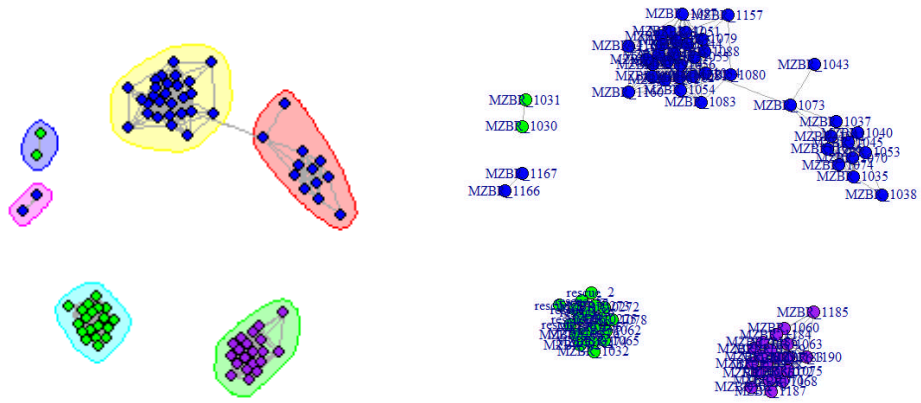
NetView v1.0 Results

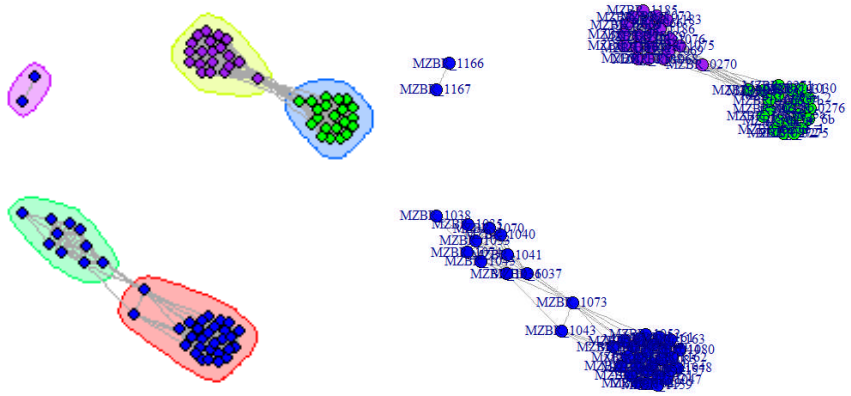
NetView Summary

When we investigated population genetic distances in NetView we could see more detailed substructure within our three key genetic clusters (Borneo, Java and Sumatra/Singapore), which was congruent with our STRUCTURE results from K = 2 to 7 (Figure 3 in the manuscript). We tested a large range of parameters in NetView using little k = 1 to 100 and we obtained a range of valid cluster results for three different models, including Walktrap, Infomap and Fast-greedy.

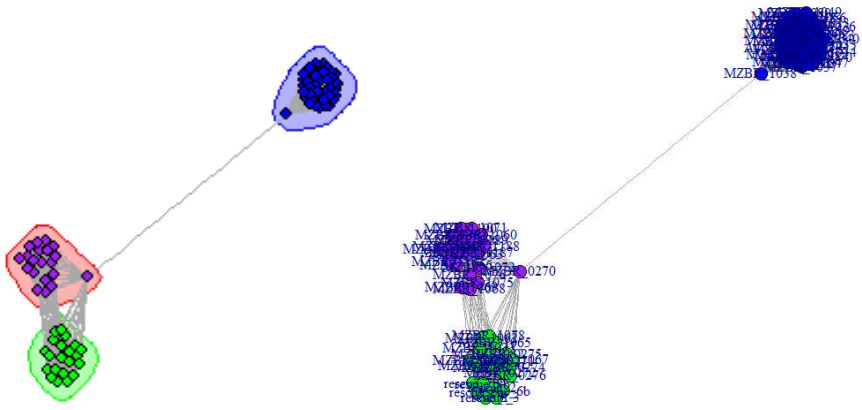
Three examples from the lowest, middle and highest numbers of possible clusters from the Walktrap model are provided below as examples. They are all valid cluster arrangements, unlike STRUCTURE, NetView does not tell you which is the most likely of the arrangements.



(a) Three clusters of Bornean samples (blue dots), one Javan cluster (purple dots) and two Sumatra/Singapore clusters (green dots). However, we know from our fineRADstructure and PCA results that MZBR1030 and 1031 are likely related so that might be why they form a distinct cluster here. The same applies to MZBR 1167 and 1166.



(b) Three clusters of Bornean samples (blue dots), one cluster for Java (purple dots) and one cluster for Sumatra/Singapore (green dots). MZBR 1166 and 1167 are likely to be related.



(c) Our three key clusters only, Borneo (blue dots), Java (purple dots) and Sumatra/Singapore (green dots).

PyRAD Scripts

Param File

```
==** parameter inputs for pyRAD version 3.0.64  **====================== affected step ==
./                  ## 1. Working directory                          (all)
                    ## 2. Loc. of non-demultiplexed files (if not line 18)  (s1)
                    ## 3. Loc. of barcode file (if not line 18)          (s1)
vsearch             ## 4. command (or path) to call vsearch (or usearch)    (s3,s6)
muscle              ## 5. command (or path) to call muscle                (s3,s7)
CG,AATT             ## 6. Restriction overhang (e.g., C|TGCAG -> TGCAG)     (s1,s2)
16                  ## 7. N processors (parallel)                       (all)
5                   ## 8. Mindepth: min coverage for a cluster           (s4,s5)
6                   ## 9. NQual: max # sites with qual < 20 (or see line 20)(s2)
.95                 ## 10. Wclust: clustering threshold as a decimal       (s3,s6)
ddrad               ## 11. Datatype: rad,gbs,pairgbs,pairddrad,(others:see docs)(all)
79                  ## 12. MinCov: min samples in a final locus           (s7)
3                   ## 13. MaxSH: max inds with shared hetero site         (s7)
pangolin5           ## 14. Prefix name for final output (no spaces)       (s7)
==== optional params below this line ==================================  affected step ==
                ## 15.opt.: select subset (prefix* only selector)          (s2-s7)
Hong_Kong_2_R1_     ## 16.opt.: add-on (outgroup) taxa (list or prefix*)       (s6,s7)
                ## 17.opt.: exclude taxa (list or prefix*)               (s7)
/Volumes/GABE/Gabriel/Pangolin/demultiplexed/*.fq.gz    ## 18.opt.: loc. of de-multiplexed data (s2)
                ## 19.opt.: maxM: N mismatches in barcodes (def= 1)        (s1)
                ## 20.opt.: phred Qscore offset (def= 33)                (s2)
                ## 21.opt.: filter: def=0=NQual 1=NQual+adapters. 2=strict   (s2)
                ## 22.opt.: a priori E,H (def= 0.001,0.01, if not estimated) (s5)
                ## 23.opt.: maxN: max Ns in a cons seq (def=5)            (s5)
```

