Supplementary Information

Side-chain determinants of biopolymer function during selection and replication

Phillip A. Lichtor[1,3], Zhen Chen[1,3], Nadine H. Elowe[3], Jonathan C. Chen[1,3], and David R. Liu[1,2,3]*

[1]*Department of Chemistry and Chemical Biology, Harvard University, Cambridge, Massachusetts 02138, USA.*

[2]*Howard Hughes Medical Institute, Harvard University, Cambridge, Massachusetts 02138, USA.*

[3]*Broad Institute of MIT and Harvard, Cambridge, Massachusetts 02142, USA.*

Correspondence to should be addressed to David R. Liu: drliu@fas.harvard.edu

Supplementary Table 1. Distribution of unfunctionalized and side-chain-functionalized building blocks in the top 100 most abundant polymers emerging from each pool in the PCSK9 selection. The theoretical distributions were calculated assuming an equal incorporation of all 32 building blocks based upon the number of building blocks used in each pool with side-chains or without. The actual distribution building blocks in the top 100 most abundant polymers was determined from HTS data.

	Unfunctionalized	Charged	Polar	Polar Plus	Nonpolar	Nonpolar Plus	Fully Functionalized
Theoretical							
Unfunctionalized	100%	75%	63%	50%	50%	38%	0%
Functionalized	0%	25%	38%	50%	50%	63%	100%
Found							
Unfunctionalized	100%	94%	88%	53%	58%	29%	0%
Functionalized	0%	6%	12%	47%	42%	71%	100%

Supplementary Table 2. Most abundant or enriched sequences from the PCSK9 binding selection. Sequences identified from HTS data that were most abundant after round 12 or exhibited the highest enrichment between any two selection rounds. Polymer sequences are shown without constant regions, which are CGAATCAGATTGGACCAG (pp2Z) and GAGTCCAGATGTAGGTAG (pp1A) on the 5'- and 3'-sides, respectively, of the polymer sequences shown. Functionalized side-chains are abbreviated as follows: histidine-containing, H; amine-containing, K; hydroxyl-containing, S; phenolyl-containing, Y; cyclopropyl-containing, 3; cyclopentyl-containing, 5; fluorophenyl-containing, F; isopentyl-containing, L.

Pool	Abund. rank	Polymer sequence	% of total	Enrichment Rank	$\%R_{max}$
Unfunctionalized	1	TTACCGCCCTCTCCCTTTTGTCCCCTCCGTTGTTTCTCTTTCCCT	0.172	24100	30
	2	CCGCTTTTGCCTTCTTACCCTTTTTGTCATCCCTGTTGTTTCCCT	0.093	10629	32
	3	CACCCTCCGTCTCCGCTCCCCTCGTCTTGTTGTTTCCCTTTCCCT	0.077	19741	34
	4	TTTTATCTTTTGCCACCCTTGCCACCTTGTTGTTGTTGTTGTCCT	0.064	23796	9.4
	5	TTTCCGCTCTCGCCCCCTTGTTTACTCTCTTGGTTCCCCTTCCCT	0.050	152	10
	63	CGCTTACCATTGTCTTGTTTGCCTTCCCCTTTGTCCTGTTGTTTA	0.015	2	37
	1754	CACTTCTCATTATTGTTCTTATGACCTTGCCCGTTTTCACCCCCT	0.002	1	25
	2223	TGCTTTTTCCTCTCTTATCACTTCTTGTTGCTTTGTTGTTTCTCT	0.002	4	7.4
	2407	CTTTCTTGTCGCCGCCCTCCTTGTTGTTGATGTCCTTCCTTTCCT	0.002	4	30
	4416	CGCTTTCTCTTTTTCCCTTTTCCATGCCCATGTTTATCTTTCCGC	0.001	3	32
Charged	1	TTTTGTTGCCACCCTTGTTGTCCCCCTHCACCCCCACGTTGTCCT	1.219	13282	20
	2	TTTCATCCCTTTTTGCCACCCCTTTGTTGTCCACCCCCATGTCCT	0.570	1434	11
	3	TTTCTCCCTCTTCCACACCTTTGTCTTTGCCACCCTCCTTGTCCT	0.451	2703	22
	4	TGATTTTGCHAACGTCTTHCACTCHCCCAACCGCACCCTCCACCG	0.445	41	9.5
	5	TTTCTTTAGCCACCCTGCCCTTGTTAGCACCCTTGTTGTTGTCCT	0.427	58	n.d.
	47	CTTTTTCTTTGTKCTHCACCCCTTCCGTGCCCCTTTHCACCCCCT	0.071	3	10
	167	CGTTTTTTGCTTCTTCTATGCCCTTTTCTCTGCCCGTTTCCCCGC	0.033	2	9.8
	168	TAGTTGCCCCCTCGTTTTHCCCCTTGTCACCCTTGTTTTTGTCTC	0.033	3	18
	928	CACTAGCCTTTTTGCCCCTTTCTTTGTCGTCTTCCCTGTTTTCTC	0.008	1	10
	10736	CCGTTTHCCCCGCTTTATCCCCGGTTTCCATGTTTTTGTCCTCCT	0.001	5	n.d.
Polar	1	TTGTTGCCCTCGCCTCCACTCCCGCCTCCTHCCCCGTGTTGTCCT	0.6155	4356	24
	2	CCCTTTCTTCCTHCCCCCSGCCTCHCCCTATTGCCATGTTGTCCT	0.5872	55895	19
	3	TTTCCGCCACCACCGCCTCTTHCCCTCTTTCCTTGTTGTCCCCCT	0.3706	6601	15
	4	TTTTTTCCTTGTCAGCCTCTCTTGCCCCCGCCTCCTTGTTGTCCT	0.3559	64417	13
	5	CCGCCTCACCTTHCCCTTTGTKTCCCTTGTTGTCTTTGTKTCCCT	0.3463	4087	n.d.
	31	TTTHCCCCTTTTCTCCCACCGTTTSATTGTCTTTTTCCCTGTCCT	0.1183	2	7.4
	1239	CAGSATKCTHCACATTTTCCTTGTTTTCCCCCTTGTKCTCGTSAG	0.0068	4	19
	101845	SGACGGCACHACCCTCGTTTTHCCCCCTGTTTTCTTCCTCTCHAC	0.0002	5	3.4
	n.d.	CCTTTTHCCHCCCGGSATCTCCCGTTTHCCCCTTGTKCTCGTSAG	n.d.	1	8.7
	n.d.	CGGHACCCATTTCTTCTCSGAHCCTGTCTTTTTTGTKCTCGTSAG	n.d.	3	10
Polar plus	1	CCACCCYTTYTTCCTYTTYTGCCTYTTCTTCTTCCTCCCYTGCTA	4.100	70	73
	2	YCGCTAYTTCCTYTTCTCCCCYTTYGTCCCYGTCCCYTTCCACTA	2.536	94	49
	3	CCCYTTYGTYTTCTCSGCCCTCGTYTGCCTYTGCCCCGCCACCCT	2.277	343	35
	4	YGTYTTYTTCCCKTCYTGCCTYTTHCCCCGYTGCCACCCCTTCCT	1.798	632	n.d.
	5	YTTYTTCCTCTTYTGCCCCGTHCCCCTYGTCCAYTGCCACACCCT	1.717	10718	n.d.
	25	CGCYTTYGTYTTYTGCCCYTTCACCCTHCCYTGCCTCTTCCGCCA	0.338	2	22
	38	CCTCTGYTGYTTYTTHCCCACYTTCTTCCCYTTCACCCAYGTCCT	0.248	4	n.d.
	64	CTTCTAYTTYTTCCCYGTYTTYTGCCTYTTCCTYGTYTGCCCCGC	0.148	4	35
	66	YCGHACYTTYTTCCGCCTYTTSGCCCTYTTCTCCCCCTTYTTCCT	0.143	1	28
	445	HCACCTYCGYCGCGCCCCCCCYTTYGTCCTYTGYTTCTTCTCCCT	0.022	3	33
Nonpolar	1	TAGTTTFGATAT3TATCTTGCLTCTCGFAT5GTTGTTCA3CT5CG	21.861	16	17
	2	TTT5GTTTCTCA5TTTCTTTT3CATTCTGTTCCTTCTGTTCT3TA	17.723	70	44
	3	TAGTTTFGATAC3TATCTTGCLTCTCGFAT5GTTGTTCA3CT5CG	6.608	2	10
	4	TAGTTTFGCTAT3TATCTTGCLTCTCGFAT5GTTGTTCA3CT5CG	1.086	6	n.d.
	5	TAGTTAFGATAT3TATCTTGCLTCTCGFAT5GTTGTTCA3CT5CG	0.902	3	5.2
	15	TAGTTTTGATAT3TATCTTGCLTCTCGFAT5GTTGTTCA3CT5CG	0.320	7	n.d.
	20	TAGTTTFGATAT3TATCCTGCLTCTCGFAT5GTTGTTCA3CT5CG	0.292	1	5.7
	51	TTT5GTTTG5TTTCTTTT3CCTTCTGTTCTTTG3TATAC3CATCG	0.068	4	15
	61	TAGTTTFGATAT3TATCTTGCLTCTCGFAT5GTTGTTCC3CT5CG	0.057	5	6.7
Nonpolar plus	1	FAG5GTYTGFATTGC3CT3CC5TGTCCTTC3CCTAT3CC5CGTCC	26.0222	250	72
	2	3CCLTCTCTYGTYGTTCC3CCYTTTCCYGTTCC3CTYTGTTC3CT	4.1645	145	71
	3	TACLTCYGTYTGTCC3CCTTATCTTTC3CC5TGYGTTCCYGTTAA	1.6224	301	35
	4	YTG3CCYTTYGTYGTTTCLTCTCCYCG3CC3CTYGTTTC3CTTGA	1.5915	1471	n.d.
	5	FAG5GTYTGFATTGC3CT3CC5TGTCCTTC3CATAT3CC5CGTCC	1.2917	295	n.d.
	88	5CGFGA5TG3CCYTTTCC3CTTACFGC3CAFGC5TGYGTTATLTC	0.0799	2	21
	395	TGCYTT5TTYTT3CT3CC5CGYTTTTCTTCYGTTTA3CCTCA3TA	0.0187	4	37
	2478	YTG3CCYTTYGTYGTTTCLTCTCCYCG3CC3CTYGTTTC3CTTGC	0.0028	1	29
	14282	5TTTCTTCAYGTTTC3CTYTTYGT5TT3CC3CCTCTTCC3CCTTA	0.0003	5	n.d.
	21725	YTT5TGYTGYGTTTC3CTTCCYTGTCTTCCYGTTTATCC3CCTCC	0.0003	3	24

