Appendix

1 Proof of Lemma 2

Proof 1 *Let $\{(f_1, f_2), (f_2, f_3), ..., (f_{k-1}, f_k), (f_k, f_1)\}$ be the edges of a simple cycle c_s in G_F of length k fragments (vertices). We can partition the fragments into two sets such that each set corresponds to the haplotypes of the individual. If k is even, then we can partition the even fragments $(f_2, ..., f_k)$ and odd fragments $(f_1, ..., f_{k-1})$ into two sets such that each set does not contain internal fragment conflicts. Likewise, if k is odd, then no such partition exists because f_k conflicts with f_1 and f_{k-1}. The function that takes a cycle in G_C and computes the number of ${}^{01}_{10}$ (negative) edges is denoted $neg()$. We claim that for k even, $neg(c_s)$ is even and for k odd $neg(c_s)$ is odd. For this proof we consider any length $k-1$ subset of vertices in c and without loss of generality we assume this subset is $v_1, ..., v_{k-1}$. Consider any two adjacent fragments in this cycle f_i and f_j such that $i < j$ and they share the k^{th} SNP. As we iterate through fragments of the cycle, we call the allele that will be paired with the next fragment the active allele. If $(s_k, s_{k+1}) < 0$ then $f_{j,k+1} = f_{i,k}$, that is, the active allele that will pair with f_{j+1} is the same allele as $f_{i,k}$. However, if $(s_k, s_{k+1}) > 0$ then $f_{j,k+1} \neq f_{i,k}$, and the active allele that will pair with f_{j+1} will be the opposite allele as $f_{i,k}$. Thus negative edges in G_C do not change the active allele while positive edges in G_C flip the active allele from 0 to 1 (or vice-versa).*

Case (1): k even. The $v_1, ..., v_{k-1}$ subset either has an even or odd number of negative pairwise phase relationships. Case 1.a: Even number of negative pairwise phase relationships; odd number of positive pairwise phase relationships. The active allele of v_{k-1} is the same as the active allele of v_1 therefore v_k must be induce a positive pairwise phase relationship. Case 1.b: Odd number of negative pairwise phase relationships; even number of positive pairwise phase relationships. The active allele of v_{k-1} is different from the active allele of v_1 therefore v_k must be induce a negative pairwise phase relationship. In both cases 1.a and 1.b the total number of negative edges is even.

Case(2): k is odd. Case 2.a: Even number of negative pairwise phase relationships; even number of positive pairwise phase relationships. The active allele of v_{k-1} is different from the active allele of v_1 therefore v_k must be induce a negative pairwise phase relationship. Case 2.b: Odd number of negative pairwise phase relationships; odd number of positive pairwise phase relationships. The active allele of v_{k-1} is the same as the active allele of v_1 therefore v_k must be induce a positive pairwise phase relationship. In both cases 2.a and 2.b the total number of negative edges is odd.

2 MWVR Proof

Theorem: MWVR is NP-hard.

Proof 2 *The reduction is from the problem of removing the minimum number of edges of a graph to make it bipartite. Let G be an arbitrary graph and M the SNP-fragment matrix as defined in Lemma 1 which encodes the fragment conflict graph $G_F = G$. G_F may contain a number of cycles of odd length which produce conflicting cycles in the compass graph G_C by Lemma 2. Each vertex in G_C corresponds to an edge in G_F by Lemma 1. The vertex set solution to the MVR optimization L yields the minimum number of vertices required to remove all of the conflicting cycles in G_C. Because a graph is bipartite if and only if it contains no odd length cycles and G_C is the line graph of G_F, the removal of these vertices corresponds to removal of edges; the minimum of which makes G_F bipartite.*

3 Pacific Biosciences run times

	HapCompass MWER	HapCompass MEC	HapCUT	Levy
avg. time (s)	10	10.8	13.6	19.3
avg. memory (MB)	1251	1489	43.2	1049

Table 1: Average resource requirements for PacBio haplotype assembly runs. The HapCompass software is not optimized for minimal memory usage which is exemplified in the memory requirement results of the Levy *et al.* (2007) algorithm. This algorithm is implemented within the HapCompass software and should have a very small fingerprint but requires about a gigabyte of memory. Reducing the input fragment set into a secondary format prior to haplotype assembly (HapCUT does this) reduces our memory footprint by a factor of 10-100 times.

4 1000 Genomes Project Results

	MWER			**MEC**			**Levy**			**HapCUT**		
Chr	FMPR	BFM	MEC	FMPR	BFM	MEC	FMPR	BFM	MEC	FMPR	BFM	MEC
1	**3421**	**2348**	**2371**	3681	2519	2545	3619	2594	2632	3520	2423	2441
2	**4891**	**2930**	**2996**	5193	3081	3166	5154	3175	3273	5140	3022	3072
3	**3696**	**2394**	**2449**	4014	2585	2629	3823	2643	2703	3789	2476	2511
4	**4846**	**2710**	**2777**	5136	2906	2976	4891	2899	2974	4971	2805	2846
5	**3569**	**2245**	**2265**	3847	2428	2451	3851	2581	2606	3650	2290	2299
6	10425	**3603**	4032	10944	3846	4265	**9468**	3700	4075	10597	3630	**4030**
7	**3512**	**2138**	**2173**	3768	2288	2330	3677	2358	2407	3621	2214	2238
8	**2894**	**1864**	**1891**	3142	1999	2029	3048	2084	2118	2979	1947	1951
9	2844	**1551**	**1572**	3039	1667	1689	**2737**	1655	1687	2884	1580	1591
10	**2743**	**1857**	**1875**	2952	1981	2001	2838	2027	2048	2836	1932	1940
11	2662	**1634**	**1650**	2837	1727	1749	**2643**	1739	1778	2728	1694	1693
12	**2620**	**1627**	**1657**	2833	1784	1811	2786	1819	1856	2676	1678	1687
13	2503	**1461**	**1477**	2625	1554	1573	**2473**	1558	1576	2548	1490	1501
14	**1442**	**1020**	**1027**	1525	1070	1079	1512	1094	1102	1471	1045	1044
15	**1635**	**1085**	**1097**	1786	1168	1189	1757	1254	1272	1696	1133	1142
16	**2158**	**1308**	**1344**	2297	1410	1435	2198	1405	1458	2205	1333	1368
17	2797	1219	1320	3099	1354	1460	**2493**	1230	1305	2788	**1216**	**1299**
18	**1457**	**982**	**985**	1629	1088	1094	1563	1118	1130	1490	1013	1009
19	**1292**	**803**	**815**	1404	865	879	1369	901	918	1324	816	826
20	**1169**	**808**	**817**	1247	859	866	1279	924	939	1210	846	847
21	**871**	**545**	**558**	916	581	588	912	589	601	901	563	574
22	**681**	**446**	**449**	709	461	465	698	485	488	700	460	463

Table 2: Results of the NA12878 1000 Genomes Project 454 haplotype assemblies for chromosomes (chr) 1-22 and algorithms HapCompass MWER, HapCompass MEC, Levy *et al.* (2007), and HapCUT.

References

Levy, S., Sutton, G., *et al.* (2007). The diploid genome sequence of an individual human. *PLoS biology*, **5**(10), e254.