thus by these choices of the algorithm $N_{\min} \leq N_{\text{opt}}$ must hold, again leading to a contradiction. Hence $N_{\text{opt}} \not< N_{\min}$. Here ends the proof of correctness of Eqn 4. Next, we give a few characterizations of the DSR assignment to facilitate the counting of the new recombinations.

Type I & II (new) Recombination Events: Let v be a recombination node in G_3 with labels lbl_1 and lbl_2 on the two incoming edges and descriptor $F_1|F_2$. The recombination event is *new* if, without loss of generality, $lbl_1 \subseteq S_1$ and $lbl_2 \subseteq S_2$. In other words, this recombination node is a result of the consensus of G_1 and G_2 (and not a recombination that existed in G_1 or G_2). A new recombination node v is of two types: Let e_1 (e_2) be a mutation edge in G_1 (G_2) with a label in F_1 (F_2). Without loss of generality, let $level(e_1, G_1) = l$. Then the recombination is of Type I at level l if $level(e_2, G_2) = l$ and is of Type II at level l if $level(e_2, G_2) > l$. Further, let the number of (non-empty) entries assigned dominant be n_l^D, subdominant be n_l^S and recombinant be n_l^R in an X-matrix X_l. Then the following can be verified.

Lemma 2 *The number of Type I recombination events at level l in G_3 is n_l^R. The number of Type II recombination events at level l in G_3 is $\leq n_l^D + n_l^S$. Also, the number of recombination events in a network is bounded below ($N_{\min}$) by the number of Type I recombination events and above ($N_{\max}$) by the sum of the number of Type I and Type II recombination events.*

Islands in *X*: We now give tighter bounds on n_l^D, n_l^S and n_l^R for our analysis. Consider a bipartite graph $B(V, E)$ with V partitioned into (1) n_l nodes, corresponding to the rows and (2) m_l nodes corresponding to the columns of X_l. The adjacency matrix X'_l is obtained from X_l where an empty set entry is replaced with 0 and a non-empty set entry with 1. Let the number of connected components of graph $B(V, E)$ be C_l. Each connected component corresponds to an *island* in X_l which is a collection of rows and columns of X_l. Thus X_l is fragmented into C_l islands, $X_{l,i}$, written as: $X_l = X_{l,1} + X_{l,2} + \ldots + X_{l,C_l}$. See Fig 17 for an example. Note that this fragmentation is for analysis purposes only. Further, $\sum_{l=1}^{l_{bnd}} \sum_{i=1}^{C_l} y_{l,i}$, for any $y_{l,i}$,



(a) X_l (b) The three islands: $X_l = X_{l,1} + X_{l,2} + X_{l,3}$

Figure 17: (a) X_l has five rows and six columns. (b) The rows and columns have been permuted (shuffled) to reveal the three islands (or three connected components in the associated bipartite graph).

will be written simply as $\sum_{l,i}^{l_{bnd}} y_{l,i}$. Let island $X_{l,i}$ have $x_{l,i}$ non-empty entries and let the number of entries assigned $Y(D$ or S or $R)$ in $X_{l,i}$ be $n_{l,i}^Y$. Within an island the number of non-recombinants cannot exceed $\max(n_{l,i}, m_{l,i})$ by Rules 1 and 2. Thus the following is easily verified:

Lemma 3 *For each l and i (i.e., island $X_{l,i}$),*

$$\begin{aligned} n_{l,i}^D + n_{l,i}^S &= \max(n_{l,i}, m_{l,i}) && \textit{(by Rule 3 in island } X_{l,i}\textit{)}, \\ n_{l,i}^R &= x_{l,i} - \max(n_{l,i}, m_{l,i}) && \textit{(since } x_{l,i} = n_{l,i}^D + n_{l,i}^S + n_{l,i}^R\textit{)}. \end{aligned}$$

Back to the proof: Next, let $N_{c\max}(\geq N_{\max})$ and $N_{c\min}(\leq N_{\min})$ be some computable functions of the input.Using Lemmas 2 and 3, we define appropriate (computable) $N_{c\max}$ and $N_{c\min}$ as follows:

$$N_{\max} \leq \sum_{l}^{l_{\min}} x_l = N_{c\max} \tag{5}$$

$$N_{\min} = \sum_{l,i}^{l_{\min}} n_{l,i}^R = \sum_{l,i}^{l_{\min}} (x_{l,i} - \max(n_{l,i}, m_{l,i})) \geq \sum_{l}^{l_{\min}} x_l - \sum_{l}^{l_{\min}} \max(n_l, m_l) = N_{c\min} \tag{6}$$

Note that the greedy Rule 3 encourages fragmentation of X_l, $l > 1$, into islands and under the best case scenario we get $n_l^D + n_l^S = \sum_l^{l_{\min}} \max(n_l, m_l)$, which is used in Eqn 6 above. Finally, using Eqn 2, we have

$$approx_{true} = \frac{N_{\text{DSR}} - N_{\text{opt}}}{N_{\text{opt}}} \leq \frac{N_{c\max} - N_{c\min}}{N_{c\min}} \approx \frac{N_{c\max} - N_{c\min}}{\max(1, N_{c\min})} \tag{7}$$

The correctness of Eqn 3 is established by setting $Z = \sum_{l,i}^{l_{\min}} \max(n_l, m_l)$ and $Y = \sum_l^{l_{\min}} x_l$. Here ends the proof. □