24.opt.: maxH: max heterozyg. sites in cons seq (def=5) (s5)

25.opt.: ploidy: max alleles in cons seq (def=2;see docs) (s4,s5)

26.opt.: maxSNPs: (def=100). Paired (def=100,100) (s7)

27.opt.: maxIndels: within-clust,across-clust (def. 3,99) (s3,s7)

28.opt.: random number seed (def. 112233) (s3,s6,s7)

29.opt.: trim overhang left,right on final loci, def(0,0) (s7)

* ## 30.opt.: output formats: p,n,a,s,v,u,t,m,k,g,* (see docs) (s7)

31.opt.: maj. base call at depth>x<mindepth (def.x=mindepth) (s5)

32.opt.: keep trimmed reads (def=0). Enter min length. (s2)

33.opt.: max stack size (int), def= max(500,mean+2*SD) (s3)

34.opt.: minDerep: exclude dereps with <= N copies, def=1 (s3)

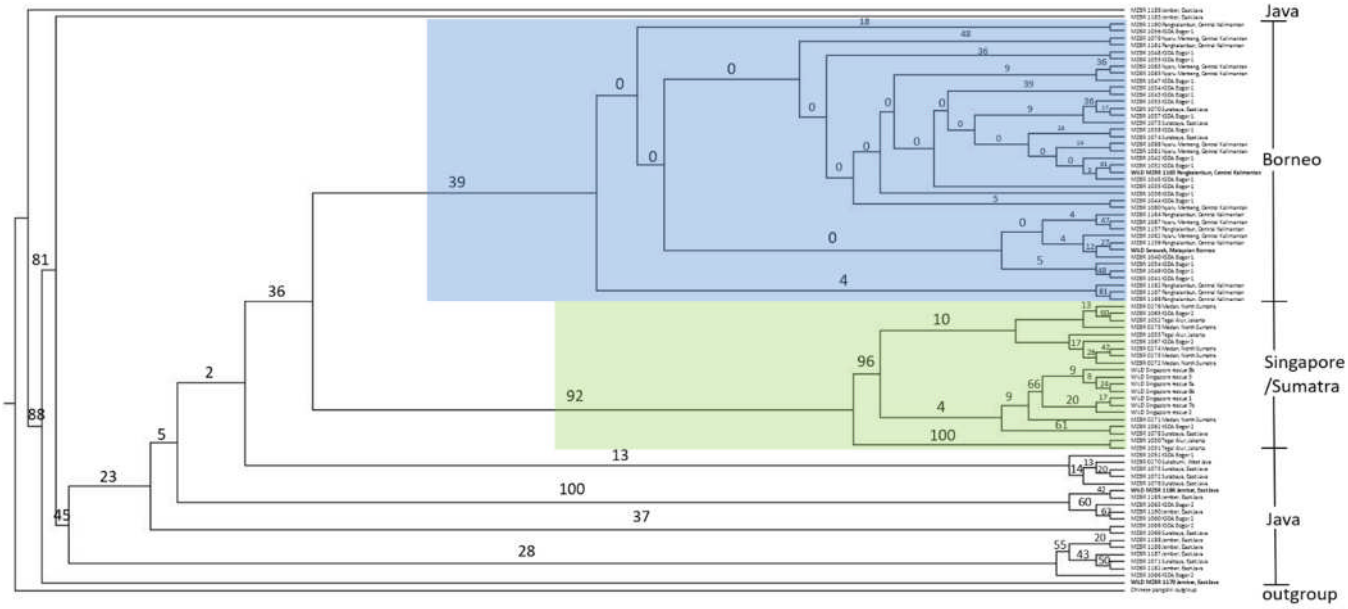
35.opt.: use hierarchical clustering (def.=0, 1=yes) (s6)

36.opt.: repeat masking (def.=1='dust' method, 0=no) (s3,s6)

37.opt.: vsearch max threads per job (def.=6; see docs) (s3,s6)

==== optional: list group/clade assignments below this line (see docs) ==========

RAxML Results



Maximum likelihood phylogeny in RAxML using 2365 SNPs. Key: blue, Borneo; green, Singapore/Sumatra, no colour, Java and Chinese pangolin outgroup. (See Figure 3a in manuscript).

Missing data of 2365 SNPs across 82 Sunda pangolins and one Chinese pangolin

Family ID	Individual ID	Missing phenotype? Y/N	N_MISS	N_GENO	F_MISS
Pop_1	Hong_Kong_2_R1_	Y	2308	5170	0.4464
Pop_1	MZBR_1189_R1_	Y	1893	5170	0.3662
Pop_1	MZBR_1179_R1_	Y	1562	5170	0.3021
Pop_1	MZBR_1043_R1_	Y	1413	5170	0.2733
Pop_1	MZBR_1067_R1_	Y	409	5170	0.07911
Pop_1	rescue_2_R1_	Y	400	5170	0.07737
Pop_1	MZBR_1037_R1_	Y	365	5170	0.0706
Pop_1	Malaysian_R1_	Y	361	5170	0.06983
Pop_1	MZBR_1182_R1_	Y	306	5170	0.05919
Pop_1	MZBR_1184_R1_	Y	277	5170	0.05358
Pop_1	MZBR_1186_R1_	Y	272	5170	0.05261
Pop_1	MZBR_1183_R1_	Y	199	5170	0.03849
Pop_1	MZBR_1052_R1_	Y	190	5170	0.03675
Pop_1	MZBR_1081_R1_	Y	156	5170	0.03017
Pop_1	MZBR_1048_R1_	Y	142	5170	0.02747
Pop_1	rescue_1_R1_	Y	138	5170	0.02669
Pop_1	MZBR_1185_R1_	Y	132	5170	0.02553
Pop_1	MZBR_1166_R1_	Y	130	5170	0.02515
Pop_1	MZBR_1164_R1_	Y	122	5170	0.0236