Fully functionalized	1	5GTYTGYTGSGC3CTLTCLTC3CTLTCSGC5TGLTCHAC3CT3CA	26.957	162	62
	2	5GTYTGYTGSGC3CTLTCLTC5CGLTCSGC5TGLTCHAC3CT3CA	6.400	187	78
	3	3CT5CGYTTYTT5TT3CCYGTYTT3CC3CA5GT3CC3CTSATYGT	5.745	3430	47
	4	YTTYGTSAT3CAHCC3CCYGTYTT3CCHCCYGTYGTLTC3CC3CT	2.494	1776	54
	5	YCGYTTYGTYTT3CTHCCYGTYTGLTCLTCLTCHCC3CTYGT3CT	2.213	43	n.d.
	9	YTTYTGYCGYTTSGC5TTHCC3CCYTG3CTLTC3CT3CC5TTLTC	1.103	2	70
	32	YTT5CG5TTYTT3CCSGC3CTYGTYTG3CC5TG3CC5TTSGA3CA	0.291	5	36
	74	YCG3CC3CTYTT5TTYGTYGTYTTHCCFATSGC3CTSATYTT3CA	0.109	3	72
	99	YCG5TTHCC3CT3CCYGTYTT5TT3CCYGT5TT3CC3CTYTGKTA	0.082	1	66
	391	5TTSGC5TT3CCYGTYTT3CTLTC3CTYGT3CTYGT3CC5TTHCA	0.015	4	48

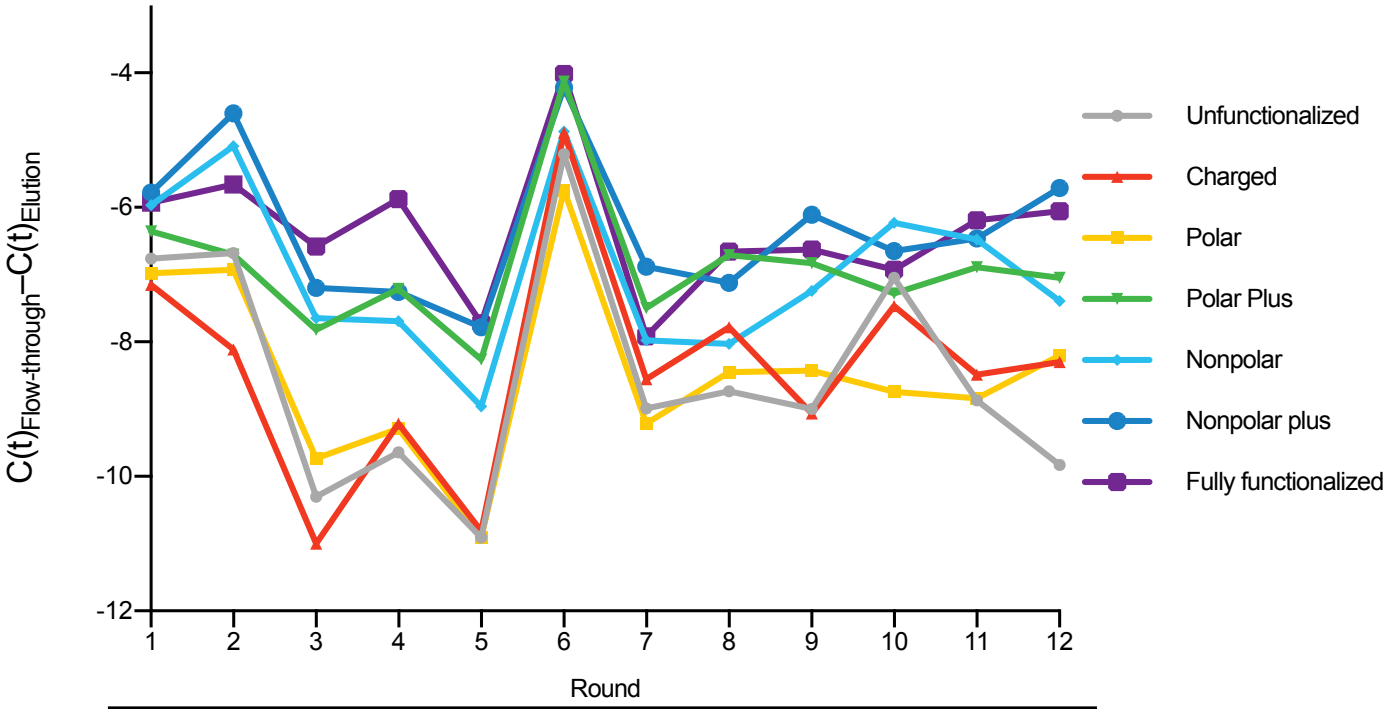
Supplementary Table 3. Distribution of unfunctionalized and side-chain-functionalized building blocks in the top 100 most abundant polymers emerging from each pool in the IL-6 binding selection. The theoretical distributions were calculated assuming an equal incorporation of all 32 building blocks based upon the number of building blocks used in each pool with side-chains or without. The actual distribution building blocks in the top 100 most abundant polymers was determined from HTS data.

	Unfunctionalized	Charged	Polar	Polar Plus	Nonpolar	Nonpolar Plus	Fully Functionalized
Theoretical							
Unfunctionalized	100%	75%	63%	50%	50%	38%	0%
Functionalized	0%	25%	38%	50%	50%	63%	100%
Found							
Unfunctionalized	100%	78%	68%	50%	64%	44%	0%
Functionalized	0%	22%	32%	50%	36%	56%	100%

Supplementary Table 4. Most abundant or enriched sequences from the IL-6 binding selection. Sequences identified from HTS data that were most abundant after round 7 or exhibited the highest enrichment between any two selection rounds. Polymer sequences are shown without constant regions, which are CTCGGATGAACCTGGACT (pp2W) and GCATCGAAGCCAAGATTC (pp1B) on the 5'- and 3'-sides, respectively, of the polymer sequence shown. Functionalized side-chains are abbreviated as follows: histidine-containing, H; amine-containing, K; hydroxyl-containing, S; phenolyl-containing, Y; cyclopropyl-containing, 3; cyclopentyl-containing, 5; fluorophenyl-containing, F; isopentyl-containing, L. Not determined (n.d.).