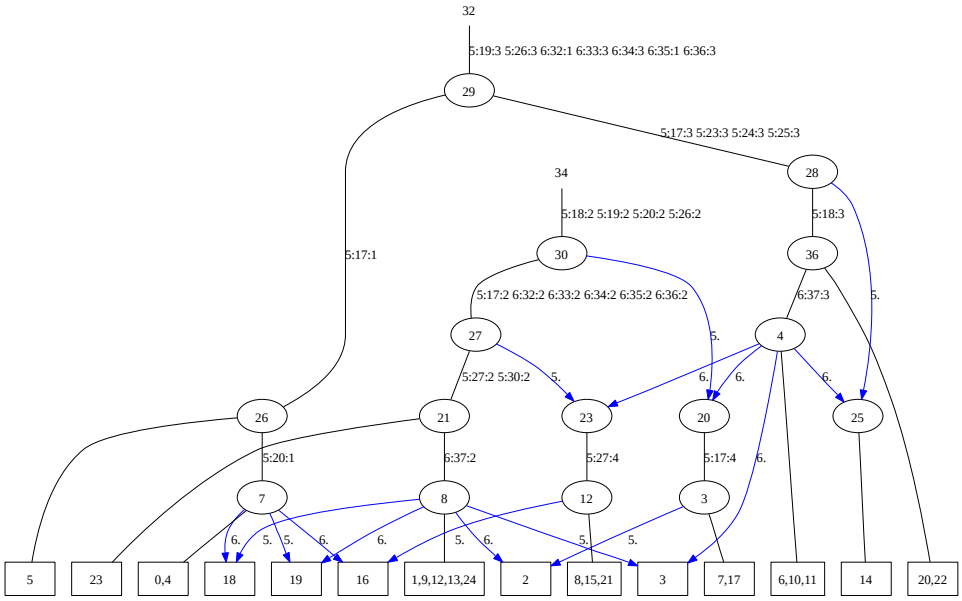
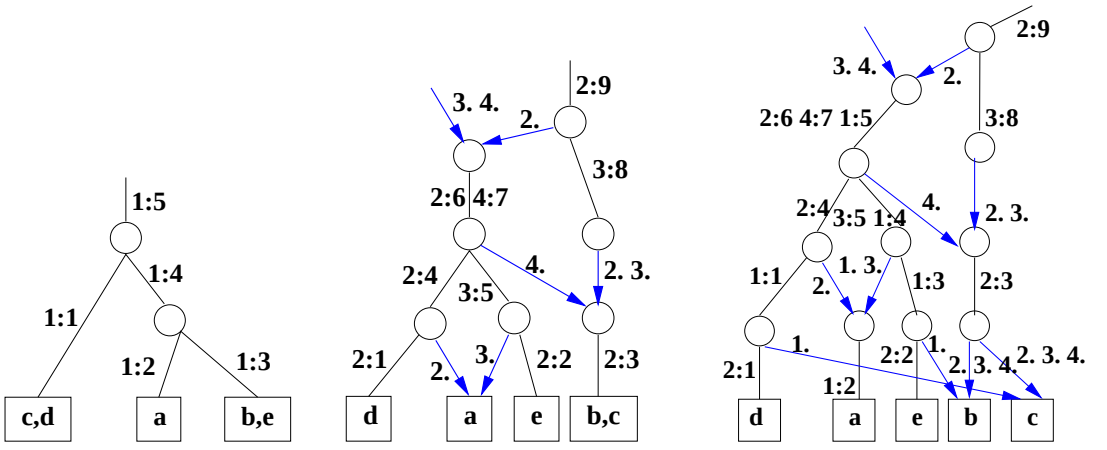


Figure 18: The network on segment Chr X: 87390235-87412114 of the three populations using HapMap II data. The leafnodes are labeled with (a set of) clusters of the input haplotypes. A label on an internal node is for reference purposes only. An element of the edge label is to be interpreted as segment-id:position-id:pattern-id. Further details are available at `http://www.cs.nyu.edu/parida/res/public/Xchr08`.



(a) Tree T_1 (b) Network G_2 (c) G_3: A consensus of T_1 and G_2

Figure 19: Consensus of a tree and a network.