Pop_1	MZBR_1162_R1_	Y	99	5170	0.01915
Pop_1	MZBR_1075_R1_	Y	90	5170	0.01741
Pop_1	MZBR_1051_R1_	Y	80	5170	0.01547
Pop_1	rescue_6b_R1_	Y	80	5170	0.01547
Pop_1	MZBR_1071_R1_	Y	58	5170	0.01122
Pop_1	MZBR_1068_R1_	Y	46	5170	0.008897
Pop_1	MZBR_1035_R1_	Y	45	5170	0.008704
Pop_1	MZBR_1088_R1_	Y	45	5170	0.008704
Pop_1	MZBR_0271_R1_	Y	44	5170	0.008511
Pop_1	MZBR_1032_R1_	Y	39	5170	0.007544
Pop_1	MZBR_1047_R1_	Y	39	5170	0.007544
Pop_1	MZBR_1031_R1_	Y	38	5170	0.00735
Pop_1	rescue_8b_R1_	Y	35	5170	0.00677
Pop_1	rescue_5a_R1_	Y	34	5170	0.006576
Pop_1	MZBR_1060_R1_	Y	33	5170	0.006383
Pop_1	MZBR_1069_R1_	Y	31	5170	0.005996
Pop_1	MZBR_1033_R1_	Y	30	5170	0.005803
Pop_1	MZBR_1066_R1_	Y	28	5170	0.005416
Pop_1	MZBR_0274_R1_	Y	27	5170	0.005222
Pop_1	MZBR_1045_R1_	Y	27	5170	0.005222
Pop_1	MZBR_1161_R1_	Y	26	5170	0.005029
Pop_1	MZBR_1030_R1_	Y	25	5170	0.004836
Pop_1	MZBR_1062_R1_	Y	25	5170	0.004836
Pop_1	MZBR_1070_R1_	Y	24	5170	0.004642
Pop_1	MZBR_1036_R1_	Y	23	5170	0.004449
Pop_1	MZBR_1063_R1_	Y	23	5170	0.004449
Pop_1	MZBR_1073_R1_	Y	23	5170	0.004449
Pop_1	rescue_3_R1_	Y	22	5170	0.004255
Pop_1	MZBR_1054_R1_	Y	21	5170	0.004062
Pop_1	MZBR_1072_R1_	Y	21	5170	0.004062
Pop_1	MZBR_1160_R1_	Y	21	5170	0.004062
Pop_1	MZBR_1049_R1_	Y	20	5170	0.003868
Pop_1	MZBR_1076_R1_	Y	20	5170	0.003868
Pop_1	MZBR_0276_R1_	Y	19	5170	0.003675
Pop_1	MZBR_1041_R1_	Y	19	5170	0.003675
Pop_1	MZBR_1065_R1_	Y	19	5170	0.003675
Pop_1	rescue_7b_R1_	Y	19	5170	0.003675
Pop_1	MZBR_1034_R1_	Y	18	5170	0.003482
Pop_1	MZBR_1080_R1_	Y	18	5170	0.003482
Pop_1	MZBR_1187_R1_	Y	18	5170	0.003482
Pop_1	MZBR_1157_R1_	Y	17	5170	0.003288
Pop_1	MZBR_1055_R1_	Y	16	5170	0.003095
Pop_1	MZBR_1167_R1_	Y	16	5170	0.003095
Pop_1	MZBR_0273_R1_	Y	15	5170	0.002901

Pop_1	MZBR_1074_R1_	Y	15	5170	0.002901
Pop_1	MZBR_1085_R1_	Y	15	5170	0.002901
Pop_1	MZBR_1188_R1_	Y	15	5170	0.002901
Pop_1	MZBR_0275_R1_	Y	14	5170	0.002708
Pop_1	MZBR_1040_R1_	Y	14	5170	0.002708
Pop_1	MZBR_1053_R1_	Y	14	5170	0.002708
Pop_1	MZBR_1079_R1_	Y	14	5170	0.002708
Pop_1	MZBR_1087_R1_	Y	14	5170	0.002708
Pop_1	MZBR_1159_R1_	Y	13	5170	0.002515
Pop_1	MZBR_1082_R1_	Y	13	5170	0.002515
Pop_1	MZBR_0272_R1_	Y	12	5170	0.002321
Pop_1	MZBR_1038_R1_	Y	12	5170	0.002321
Pop_1	MZBR_1190_R1_	Y	12	5170	0.002321
Pop_1	MZBR_1163_R1_	Y	12	5170	0.002321
Pop_1	MZBR_1078_R1_	Y	11	5170	0.002128
Pop_1	MZBR_1083_R1_	Y	11	5170	0.002128
Pop_1	MZBR_0270_R1_	Y	10	5170	0.001934
Pop_1	MZBR_1042_R1_	Y	9	5170	0.001741
Pop_1	MZBR_1044_R1_	Y	9	5170	0.001741
Pop_1	MZBR_1056_R1_	Y	9	5170	0.001741

Barcode labels, concentrations, & reads per sample in Illumina sequencing

We used one Illumina HiSeq 2500 Rapid Sequencing Run, which included two lanes. In total we obtained 49.67 GB of 150 bp x 150 bp paired-end reads from 96 samples across the two lanes, read ones = 24.12 GB and read twos = 25.55 GB. Overall only 7 samples had to be discarded from further analysis due to low coverage, MZBR.1158, MZBR.1064, MZBR.1046, MZBR.1084, MZBR.1086, MZBR.1077 and MZBR.1061

Pool 1 (Lane 1)