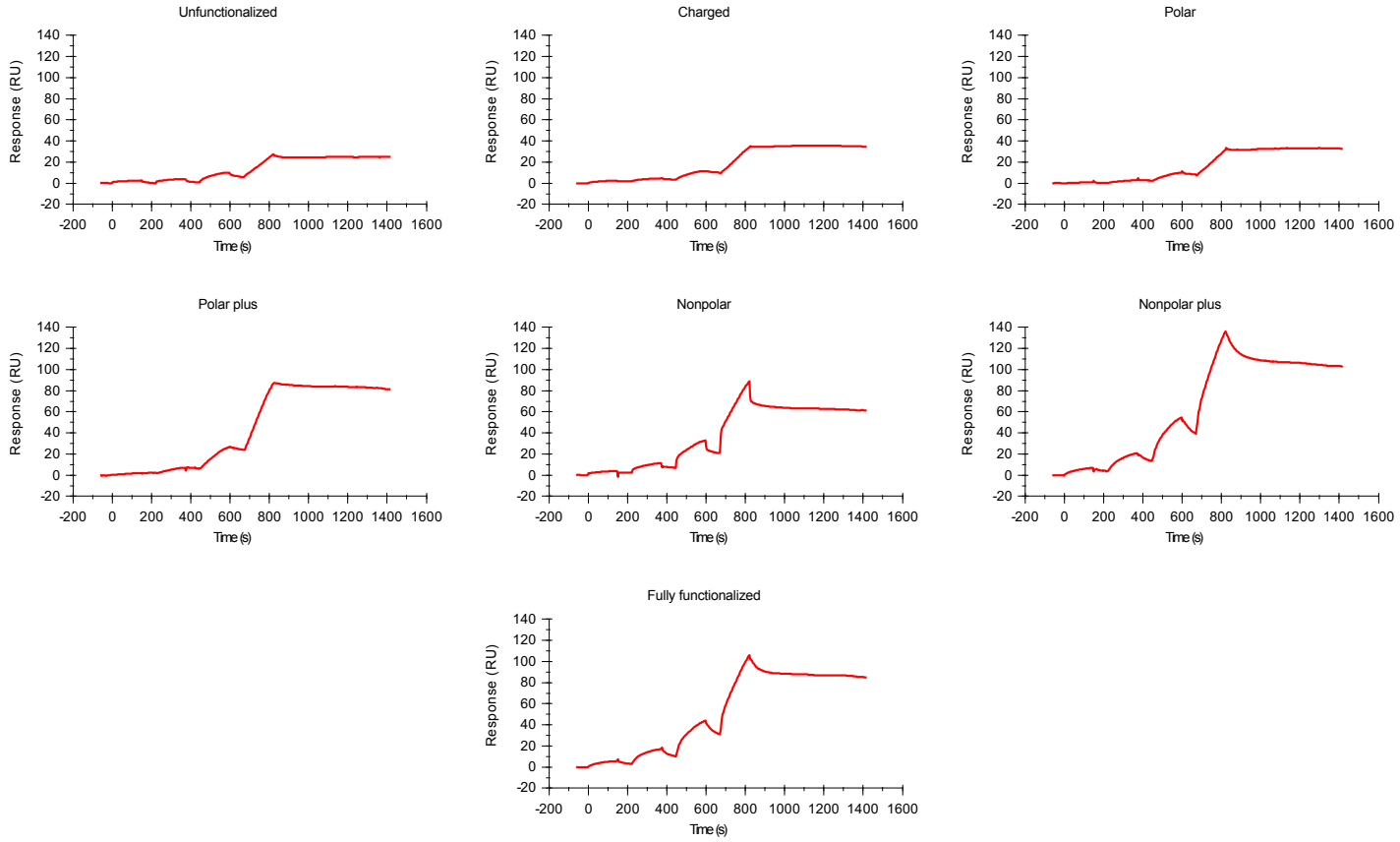
Pool	Abund. rank	Polymer sequence	% of total	Enrichment Rank	%R_{max}
Unfunctionalized	1	TTATTGCACTGGCTGTACTTTTCGTATCCATGGTCATCTTTTCTA	3.830	34	4
	2	TTACTGCGGCGCCAGCCGTGCCCATCCTTTCTTCATCCGTGCCCA	1.741	10	3
	3	CTGTTTTTACTACCGCGCTAGCCGCCTCATCCGTTTTAGCGCCAT	0.324	105	3
	4	TAATGATGCTCATGGTGCTCTTCATCACCATCTTCGTTCCATCCT	0.320	16851	4
	5	TCTTCATTGCGCTGGTGCCTGCCATCCTTGTTCCATCCCCTTTGC	0.236	9068	4
	9	TTATTGCACTGGCTGTACTTTTCGTATCCATGGTCATCTTTCCTA	0.118	1	-1
	46	TTATTGCACTGGCTGTACTTTTCGTATCCATGGTCATCTTTTCTG	0.034	5	4
	55	TTATTGCACTGGCTGTACTTCTCGTATCCATGGTCATCTTTTCTA	0.030	4	5
	58	TTATTGCACTGGCTGTACTTTTCGTATCCATGGTCATCCTTTCTA	0.029	3	5
	232	TTACTGCGGCGCCAGCCGTGCCCATCCTTTCTCCATCCGTGCCCA	0.011	2	0
Charged	1	CTGCAATGCKTAKGGHCCKGGCGTTAGTGTTGTTGCKTCCACCGG	9.294	1	4
	2	TCGCGAKGGTCGTGCHCCCGATTTCTAKTACCGCGCTAGCGGCTC	1.812	20	5
	3	CTGCAATGCKTAKGGHCCKGGCGGTAGTGTTGTTGCKTCCACCGG	0.647	4	1
	4	TAGKGGCTTCTATGCTGCCTGCGCTTGHCACCCCGCCCTCTTCGT	0.406	255	-3
	181	TCGCGAKGGTCGHACHCCCGATTTCTAKTACCGCGCTAGCGGCTC	0.013	3	6
	294	TCGCGAKGGTCGKGGHCCCGATTTCTAKTACCGCGCTAGCGGCTC	0.010	2	2
	788	CTGCAATGAKTAKGGHCCKGGCGTTAGTGTTGTTGCKTCCACCGG	0.006	5	2
	985	TCGCGAKGGTCGTGCCCCCGATTTCTAKTACCGCGCTAGCGGCTC	0.005	6	3
	25	TCGCGAKGGTCGTGAHCCCGATTTCTAKTACCGCGCTAGCGGCTC	0.070	7	2
Polar	1	KGGCCCCCTHAATGTTGTHACCTCHCASGAKCTCGGSGCCCGCCC	14.160	3	4
	2	CGGCAATTTCTTTCGKTCKTACTGKTCCTTCCCTTGCCCCCAHCC	8.281	6	2
	3	HCATGTKGGCTCSGATGTCCTTCGKTCCTACTGKTCCATCCTCCG	3.552	11	4
	4	CTGCCTSGCKTACGGTCGSGATTGCACSATTGTCCAHCCCCTCTT	0.671	49	n.d.
	5	CTGSATKGGCTGSGACCCSGCTTGCCCCTCTTGCGTTGTCATCCC	0.655	13	n.d.
	7	HCATGTKGGCTCSGASGCCCTTCGKTCCTACTGKTCCATCCTCCG	0.294	7	2
	10	CGGCAATTTCTCTCGKTCKTACTGKTCCTTCCCTTGCCCCCAHCC	0.167	1	3
	15	HCATGTKGGCTCSGATGTCCTTCGKTCCTACTGKTCCACCCTCCG	0.119	2	2
	35	KCTTGTKGGCTCSGATGTCCTTCGKTCCTACTGKTCCATCCTCCG	0.059	5	2
	130	KGGCCCCCCHAATGTTGTHACCTCHCASGAKCTCGGSGCCCGCCC	0.021	4	2
Polar plus	1	YTTCGAYTGCCAYTGSGCYGTCATCCCCGTHCASGAYGTCGACTG	10.140	57	9
	2	KTCCCTHAACTGYTGCTGSATYTTSGCYTGSGAKCTCATCCCCTG	7.954	6	5
	3	CCAHCACGGCTTYTTYCGKTCKTACTGKTCCACCCTKTCCCCCGT	2.146	5	7
	4	CCAHAACTGYGTCGACTACGGHCCHCCCGACTGYGTKTCCATYGT	1.476	30	n.d.
	5	CAGSGAKCTSATCCAHCACCTYGTCATCGCYTGCTGYGTHCCKGG	0.516	74	n.d.
	23	YTTCGAYTGCCAYTGSGCYGTCATCCCCGCHCASGAYGTCGACTG	0.083	3	-2
	68	HCCCTGSATKGGCTTCAAKCTCCAKCTCTGCCTCACCCCHCACCC	0.036	7	1
	469	CGGYTGCTGCTTYTGYTTYCGKTCKTACTGKTCCTTCTCCGTCGC	0.011	4	2
	2262	CCAHAACTGYGTCGACTACGGHCCHCCCGACTGYGTKTCCACYGT	0.004	2	-2
	26647	CCTKTAHCAHAAHACKGGSGAKCTYTGCTGKGGCGAYTGCTCCGG	0.001	1	-2
Nonpolar	1	3CTTTATCATAATACTGGTGATCTTTG5TGTGGFGATTGLTCLGG	35.775	13	83
	2	TACTTATAC3CTTTG5GTTTATGTTGG5TGTGGFGATTGLTCLGG	10.110	16	66
	3	LTC3CATGGTTTTGGTCCTAT3TATGG5TGTGGFGATTGLTCLGG	9.371	9	83
	4	5TG3CATGGTTTTGGTCCTAT3TATGG5TGTGGFGATTGLTCLGG	7.403	53	n.d.
	5	3CTTTATCATAATGCTGGTGATCTTTG5TGTGGFGATTGLTCLGG	5.254	3	n.d.
	27	TTCTATTCT5TTTGCTGGFGATAT5TGTGCTTTTGTTTA3TA5TG	0.159	1	1
	28	3CTTTGTCATAATGCTGGTGATCTTTG5TGTGGFGATTGLTCLGG	0.139	6	43
	67	TTCTTGTAC3CTTTG5GTTTATGTTGG5TGTGGFGATTGLTCLGG	0.038	4	54
	87	TTTTAGTGTTCTTCCTTGTTC3CTTGGTTGTGGFGATTGLTCLGG	0.029	2	n.d.
	11	5TGTGGTATTCGTTGTCCLTCTCC5CGTGC3CCTCALTCTCCFGA	1.744	5	1
Nonpolar plus	1	TCA5CG3CALAALGGLTCTTCYCGFGCYTTTAGTGATGA3TATGG	3.646	4	17
	2	FGCFGA3CCYTGYTGYTTTTCTCTTCCTGCTTCTGAYTGLTCLGG	3.191	208	103
	3	TGGTGGYTGYCGFGC5CGFGCTGATAC3CT3CTTCATGATGCTTC	2.412	43	15
	4	3TATTCTATTGCTTATATTCTTCCYTGTGGTGATGAYTGLTCLGG	1.785	114	90
	5	3CT5CGTGAYGTYGTTCC3CTTTCYTGYGT5TTTGAYTGLTCLGG	1.185	163	n.d.
	100	3TATTCYGTTGCTTATATTCTTCCYTGTGGTGATGAYTGLTCLGG	0.074	6	76
	132	FGCFGC3CCYTGYTGYTTTTCTCTTCCTGCTTCTGAYTGLTCLGG	0.051	5	70
	135	5CG5CGYTTYTTYTGTCTTCATACTCATGCTTCTGAYTGLTCLGG	0.051	2	37
	230	TGGFGCTGAYTT5TTTGCTACYTG3TAYTGTGGYGT5GT3CATGG	0.031	1	4
	237	TGGTGGFGCYTTFAGTGG3CTLTC5TGTCATAT3TA3CAFGCTGC	0.030	3	2