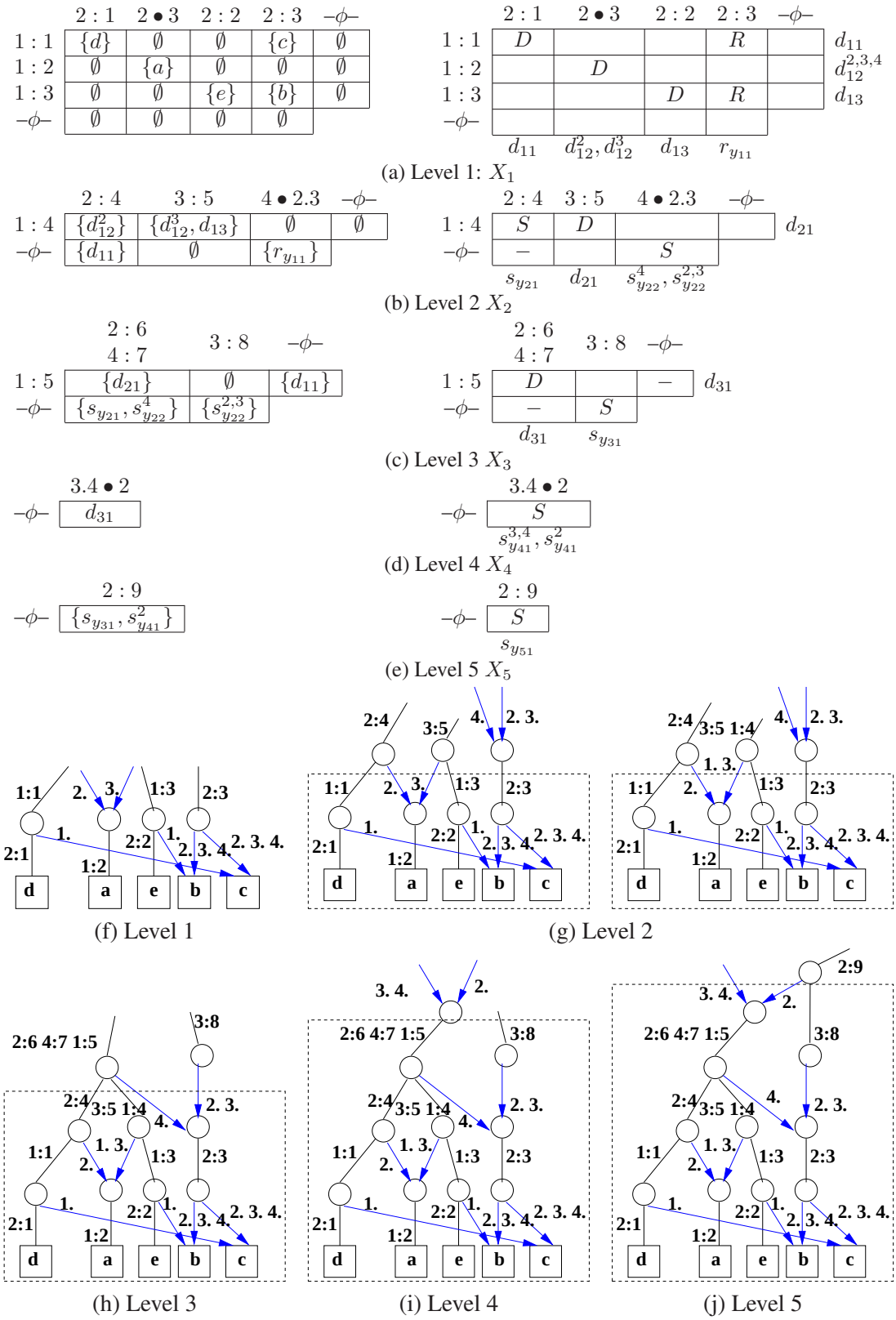


Figure 20: Stepwise construction of G_3 of Fig 19 (c) as consensus of T_1 and G_2: (a)-(e) The X matrices and the DSR assignments. (f)-(j) The construction of G_3 using the DSR assignments of (a)-(e).

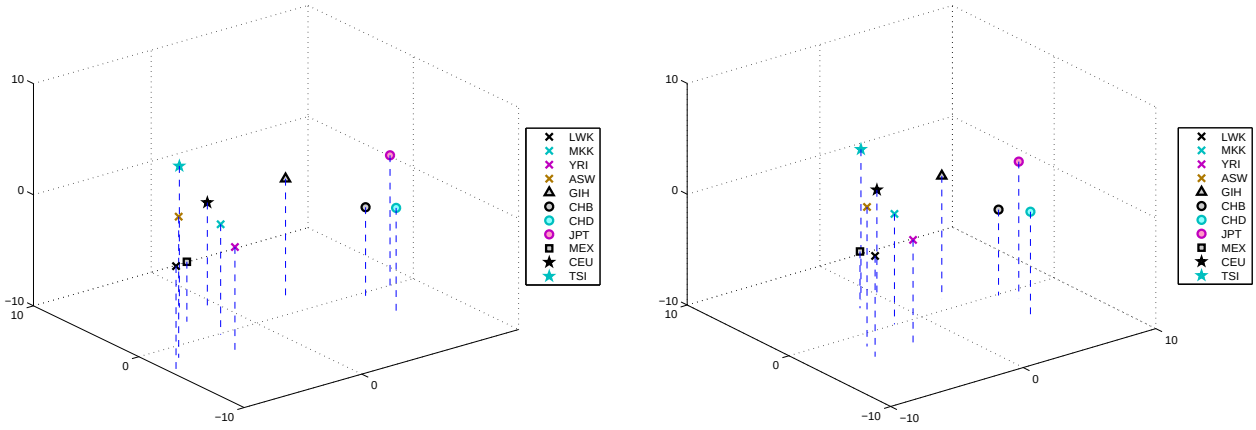


Figure 21: MDS plots of all regions and the best thirteen: stress factor 4% and 4.64% respectively.