Number	AATI conc ng/ul	Peak average bp	Code	PCR index	P1 tag	Barcode	Index1and5	number of Illumina reads retained after demultiplexing
18	0.2782	374	MZBR.1167	1	1	GCATG	ATCACG	3257588
12	0.2431	385	MZBR.1183	1	3	CGATC	ATCACG	1505418
8	0.2099	422	MZBR.1179	1	5	TGCAT	ATCACG	14291542
34	0.4209	415	MZBR.0276	1	6	CAACC	ATCACG	3431735
15	0.2567	384	MZBR.0271	1	7	GGTTG	ATCACG	2242497
13	0.2459	375	MZBR.0275	1	8	AAGGA	ATCACG	2263904
27	0.3744	375	MZBR.1184	1	9	AGCTA	ATCACG	1672559
17	0.2716	384	MZBR.1166	1	12	ACGGT	ATCACG	1282088
37	0.4482	376	MZBR.1187	1	15	ATACG	ATCACG	4443611
24	0.3384	355	MZBR.1069	1	18	CATAT	ATCACG	2441036
32	0.4126	375	MZBR.1078	1	19	CGAAT	ATCACG	3484361
25	0.3539	364	MZBR.1082	1	23	CGTCG	ATCACG	1859968
26	0.3663	387	MZBR.1085	1	26	CTGTC	ATCACG	2467778
1	0.0159	538	MZBR.1158	1	29	GAGAT	ATCACG	9277
30	0.4105	366	MZBR.1162	1	33	GGATA	ATCACG	1531530
16	0.2629	382	MZBR.1163	1	34	GGCCA	ATCACG	2808870
9	0.2155	415	Hong Kong 2	1	42	TATAC	ATCACG	11287530
31	0.4109	419	MZBR.1055	5	5	TGCAT	ACAGTG	5413205
11	0.2415	426	MZBR.1056	5	6	CAACC	ACAGTG	4280139
20	0.2876	425	MZBR.1087	5	7	GGTTG	ACAGTG	4512961
10	0.2173	430	MZBR.1060	5	10	ACACA	ACAGTG	4173103
36	0.4371	414	MZBR.1062	5	12	ACGGT	ACAGTG	3654199
35	0.422	402	MZBR.1064	5	14	ACTTC	ACAGTG	6619
33	0.4176	377	MZBR.1065	5	18	CATAT	ACAGTG	4713201
41	0.6098	418	MZBR.1066	5	19	CGAAT	ACAGTG	4206175
5	0.1946	415	MZBR.1067	5	20	CGGCT	ACAGTG	1563632
40	0.4755	425	MZBR.1068	5	21	CGGTA	ACAGTG	5895334
7	0.2083	425	MZBR.1034	5	22	CGTAC	ACAGTG	5629543
2	0.0532	414	MZBR.1035	5	23	CGTCG	ACAGTG	2749908
38	0.4542	411	MZBR.1036	5	24	CTGAT	ACAGTG	4074762

29	0.4001	363	MZBR.1047	5	25	CTGCG	ACAGTG	2178684
60	0.7577	412	MZBR.1046	5	26	CTGTC	ACAGTG	5022
73	1.1436	421	MZBR.1040	5	28	GACAC	ACAGTG	9337426
23	0.3277	416	MZBR.1041	5	29	GAGAT	ACAGTG	6918632
14	0.2467	421	MZBR.1043	5	31	GCCGT	ACAGTG	12543931
39	0.4548	446	MZBR.1044	5	32	GCTGA	ACAGTG	6062088
19	0.2828	414	MZBR.1045	5	33	GGATA	ACAGTG	4223924
28	0.3878	377	MZBR.1050	5	38	GTCGA	ACAGTG	124630
4	0.1523	376	MZBR.1165	5	41	TAGTA	ACAGTG	221432
21	0.2966	364	MZBR.1180	5	43	TCACG	ACAGTG	574352
3	0.1063	354	MZBR.1181	5	45	TCCGG	ACAGTG	850817
22	0.324	428	MZBR.0272	5	47	TGGAA	ACAGTG	4014965
								Pool 1 Total Reads =158209976

Pool 2 (Lane 2)

Number	AATI conc ng/ul	Peak average bp	Code	PCR index	P1 tag	Barcode	Index1and5	number of Illumina reads retained after demultiplexing
43	0.4864	410	MZBR.1032	1	4	TCGAT	ATCACG	2041608
44	0.4869	412	MZBR.1186	1	14	ACTTC	ATCACG	1468026
45	0.5181	373	MZBR.1080	1	21	CGGTA	ATCACG	2424672
46	0.5348	377	MZBR.1079	1	20	CGGCT	ATCACG	2118608
48	0.5755	375	MZBR.1216	1	13	ACTGG	ATCACG	2786285
49	0.5759	385	rescue 1 10/07/2015	1	36	GTAGT	ATCACG	2448244
50	0.5773	376	MZBR.0274	1	10	ACACA	ATCACG	2259186
52	0.6175	417	rescue 6b 03/12/2015	1	41	TAGTA	ATCACG	3831031
53	0.6568	292	MZBR.1084	1	25	CTGCG	ATCACG	18889
54	0.6675	411	MZBR.1182	1	2	AACCA	ATCACG	1083732
55	0.6718	385	MZBR.1164	1	35	GGCTC	ATCACG	1279889
56	0.6822	373	MZBR.1083	1	24	CTGAT	ATCACG	2674188
57	0.7071	380	MZBR.1081	1	22	CGTAC	ATCACG	1256122
61	0.765	384	rescue 2 26/07/2015	1	37	GTCCG	ATCACG	2248031
62	0.7824	416	MZBR.1033	1	11	AATTA	ATCACG	2929871
63	0.792	382	rescue 3 21/09/2015	1	38	GTCGA	ATCACG	5339750
69	0.9533	381	rescue 5a 02/12/2015	1	40	TACGT	ATCACG	4727700
70	0.9701	387	rescue 7b	1	43	TCACG	ATCACG	9786034