Fully functionalized	1	HCALGG3CTKCTFGA5TGSGC3CCYCGFGC3CAYTGHAC3TASGC	47.467	2	17
	2	YCGYGTKCTYGTYTGKGGYTGSGCYTTYTG5TGYTG5GTLTCFGC	19.972	9	98
	3	HCAFGA3CTKCTFGA5TGSGC3CCYCGFGC3CAYTGHAC3TASGC	13.238	4	13
	4	YCGLGG3CTKCTFGA5TGSGC3CCYCGFGC3CAYTGHAC3TASGC	8.110	3	10
	13	YCG5GTYTTLGGKTASGCKTCLGGKTAHCC5CGLGGHCCKCTKGG	0.175	8	8
	30	HCALGG5CGKCTFGA5TGSGC3CCYCGFGC3CAYTGHAC3TASGC	0.074	1	1
	32	KCT3TAHCA3CTKCTKGG5GT5GTKTALTCFAT3CAYTTHAC3CA	0.072	5	n.d.
	44	KCT3TAKCT3CTKCTKGG5GT5GTKTALTCFAT3CAYTTHAC3CA	0.048	7	n.d.
	60	HCALGG3CTYCGFGA5TGSGC3CCYCGFGC3CAYTGHAC3TASGC	0.030	6	1

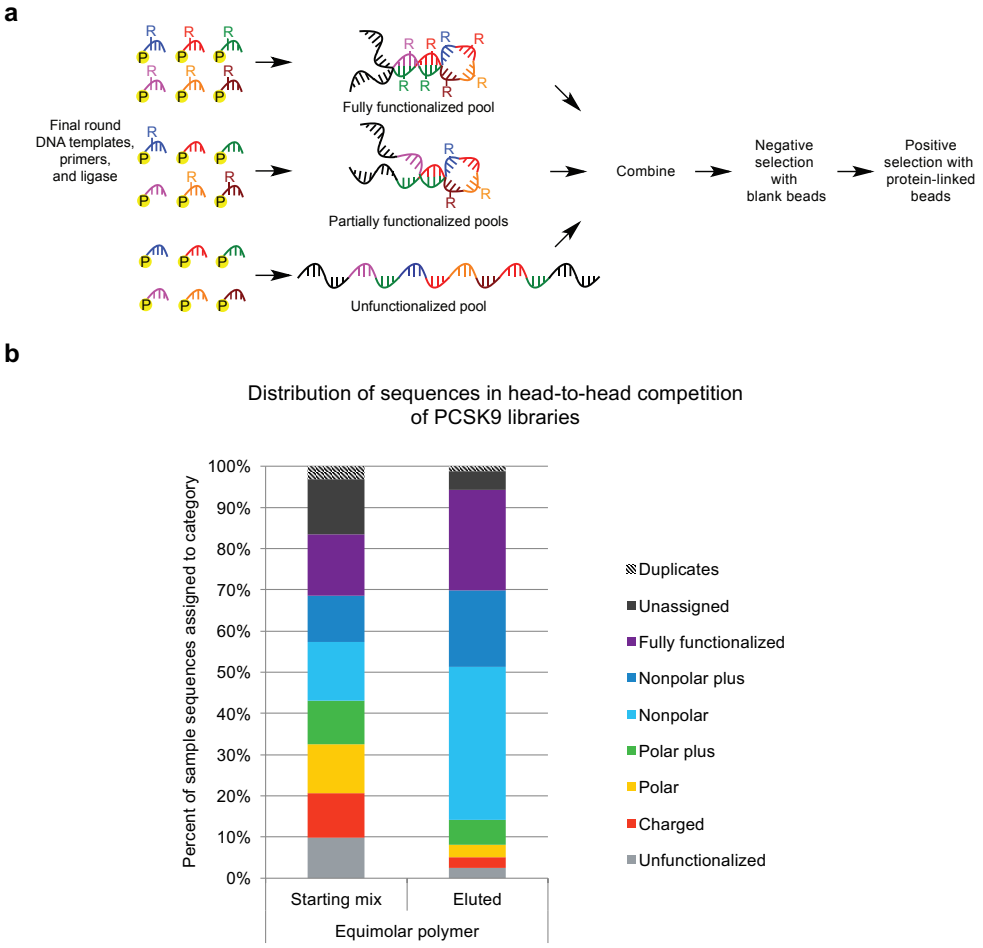


Template (pmol)	125	1	1	1	1	1	1	1	1	1	1	1
PCSK9 (pmol)	125	13	3.7	3.7	3.7	1	1	1	1	1	1	1
Neg Blank beads (µL)	-	-	-	-	-	-	1.4	4	6	6	6	8
Time (h)	1	1	1	1	1	1	1	1	1	0.75	0.5	0.5