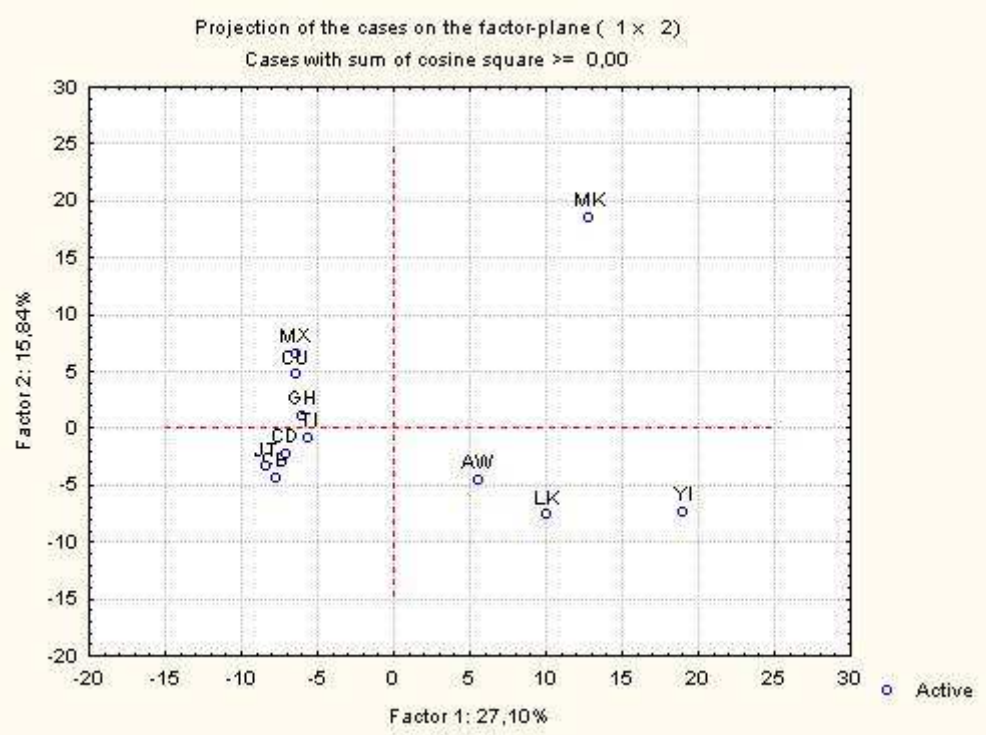


Figure 22: Region 1 Principal Component Analysis

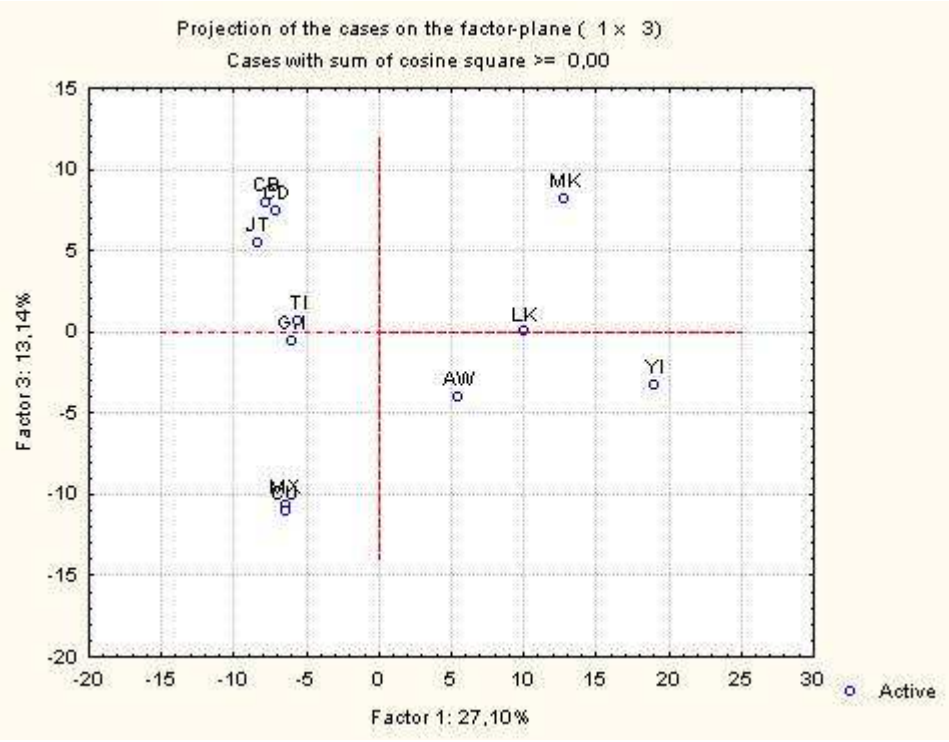


Figure 23: Region 1 Principal Component Analysis

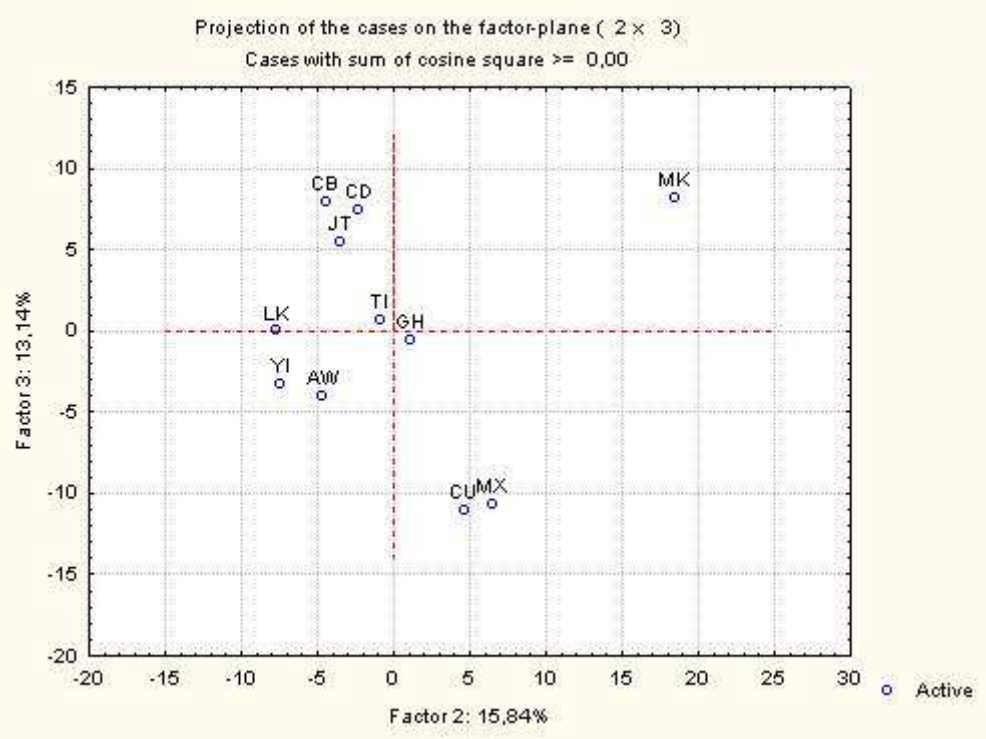


Figure 24: Region 1 Principal Component Analysis

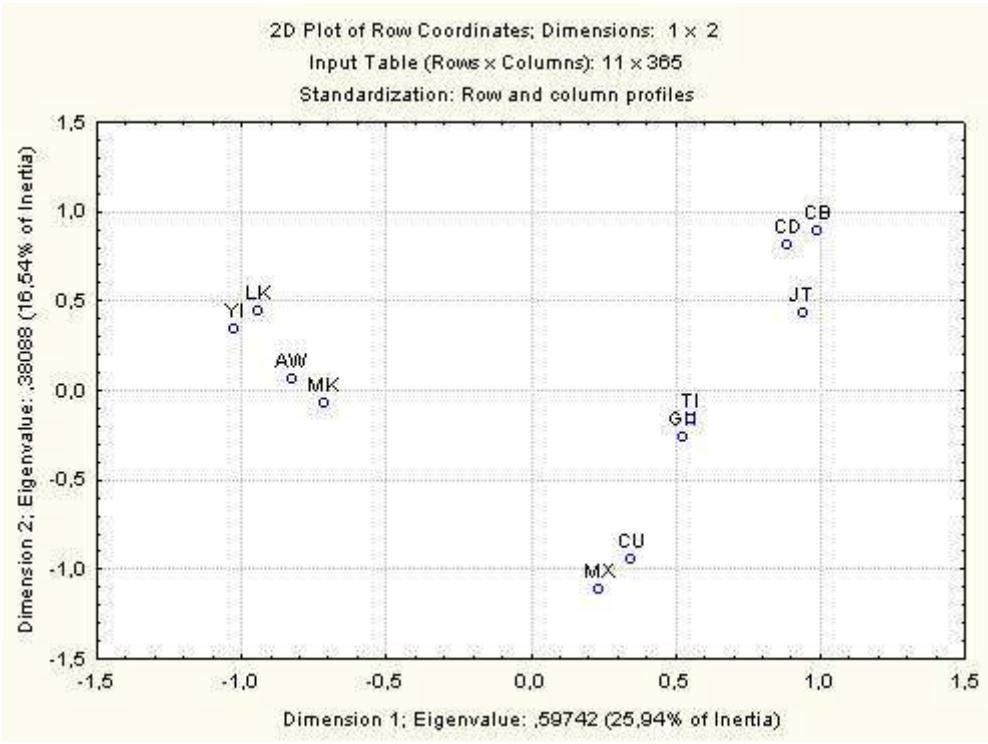


Figure 25: Region 1 Correspondence Analysis

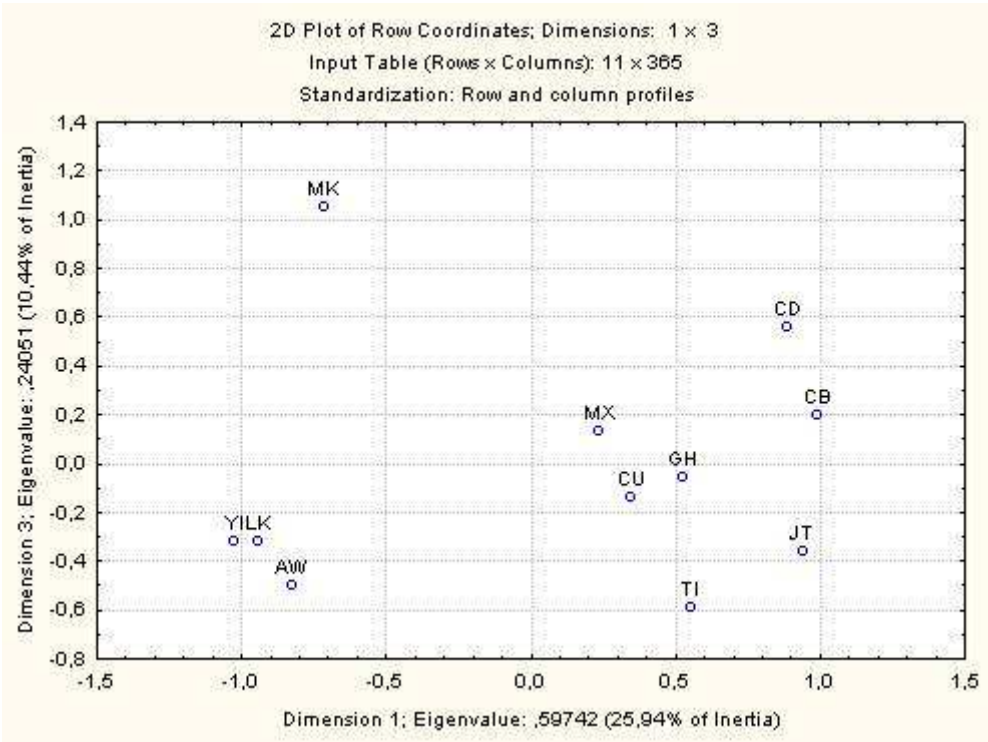


Figure 26: Region 1 Correspondence Analysis

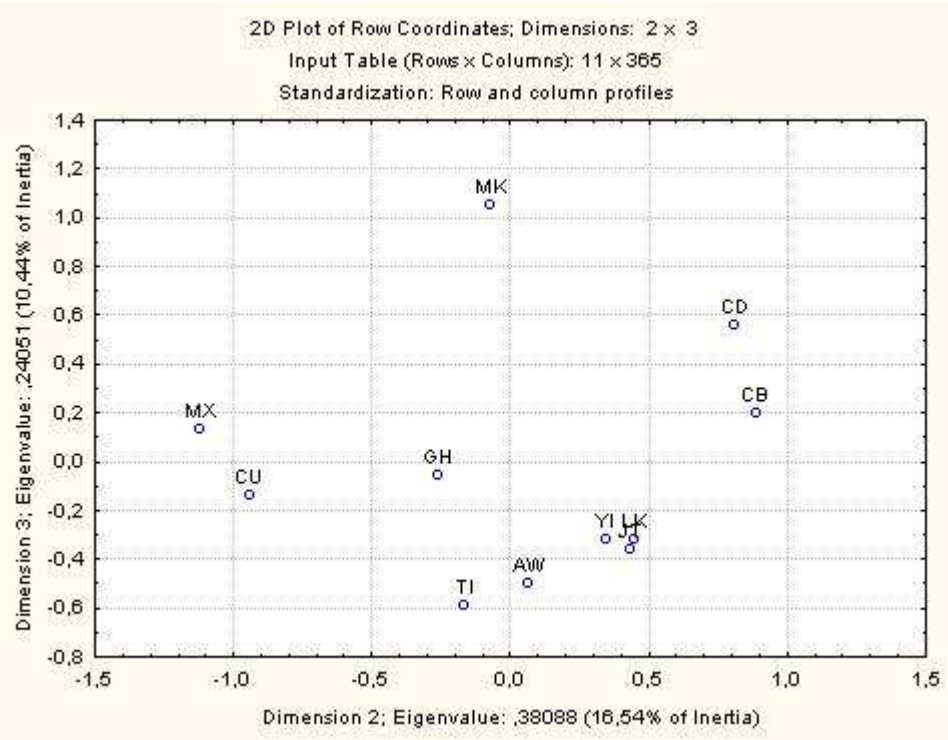


Figure 27: Region 1 Correspondence Analysis

	Fact.1	Fact.2	Fact.3	Fact.4	Fact.5	Fact.6	Fact.7	Fact.8	Fact.9	Fact.10
LWK	*10.23*	*10.16*	0.00	*31.29*	*25.93*	*11.29*	1.35	0.54	0.11	0.02