72	1.1394	389	MZBR.1086	1	27	CTTGG	ATCACG	11439
84	1.7331	398	MZBR.1161	1	32	GCTGA	ATCACG	1885981
88	2.067	413	MZBR.1188	1	17	ATTAC	ATCACG	8346659
90	2.6928	426	MZBR.1159	1	30	GAGTC	ATCACG	3317875
92	3.88	387	MZBR.1160	1	31	GCCGT	ATCACG	4492727
95	4.647	423	rescue 8b	1	44	TCAGT	ATCACG	5994813
96	8.6168	387	MZBR.1157	1	28	GACAC	ATCACG	3530997
6	0.1962	414	MZBR.1077	5	26	CTGTC	ACAGTG	172267
42	0.4635	356	MZBR.1075	5	24	CTGAT	ACAGTG	2013952
47	0.5575	421	MZBR.1185	5	48	TTACC	ACAGTG	3114111
51	0.5861	425	MZBR.1061	5	11	AATTA	ACAGTG	328178
58	0.7203	377	MZBR.1038	5	15	ATACG	ACAGTG	4837479
59	0.7327	416	MZBR.1030	5	46	TCTGC	ACAGTG	4177097
64	0.8232	387	MZBR.1052	5	40	TACGT	ACAGTG	1484730
65	0.8441	377	MZBR.1076	5	25	CTGCG	ACAGTG	11143094
66	0.8546	414	MZBR.1054	5	34	GGCCA	ACAGTG	3728779
67	0.8974	411	MZBR.1063	5	13	ACTGG	ACAGTG	5069076
68	0.9249	418	MZBR.1177	5	8	AAGGA	ACAGTG	3874730
71	1.0637	414	MZBR.1053	5	35	GGCTC	ACAGTG	6599854
74	1.2173	335	MZBR.1071	5	20	CGGCT	ACAGTG	6514804
75	1.2178	337	MZBR.1072	5	21	CGGTA	ACAGTG	6664228
76	1.2443	413	MZBR.1031	5	4	TCGAT	ACAGTG	10605820
77	1.2612	415	MZBR.1074	5	23	CGTCG	ACAGTG	5789616
78	1.39	377	MZBR.1048	5	36	GTAGT	ACAGTG	1730982
79	1.438	414	MZBR.1042	5	30	GAGTC	ACAGTG	6999067
80	1.5252	398	MZBR.0270	5	42	TATAC	ACAGTG	5150309
81	1.5427	423	MZBR.1178	5	9	AGCTA	ACAGTG	705925
82	1.6597	424	MZBR.1037	5	17	ATTAC	ACAGTG	2327449
83	1.6964	415	MZBR.0273	5	44	TCAGT	ACAGTG	9610045
85	1.7538	386	MZBR.1051	5	39	TACCG	ACAGTG	3534853
86	1.7783	371	MZBR.1073	5	22	CGTAC	ACAGTG	5086488
87	1.9534	383	MZBR.1049	5	37	GTCCG	ACAGTG	2627054
89	2.2464	422	MZBR.1088	5	2	AACCA	ACAGTG	4784330
91	3.1713	416	MZBR.1190	5	3	CGATC	ACAGTG	7576140
93	4.3819	422	MZBR.1070	5	19	CGAAT	ACAGTG	4491169
94	4.5089	408	MZBR.1189	5	1	GCATG	ACAGTG	12069327
								Pool 2 Total Reads =221113310

One additional sample from Malaysian Borneo, likely from Sarawak, was sequenced later in a shared run with other researchers, read count for that sample was 2131545.

Mitochondrial DNA Primers and Method

Cyt*b* primer:
Cyt*b* Pangolin F: GCCGAGATGTAAACTACGGA; Cyt*b* pangolin R : GTCCGATTAGGA TGAAGGGG

CO1 primer:
CO1 Pangolin F : TGGAAACTGACTAGTGCCCC; CO1 Pangolin R : GCTCCCATGGAGAGAACGTA

5ul of 100uM primer plus 45ul H_20 makes the primer working solution of 0.01nM/ul (make a separate working solution for each primer)

Step 1: Amplification (First batch was for 7 samples only)

Reagent	Reaction Mix (25ul)	Mastermix (x7.5 The extra 0.5ul is just in case) –make this twice: once for *CO1* & once for *Cytb*
Molecular grade water	16.9	126.75
10xbuffer (part of Thermoscientific Taq Polymerase package)	2.5	18.75
$MgCl_2$	2.0	15
dNTPs	0.5	3.75
Forward Primer (use 0.01nM/ul working solution)	0.5	3.75
Reverse Primer (use 0.01nM/ul working solution)	0.5	3.75
Taq Polymerase 1.25U	0.125	0.9375
DNA (ok with 1 to 100ng of DNA, if over 100ng get inhibition)	2	

Add mastermix to strip wells first, 23ul to all, then DNA after.

We used a Kryatec Thermocycler for PCR.

PCR programme for *CO1*: 95°C, 3 mins pre-heat activation.

35 cycles (denaturation 95°C 20secs, annealing 56°C 20secs, extension 72°C 1 min)

Post-cycle extension 72°C 10 mins.

Hold 4°C

PCR programme for *Cytb*: 95°C, 5 mins pre-heat activation.

40 cycles (denaturation 95°C 30secs, annealing 56°C 30secs, extension 72°C 1 min)

Post-cycle extension 72°C 5 mins.

Hold 4°C

To check that amplification had worked we ran gels of the PCR products for 45 mins at 110 volts. The agar gel comprised of 1g agar in 50ml TAE buffer (50xTAE buffer), plus 3.5ul of Gel Red per gel. The wells included 5ul of PCR product plus 1ul of Thermoscientific blue loading dye. The DNA ladder included 2ul of Thermoscientific 1Kb Generuler.

Step 2: Exosap & Clean up

1. Withdraw 5ul of exosap and mix with 45ul H_2O to make the working exosap aliquot. Store at -20°C after use.
2. Clean up: Add 13.5ul of final PCR product + 1.5ul of working exosap aliquot. (9ul of PCR product also works but only use 1ul of working exosap if using 9ul of PCR product)
3. We ran the following exosap program on a Kryatec PCR/Thermocycler machine:
   37°C 1hr 30 mins
   80°C 15 mins
   Hold 4°C
   Then store in 4°C fridge.

Step 3: Big Dye Prep (First batch was for 5 samples only)

Reagent	10ul	Mastermix (5.5 The extra 0.5ul is just in case) Cover in aluminium foil & put on ice due to Big Dye
Molecular grade water	3.92	21.56
Big Dye	2	11
5 x Big Dye buffer	3	16.5
Primer	0.08	0.44
Exosap/DNA product	1	

Do this four times for each primer, i.e., *CO1* forward, *CO1* reverse, *Cytb* forward, *Cytb* reverse.

Each well gets 9ul of Mastermix plus 1ul of DNA

We ran the following SEQ program on a Kryatec Thermocycler machine:

96°C 1 min
96°C 10 secs
50°C 5 secs
60°C 4 mins
} 24 cycles

Hold 4°C
Then store in 4°C fridge.

Step 4: Pure Seq beads for Big Dye Terminator Removal (removes excess Big Dye)

1. 10ul of Pureseq Beads + 41ul 85% ethanol to each well needed on a 96 well plate. Pipette up and down 7 times in the well.
2. Add 10ul of primer/DNA mix per well, from the previous Big Dye Prep stage.
3. Put on a magnetic stand to form DNA/primer beads, wait a few mins.

4. Withdraw the excess ethanol so only the beads (with DNA attached) remain.
5. Add 95ul of 85% ethanol for a first wash of the beads. Wait for the beads to form again then withdraw the ethanol.
6. Repeat the wash again using 95ul of 85% ethanol.
7. Air dry the beads for 5 mins.
8. Make sure no liquid remains and then add 40ul of molecular grade water. Store for up to 24hrs at 4°C, or freeze at -20°C for up to 3 months. (we usually took for Sanger sequencing immediately)

Step 5: Sanger Sequencing

We used the 3130x1 Genetic Analyzer by Life Technologies, with Foundation Data Collection Version 3.0.