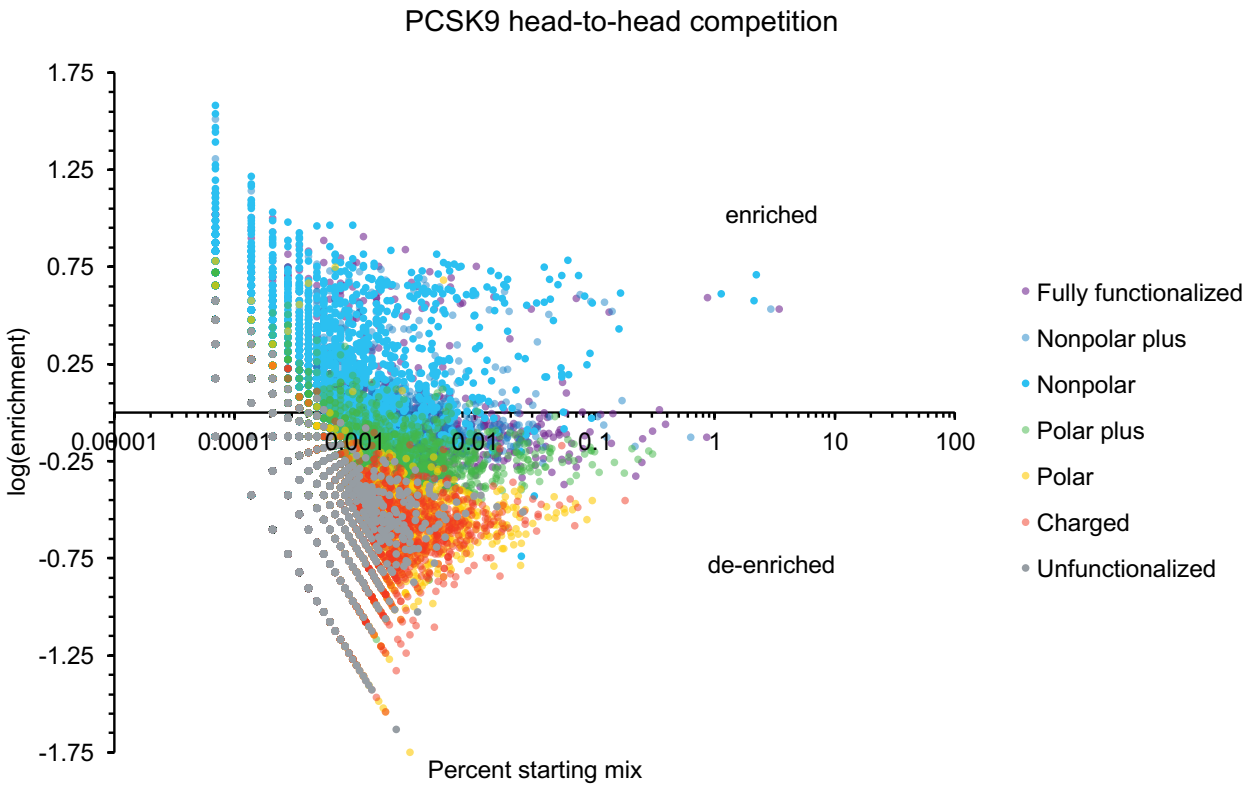
Supplementary Figure 1. PCSK9 selection conditions and progress. Population-level retention of each library on PCSK9-linked beads over seven rounds of selection, as measured by qPCR. Higher values indicate a greater proportion of amplifiable material eluted from the PCSK9-bound beads relative to the unbound material in the flow-through. The table indicates the selection conditions used at each round.



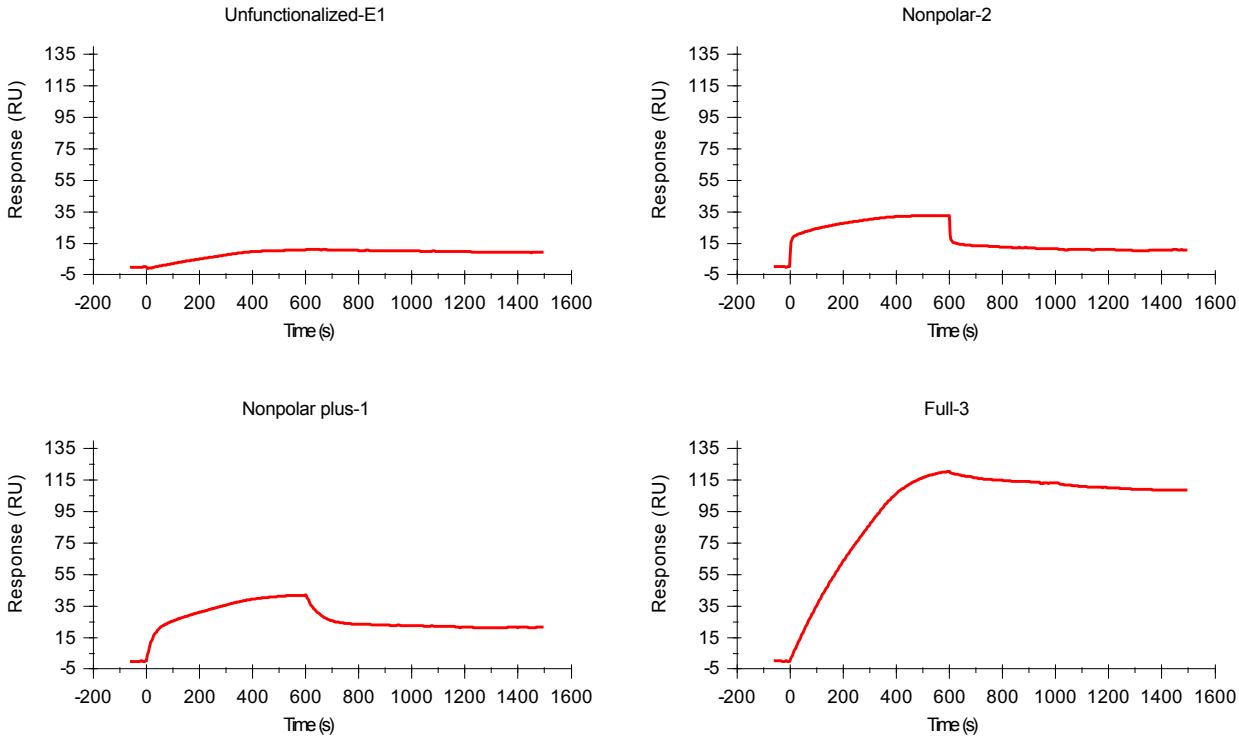
Supplementary Figure 2. PCSK9 affinity of selection-enriched polymer pools. Representative SPR sensorgrams of single cycle kinetics run of different 3'-biotinylated enriched pools following selection round 12. The polymer pools were bound to the surface of a CAP chip and concentrations of 7.3, 22, 67, and 200 nM PCSK9, each with 0.05 mg/mL BSA, were flowed over the chip's surface at 30 µL/min in HBS-P+ buffer with $CaCl_2$, $MgCl_2$, and KCl. Data in Fig. 2a were plotted based on the response at third binding point, at the end of the 200 nM injection of PCSK9 to minimize any influence of reference binding in the pool comparison. Sensorgrams shown represent the first of three replicates collected with the enriched polymer libraries and PCSK9 shown in Fig. 2a.