MKK	*16.74*	**59.07**	*13.75*	0.54	0.08	0.08	0.27	0.07	0.28	0.04
YRI	**36.73**	9.45	2.19	*27.41*	8.47	3.49	1.17	1.66	0.30	0.05
ASW	3.09	3.72	3.38	5.72	1.16	**60.03**	0.25	*12.22*	1.28	0.05
GIH	3.61	0.19	0.07	**32.67**	**44.36**	3.34	5.75	0.27	0.49	0.15
CHB	6.07	3.37	*13.13*	0.02	1.60	1.90	9.24	1.70	9.32	**44.55**
CHD	5.08	0.88	*11.64*	0.08	0.04	1.21	*15.19*	*10.55*	*24.70*	*21.53*
JPT	7.01	2.04	6.19	0.04	8.78	0.03	*11.57*	7.98	*24.06*	*23.20*
MEX	4.20	7.24	*23.98*	0.81	0.57	0.00	**32.47**	*19.18*	1.68	0.78
CEU	4.14	3.76	**25.58**	0.23	5.75	*15.03*	11.23	**24.75**	0.08	0.37
TSI	3.10	0.13	0.10	1.19	3.26	3.60	11.49	*21.07*	**37.70**	9.26

	Fact.1	Fact.2	Fact.3	Fact.4	Fact.5	Fact.6	Fact.7	Fact.8	Fact.9	Fact.10