ABBA BABA Results

First Run: Our first ABBA BABA test used a wild sample from Borneo, MZBR 1163, as group A; an anomalous sample, MZBR 1040, which falls within the Bornean cluster of SNPs, but has suspected introgression from Java based on the mtDNA result, as group B; and a wild sample from Java, MZBR 1184, as group C; the outgroup was our Chinese pangolin. This tested MZBR 1040 for suspected introgression from the Javan lineage.

Sample1	Sample2	Sample3	Sample4	d	SD	Z	ABBA	BABA		p-val
WILD BORNEO		WILD JAVA								
MZBR_1163_R1	MZBR_1040_R1	MZBR_1184_R1	Hong_Kong_2_R1	**-0.4**	**0.01042**	-38.3877	3	7	0	0.34375*

*p-value was not significant for this run.

Second Run: Our second test used a wild sample from Java, MZBR 1184, as group A; and an anomalous sample, MZBR 0270, which falls within the Javan cluster of SNPs, but has suspected introgression from Singapore/Sumatra based on the mtDNA result, as group B; group C was a wild sample from Singapore/Sumatra, rescue 1; the outgroup was our Chinese pangolin. This tested MZBR 0270 for suspected introgression from the Singapore/Sumatra lineage.

Sample1	Sample2	Sample3	Sample4	d	SD	Z	ABBA	BABA		p-val
WILD JAVA		WILD SUM/SIN								
MZBR_1184_R1	MZBR_0270_R1	rescue_1_R1	Hong_Kong_2_R1	**0.913025**	**0.00925**	98.70536	8820	401	0	4.941e-324

Third Run: Our third test used a wild sample from Java, MZBR 1184, as group A; and an anomalous sample, MZBR 0270, which falls within the Javan cluster of SNPs, but has suspected introgression from Borneo based on the mtDNA result, as group B; group C was a wild sample from Borneo, MZBR 1163; the outgroup was our Chinese pangolin. This tested MZBR 0270 for suspected introgression from the Bornean lineage.

Sample1	Sample2	Sample3	Sample4	d	SD	Z	ABBA	BABA		p-val
WILD JAVA		WILD BORNEO								
MZBR_1184_R1	MZBR_0270_R1	MZBR_1163_R1	Hong_Kong_2_R1	**0.971005**	**0.00972**	99.89767	9381	138	0	4.941e-324

If d is positive, then sample 2 is closer to sample 3 than sample 1 is to sample 3.

If d is negative, then sample 1 is closer to sample 3 than sample 2 is to sample 3.

FineRADstructure (Malinsky et al. 2016)

Below is the Python script which we used to convert our haplotypes.tsv file (from populations analysis in ref_map.pl) into a fineRADstructure input format. The script was provided by Emiliano Trucchi and Milan Malinksy (http://cichlid.gurdon.cam.ac.uk/fineRADstructure.html). We tired a few different runs and finally used max_snps = 1, max_missing_data = 50. The missing data report, and RStudio visualization scripts are also provided below, and a larger version of Figure 2d from the manuscript.

```
#!/usr/bin/env python

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.decomposition import PCA as sklearnPCA

import argparse as ap

#parse arguments

parser = ap.ArgumentParser()

parser.add_argument('-i', '--infile', help='A haplotypes.tsv file as created by populations in the Stacks package (Catchen et al 2013)', required=True, type=str)

parser.add_argument('-n', '--max_snps', help='Maximum number of SNPs per locus', required=True, type=int)

parser.add_argument('-m', '--max_missing_data', help='Maximum %% of missing loci to be included in the PCA', required=True, type=int)

args = parser.parse_args()

#set input and output (not input of fineRADpainter) file

input_file = open(args.infile, "r")
```

```
output_file = args.infile
output_file += '.lociFilt.txt'
output = open(output_file, 'w')

#variables
max_snps = args.max_snps

#set counters
triallelic = 0
invariants = 0
too_many_snps = 0
indels = 0

#set lists
num_alleles_list = []
num_snp_list = []
cat_id_list = []

#filter loci
for line in input_file.readlines():
        #replace missing data value from "-" to ""
        line = line.replace('-','')
        row = line.rstrip().split('\t')
        if row[0] == 'Catalog ID':
                header = row
                samples = header[2:len(header)]
```

```
            output.write(line)
        else:
            #separate genotypes into alleles
            row = [i.split('/') for i in row]
            #drop loci if more than 2 alleles are present (likely already filtered)
            if any(len(i) > 2 for i in row):
                triallelic += 1
            #drop invariant loci
            elif any(i == ["consensus"] for i in row):
                invariants += 1
                num_alleles_list.append(1)
                num_snp_list.append(0)
                cat_id_list.append(row[0])
            else:
                genodata = row[2:len(row)]
                all_alleles = [item for sublist in genodata for item in sublist]
                if any(len(ind) > max_snps for ind in all_alleles): #drop loci if they have more than max_snps allowed
                    too_many_snps += 1
                #drop loci with "N" likely indels but we need to check
                elif any('N' in ind for ind in all_alleles):
                    indels += 1

                else:
                    alleles_set = set([allele for allele in all_alleles if allele != ''])
                    num_alleles_list.append(len(alleles_set))
```

```
                                num_snp_list.append(len(list(alleles_set)[0]))
                                cat_id_list.append(row[0])
                                output.write(line)
output.close()
input_file.close()

print 'Loci filtered sequentially as follows:'
print 'Triallelic =', triallelic, '; Invariants =', invariants, '; More than', max_snps, 'snps =', too_many_snps, '; Indels =', indels
print '---'
print 'Filtered loci for all samples saved to '+args.infile+'.filtered'
print '---'

#plot distribution of number of alleles across loci
plt.ioff()
fig = plt.figure()
ax = fig.add_subplot(111)
counts = np.bincount(num_alleles_list)
plt.bar(range(max(num_alleles_list)+1), counts, width=1, align='center', color='DarkKhaki')
plt.xticks(range(max(num_alleles_list)+1))
plt.xlim(-1, max(num_alleles_list)+1)
plt.ylabel('# LOCI')
plt.xlabel('# ALLELES')
plt.title(str(len(num_alleles_list))+' loci analyzed (including invariants)')
missing_plot_file = args.infile+'.alleles.pdf'
fig.savefig(missing_plot_file,bbox_inches='tight')
```