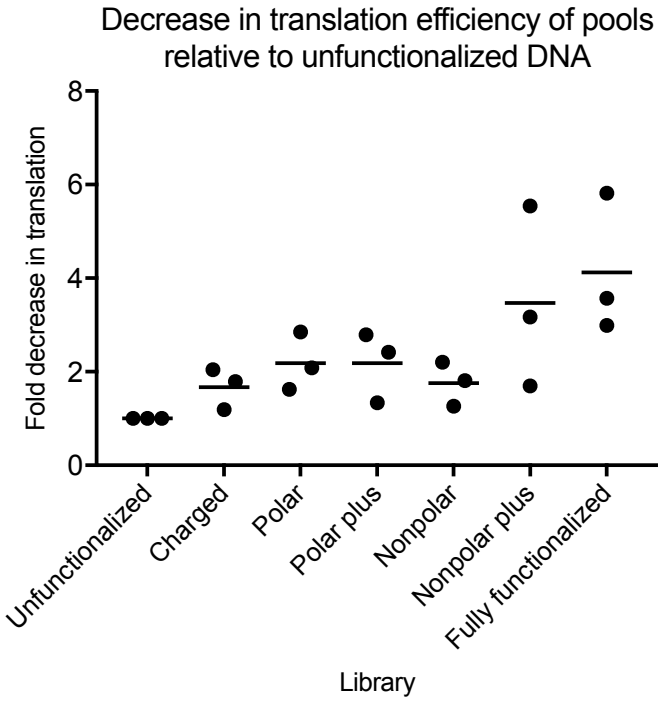
Supplementary Figure 3. Analysis of mixed samples in polymer pool competition. a, Schematic representation of head-to-head competition. **b**, Stacked bar plot showing the distribution of sequences from the starting and eluted material from the head-to-head competition experiment in which a round of selection against bead-bound PCSK9 was performed. Sequences are assigned to a pool of origin based upon the occurrence of the same sequence in any of the 12 previous selection rounds. Sequences that were not previously found from HTS analysis of previous rounds are labeled as unassigned. Sequences appearing in multiple pools where assignment was ambiguous were removed, but labelled as duplicates, so bars add to 100%.



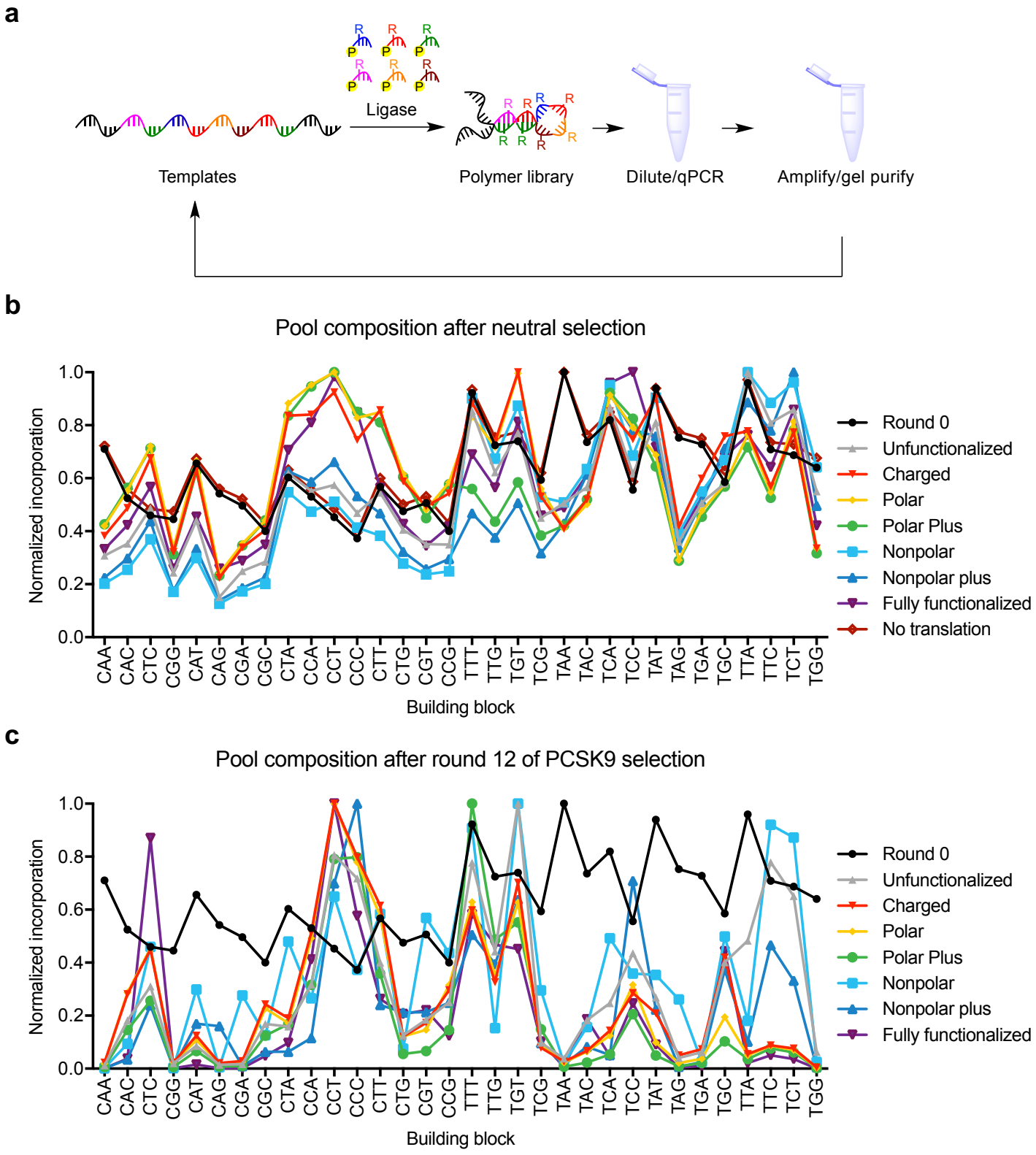
Supplementary Figure 4. Individual polymer enrichment in PCSK9 competition experiment. Each point represents a sequence that could be assigned to a pool and was identified in both the starting and eluted mixes.



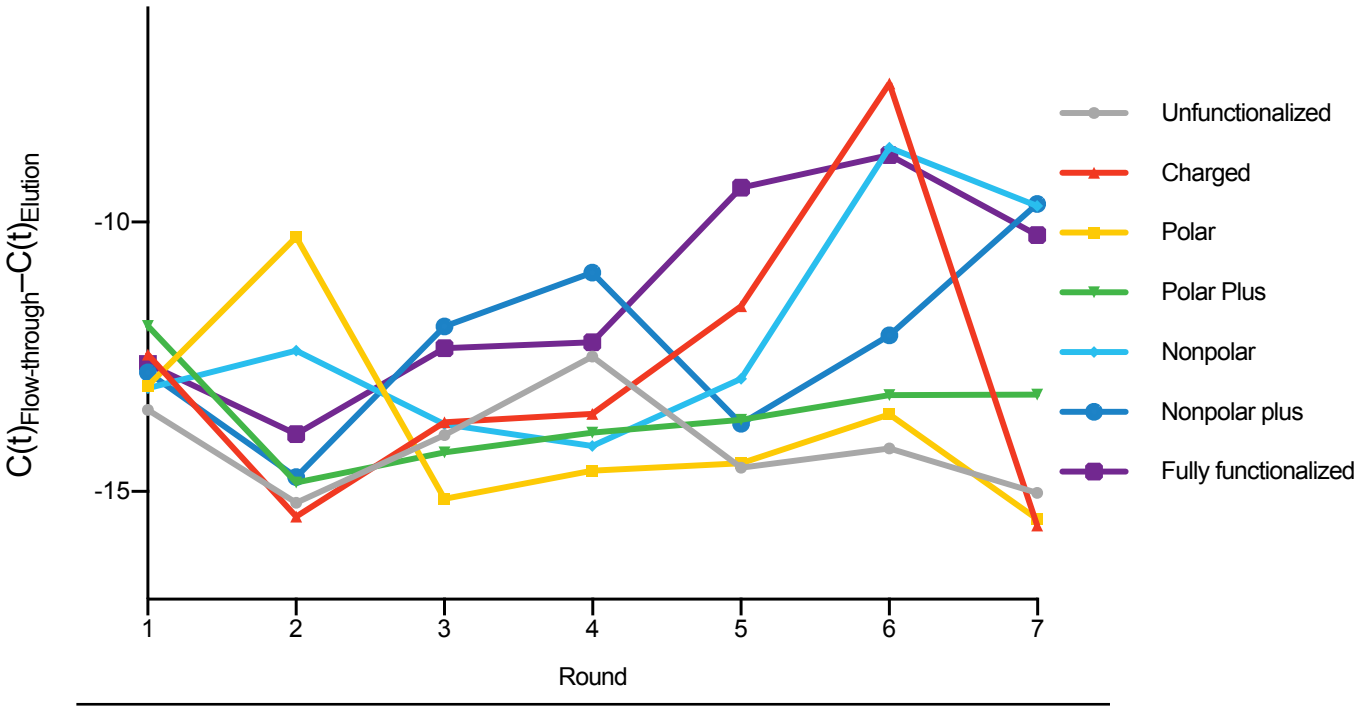
Supplementary Figure 5. Individual polymer binding to PCSK9. Reference- and background-subtracted sample sensorgrams in response to a single 600-s injection of 60 nM PCSK9. Polymer Unfunctionalized-E1 is the polymer exhibiting the greatest enrichment from any round in the IL-6 selection from the unfunctionalized (DNA) library. Polymers Nonpolar-2, Nonpolar Plus-1, and Full-3 are the 2nd-, 1st-, and 3rd-most abundant sequences from the nonpolar, nonpolar plus, and fully functionalized pools following PCSK9 selection, respectively. Data presented are representative of sensorgrams used to derive data in Fig. 4a and Supplementary Table 2. For this measurement, most polymers were analyzed once. For the above sensorgrams, Full-3 was analyzed twice, giving similar results.