LWK	3.29	0.00	0.85	0.27	**36.84**	4.92	**41.93**	0.34	2.08	0.39
MKK	*17.07*	**63.60**	4.76	0.62	2.67	0.00	1.81	0.02	0.08	0.29
YRI	**47.62**	*19.09*	2.43	*14.44*	1.95	2.60	1.74	0.01	0.21	0.81
ASW	2.26	9.54	0.09	**73.98**	1.78	2.06	0.19	0.19	0.00	0.81
GIH	3.93	0.01	1.67	1.41	2.67	**64.12**	0.34	*16.04*	0.27	0.45
CHB	5.59	0.76	2.37	3.78	*32.11*	7.90	*15.08*	*19.30*	2.62	1.40
CHD	4.26	0.46	8.00	0.84	0.92	2.90	3.80	2.53	**37.04**	*30.17*
JPT	5.61	0.91	8.23	1.90	1.43	5.06	0.04	**51.31**	5.32	*11.09*
MEX	2.87	0.96	1.85	2.02	*17.28*	9.58	*35.05*	8.60	2.00	*10.70*
CEU	3.89	3.11	**56.96**	0.69	0.83	0.10	0.02	0.57	*15.24*	9.50
TSI	3.61	1.56	*12.78*	0.06	1.52	0.77	0.00	1.09	*35.13*	**34.39**