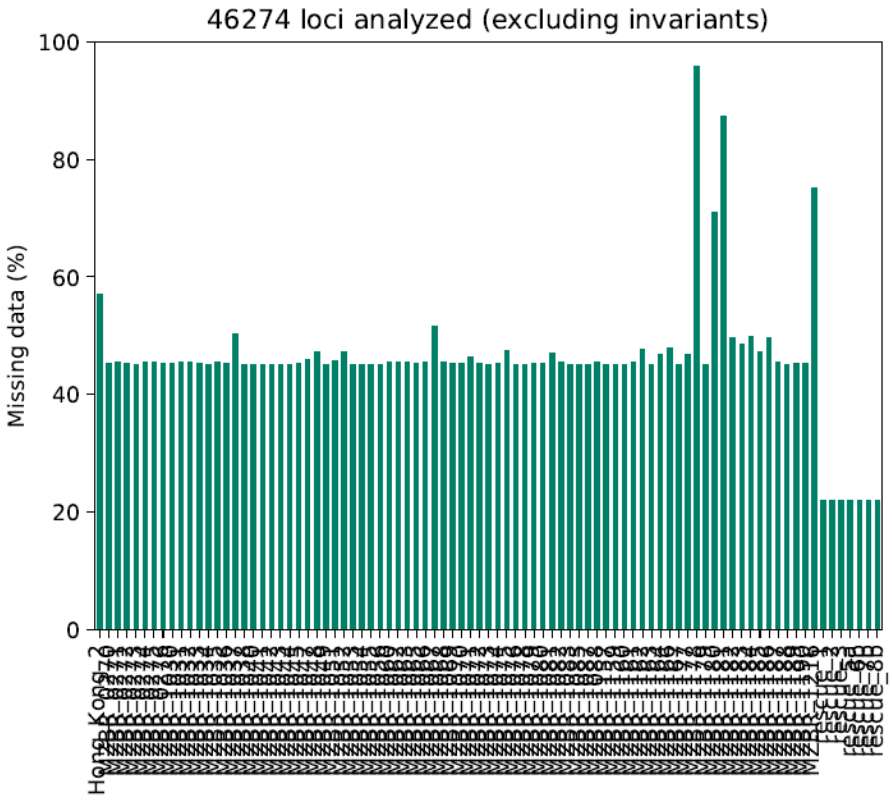
```
plt.close()

print 'Distribution of alleles per locus plotted in '+args.infile+'.alleles.pdf'

print '---'


#plot distribution of number of snps across loci

plt.ioff()

fig = plt.figure()

ax = fig.add_subplot(111)

counts = np.bincount(num_snp_list)

plt.bar(range(max(num_snp_list)+1), counts, width=1, align='center', color='DarkSeaGreen')

plt.xticks(range(max(num_snp_list)+1))

plt.xlim(-1, max(num_snp_list)+1)

plt.ylabel('# LOCI')

plt.xlabel('# SNPS')

plt.title(str(len(num_snp_list))+' loci analyzed (including invariants)')

missing_plot_file = args.infile+'.snps.pdf'

fig.savefig(missing_plot_file,bbox_inches='tight')

plt.close()

print 'Distribution of SNPs per locus plotted in '+args.infile+'.snps.pdf'

print '---'


#Read the output file to plot missing data per sample and PCA of missing data and prepare the fineRADpainter input files

data = pd.read_csv(output_file, sep='\t', header=0, usecols=samples)


fineRADpainter_input = args.infile+'.fineRADpainter.lociFilt.txt'
```

```
data.to_csv(fineRADpainter_input, header=True, sep='\t', index=False)

print 'FineRADpainter input file with filtered loci saved to
'+args.infile+'.fineRADpainter.lociFilt.txt'

print '---'


num_row = data.shape[0]

missing_data = data.isnull().sum().apply(lambda x: round(x*1.0/num_row*100,1))



plt.ioff()

fig = plt.figure()

ax = fig.add_subplot(111)

missing_data.plot(kind='bar', ylim=(0,100), colormap='summer', ax = ax)

plt.ylabel('Missing data (%)')

plt.title(str(num_row)+' loci analyzed (excluding invariants)')

missing_plot_file = args.infile+'.missingdata.pdf'

fig.savefig(missing_plot_file,bbox_inches='tight')

plt.close()

print 'Missing data per sample plotted in '+args.infile+'.missingdata.pdf'

print '---'


fact = 100/args.max_missing_data


thr = data.shape[0]-data.shape[0]/fact

data_20_01 = data.dropna(thresh=thr, axis=1).notnull().astype('int')

sklearn_pca20 = sklearnPCA(n_components=2)
```

```
sklearn_pca20.fit(data_20_01)


filtered_missingX_file =
args.infile+'.fineRADpainter.lociFilt.samples'+str(args.max_missing_data)+'%missFilt.txt'

data.dropna(thresh=thr, axis=1).to_csv(filtered_missingX_file, header=True, sep='\t',
index=False)

print 'FineRADpainter input file with filtered loci and filtered samples with max missing loci
=',str(args.max_missing_data),'% saved to
'+args.infile+'.fineRADpainter.filt'+str(args.max_missing_data)+'%Missing'

print '---'


plt.ioff()

fig = plt.figure()

ax = fig.add_subplot(111)


plt.plot(sklearn_pca20.components_[0], sklearn_pca20.components_[1], 'ro' )


labels = list(data_20_01.columns.values)


for label, x, y in zip(labels, sklearn_pca20.components_[0] ,sklearn_pca20.components_[1] ):

    plt.annotate(label, xy=(x, y), xytext=(-20, 20), textcoords='offset points', ha='right',
va='bottom', arrowprops=dict(arrowstyle = '->', connectionstyle='arc3,rad=0'))


plt.title('PCA of missing data')

missing_plot_file = args.infile+'.missingdataPCA.pdf'

fig.savefig(missing_plot_file,bbox_inches='tight')

plt.close()

print 'PCA of missing data plotted in '+args.infile+'.missingdataPCA.pdf'
```

print '---'

Missing data plot from fineRADstructure: (we therefore only allowed 50% missing data)



RStudio script to visualize the fineRADstructure output:

###

A simple R example for plotting fineRADstructure output

Author: Milan Malinsky (millanek@gmail.com), adapted from a Finestructure R Example by Daniel Lawson (dan.lawson@bristol.ac.uk) and using his library of R functions

Date: 04/04/2016

Notes:

These functions are provided for help working with fineSTRUCTURE output files

but are not a fully fledged R package for a reason: they are not robust

and may be expected to work only in some specific cases - often they may require

at least minor modifications! USE WITH CAUTION!