Supplementary Figure 6. Translation efficiency of each pool. Calculated fold decrease in polymer populations as determined by qPCR following translation of the first-round libraries for the PCSK9, IL-6, and neutral selections, before any selection step has taken place. The horizontal bars show the mean values of three replicates; individual values for each replicate are shown. Assessment of translation efficiency assumes the PCR amplification rates of the polymer pool templates is equal.

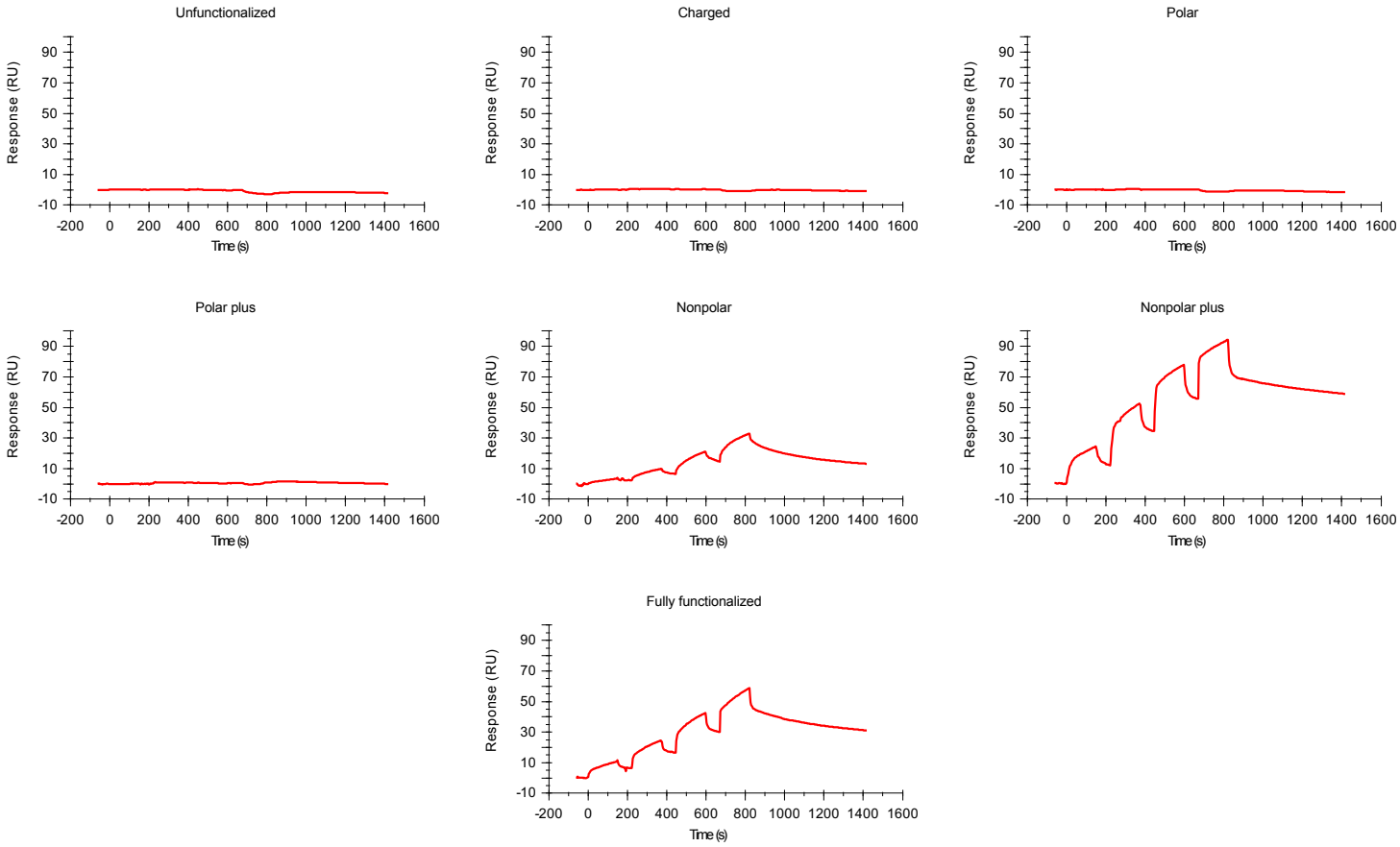


Supplementary Figure 7. Changes in codon distribution. a, Schematic representation of a "neutral selection" with a cycle of translation, reverse translation, and amplification. The seven different pools underwent two rounds of neutral selection and were interrogated by HTS. **b**, Plot of the distribution of the codons following blank selection with each of the different building-block sets (pools). Each pool's data is normalized to the most abundant codon in each library. The "no translation" sample shows data from two rounds of just dilution and amplification. **c**, Plot of the distribution of the codons following round 12 of the PCSK9-targeted selection.

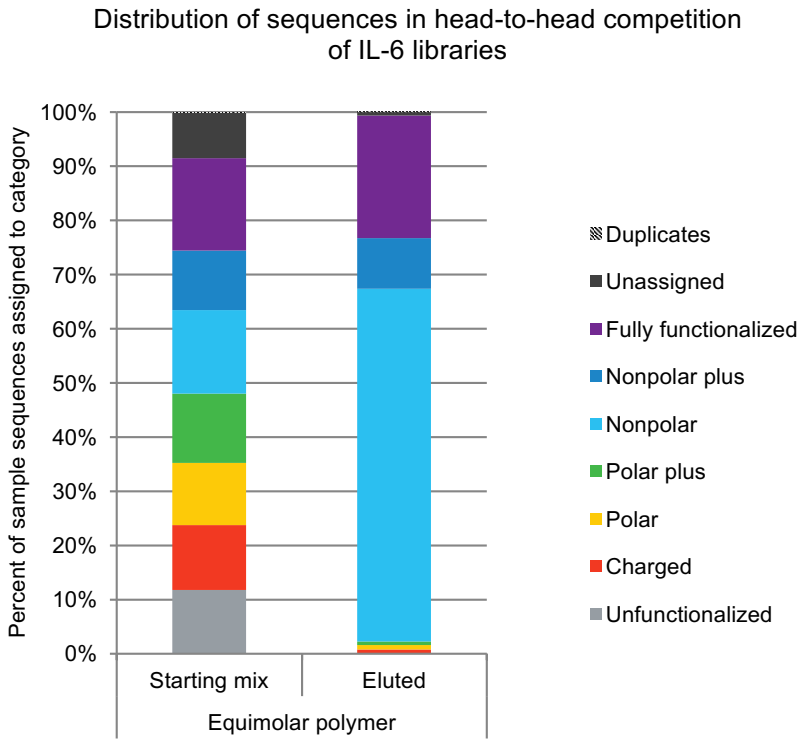


Template (pmol)	125	5	1	2	2	2	2
IL-6 (pmol)	125	1	1	2	2	2	2
Neg Blank beads (µL)	-	-	2	4	4	4	4
Time (h)	1	1	1	1	1	0.83	0.66