Figure 28: Regions 1 and 2: Principal Component Analysis

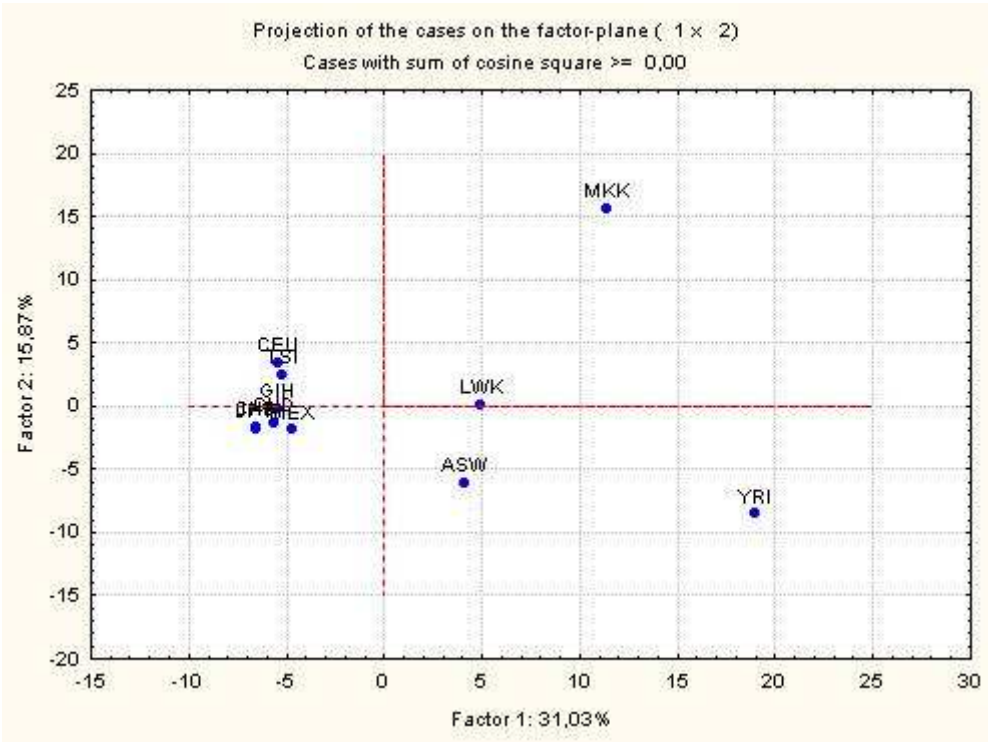


Figure 29: Region 2 Principal Component Analysis

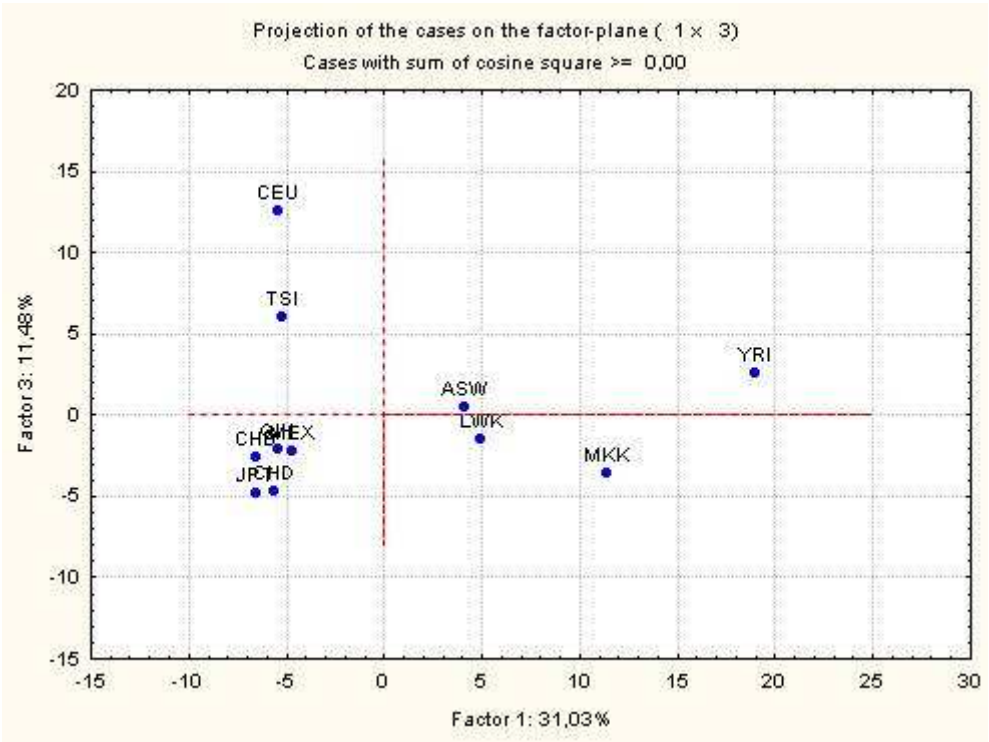


Figure 30: Region 2 Principal Component Analysis

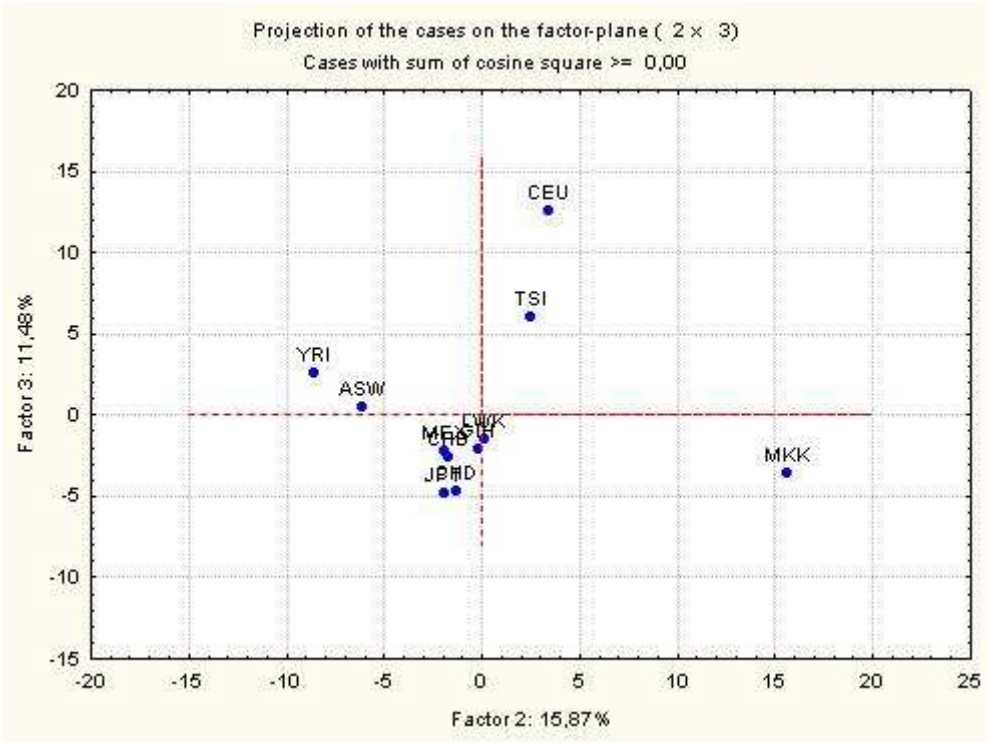