SEE FinestructureLibrary.R FOR DETAILS OF THE FUNCTIONS

##

1) EDIT THE FOLLOWING THREE LINES TO PROVIDE PATHS TO THE fineRADstructure OUTPUT

setwd("~/PycharmProjects/pangolin/") ## The directory where the files are located

chunkfile<-"batch_1.haplotypes.tsv.fineRADpainter.lociFilt.samples50%missFilt_chunks.out" ## RADpainter output file

mcmcfile<-
"batch_1.haplotypes.tsv.fineRADpainter.lociFilt.samples50%missFilt_chunks.mcmc.xml" ## finestructure mcmc file

```
treefile<-
"batch_1.haplotypes.tsv.fineRADpainter.lociFilt.samples50%missFilt_chunks.mcmcTree.xml
" ## finestructure tree file

### 2) EDIT THIS PATH TO WHERE YOU WANT THE PLOTS:

plotsFolder <- "/Volumes/HELEN/RADseq edits Dec 2017/"

### 3) SET VALUES FOR THESE VARIABLES: "analysisName" will be included in output plots

analysisName <- "Second_run_pangolin";  maxIndv <- 10000; maxPop<-10000



### 4) EDIT THE PATH TO YOUR COPY of FinestructureLibrary.R

source("/Volumes/HELEN/RADseq edits Dec 2017/FinestructureLibrary.R", chdir = TRUE) #
read in the R functions, which also calls the needed packages


### 5) EXECUTE THE CODE ABOVE AND THE REST OF THE CODE BELOW

## make some colours

some.colors<-MakeColorYRP() # these are yellow-red-purple

some.colorsEnd<-MakeColorYRP(final=c(0.2,0.2,0.2)) # as above, but with a dark grey final
for capped values

###### READ IN THE CHUNKCOUNT FILE

dataraw<-as.matrix(read.table(chunkfile,row.names=1,header=T,skip=1)) # read in the
pairwise coincidence

###### READ IN THE MCMC FILES

mcmcxml<-xmlTreeParse(mcmcfile) ## read into xml format

mcmcdata<-as.data.frame.myres(mcmcxml) ## convert this into a data frame

###### READ IN THE TREE FILES

treexml<-xmlTreeParse(treefile) ## read the tree as xml format

ttree<-extractTree(treexml) ## extract the tree into ape's phylo format
```

Reduce the amount of significant digits printed in the posteror assignment probabilities (numbers shown in the tree):

ttree$node.label[ttree$node.label!=""] <-
format(as.numeric(ttree$node.label[ttree$node.label!=""]),digits=2)

convert to dendrogram format

tdend<-myapetodend(ttree,factor=1)

Now we work on the MAP state

mapstate<-extractValue(treexml,"Pop") # map state as a finestructure clustering

mapstatelist<-popAsList(mapstate) # .. and as a list of individuals in populations

popnames<-lapply(mapstatelist,NameSummary) # population names IN A REVERSIBLE FORMAT (I.E LOSSLESS)

NOTE: if your population labels don't correspond to the format we used (NAME<number>) YOU MAY HAVE TROUBLE HERE. YOU MAY NEED TO RENAME THEM INTO THIS FORM AND DEFINE YOUR POPULATION NAMES IN popnamesplot BELOW

popnamesplot<-lapply(mapstatelist,NameMoreSummary) # a nicer summary of the populations

names(popnames)<-popnamesplot # for nicety only

names(popnamesplot)<-popnamesplot # for nicety only

popdend<-makemydend(tdend,mapstatelist) # use NameSummary to make popdend

popdend<-fixMidpointMembers(popdend) # needed for obscure dendrogram reasons

popdendclear<-makemydend(tdend,mapstatelist,"NameMoreSummary")# use NameMoreSummary to make popdend

popdendclear<-fixMidpointMembers(popdendclear) # needed for obscure dendrogram reasons

########################

Plot 1: COANCESTRY MATRIX

fullorder<-labels(tdend) # the order according to the tree

```
datamatrix<-dataraw[fullorder,fullorder] # reorder the data matrix


tmpmat<-datamatrix

tmpmat[tmpmat>maxIndv]<-maxIndv #  # cap the heatmap

pdf(file=paste(plotsFolder,analysisName,"-
SimpleCoancestry.pdf",sep=""),height=25,width=25)

plotFinestructure(tmpmat,dimnames(tmpmat)[[1]],dend=tdend,cols=some.colorsEnd,cex.ax
is=1.1,edgePar=list(p.lwd=0,t.srt=90,t.off=-0.1,t.cex=1.2))

dev.off()


########################

## Plot 2: POPULATIONS AND COANCESTRY AVERAGES

popmeanmatrix<-getPopMeanMatrix(datamatrix,mapstatelist)


tmpmat<-popmeanmatrix

tmpmat[tmpmat>maxPop]<-maxPop # cap the heatmap

pdf(file=paste(plotsFolder,analysisName,"-
PopAveragedCoancestry.pdf",sep=""),height=20,width=20)

plotFinestructure(tmpmat,dimnames(tmpmat)[[1]],dend=tdend,cols=some.colorsEnd,cex.ax
is=1.1,edgePar=list(p.lwd=0,t.srt=90,t.off=-0.1,t.cex=1.2))

dev.off()

########################

## Plot 3: POPULATIONS AND COANCESTRY AVERAGES WITH PERHAPS MORE INFORMATIVE
LABELS

mappopcorrectorder<-NameExpand(labels(popdend))

mappopsizes<-sapply(mappopcorrectorder,length)

labellocs<-PopCenters(mappopsizes)

xcrt=0
```

ycrt=45

pdf(file=paste(plotsFolder,analysisName,"-PopAveragedCoancestry2.pdf",sep=""),height=25,width=25)

plotFinestructure(tmpmat,dimnames(tmpmat)[[1]],labelsx=labels(popdendclear),labelsatx=labellocs,xcrt=xcrt,cols=some.colorsEnd,ycrt=ycrt,dend=tdend,cex.axis=1.1,edgePar=list(p.lwd=0,t.srt=90,t.off=-0.1,t.cex=1.2),hmmar=c(3,0,0,1))

dev.off()

This was our final visualization of 80 samples (with 50% missing data removed).