Figure 31: Region 2 Principal Component Analysis

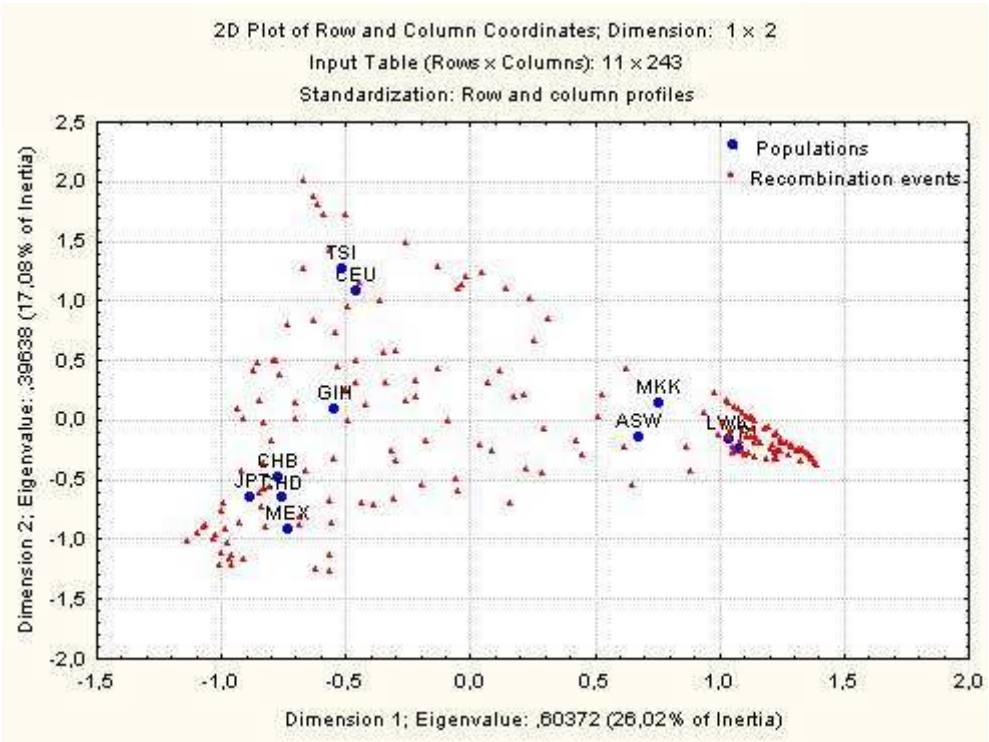


Figure 32: Region 2 Correspondence Analysis

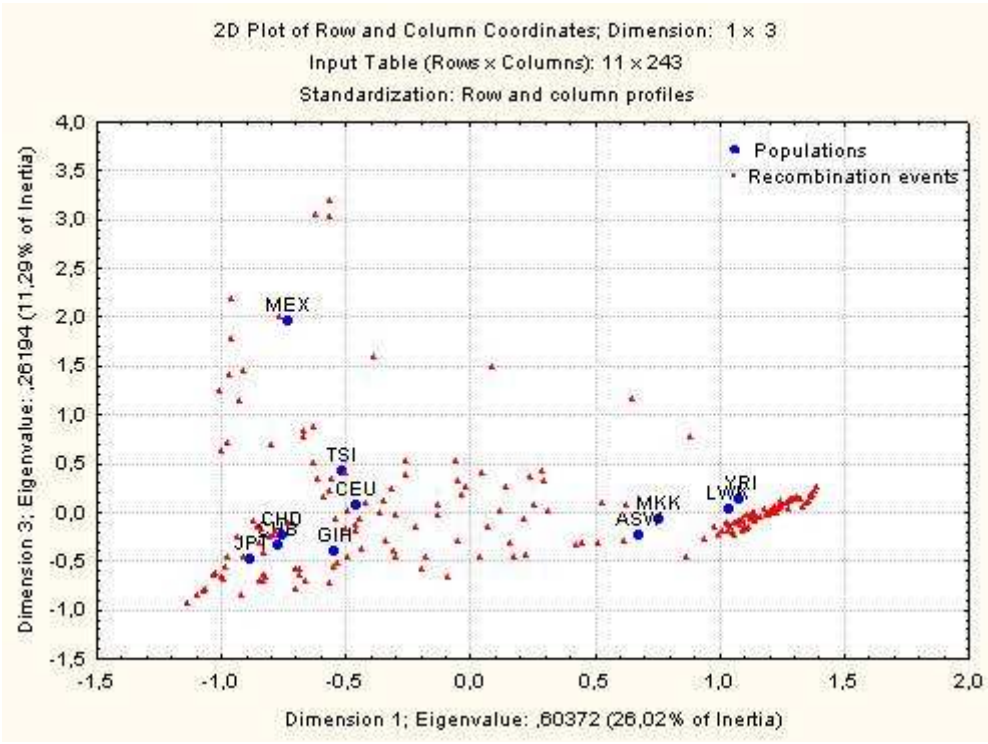


Figure 33: Region 2 Correspondence Analysis

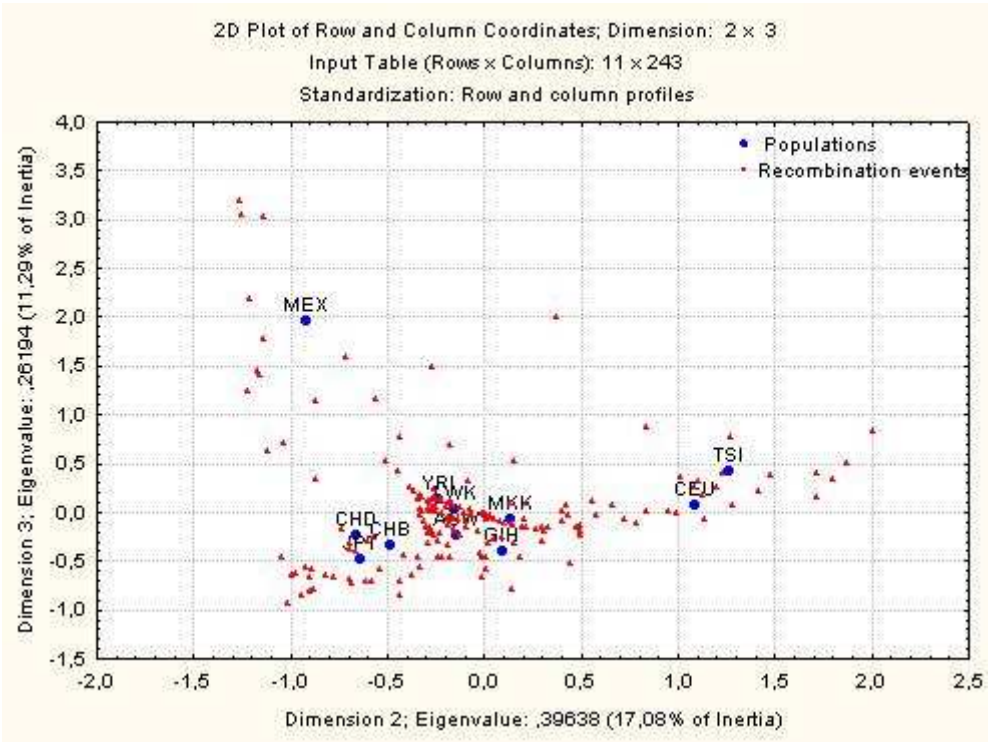


Figure 34: Region 2 Correspondence Analysis