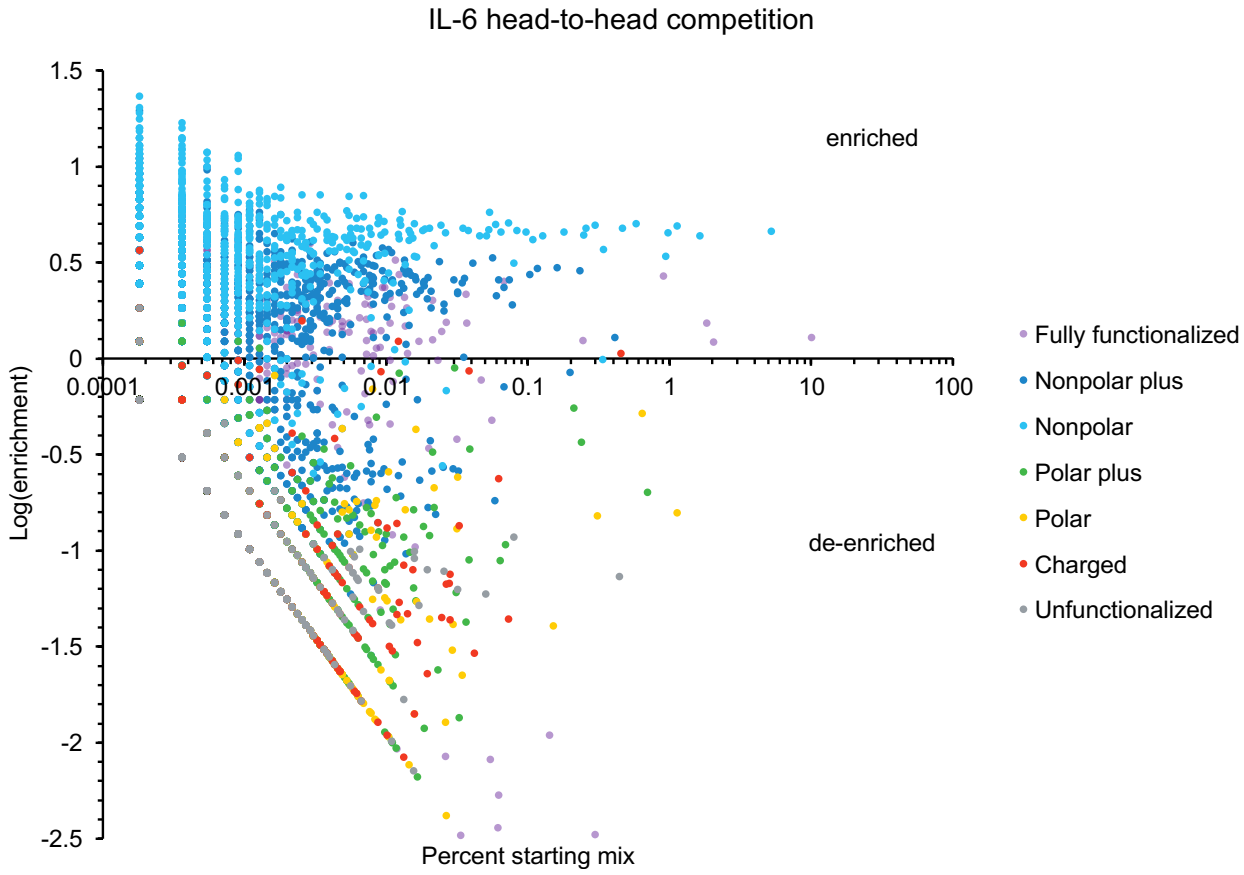
Supplementary Figure 8. IL-6 selection conditions and progress. Population-level retention of each library on IL-6-linked beads over seven rounds of selection, as measured by qPCR. Higher values indicate a greater proportion of amplifiable material eluted from the IL-6-bound beads relative to the unbound material in the flow-through. The table indicates the selection conditions used at each round.



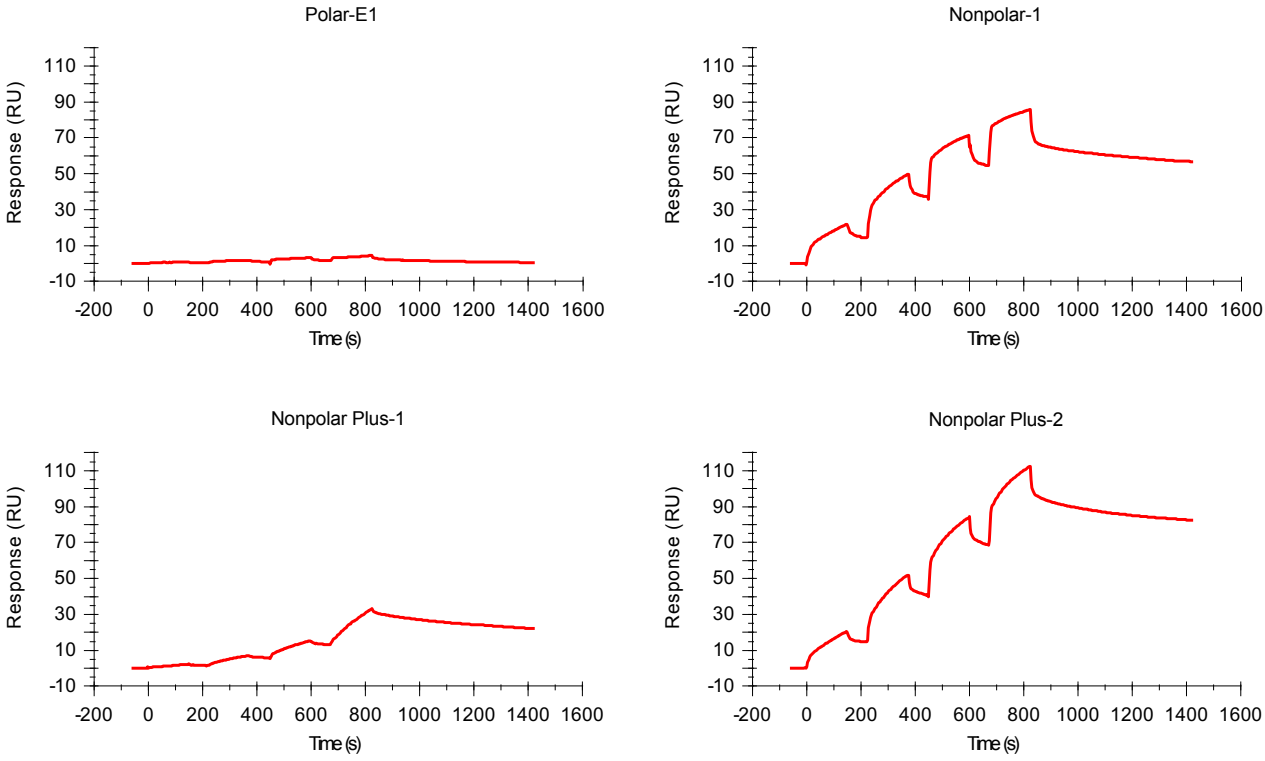
Supplementary Figure 9. Affinity of selection-enriched polymer pools to IL-6. SPR sensorgrams for single cycle kinetics run of different 3'-biotinylated enriched pools following selection round 7. The polymer pools were bound to the surface of a CAP chip and concentrations of 8.9, 27, 80, and 240 nM IL-6 were flowed over the chip's surface at 30 µL/min. Data plotted in Fig. 2c were plotted based on the response at the third binding point, at the end of the 80 nM injection of IL-6 to minimize any influence of reference binding in the pool comparison. Sensorgrams shown represent the first of 3-4 replicates collected with the enriched polymer libraries and IL-6 shown in Fig. 2c.



Supplementary Figure 10. Analysis of mixed samples in polymer pool competition. Stacked bar plot showing the distribution of sequences from the starting and eluted material from the head-to-head competition experiments in which a round of selection against bead-bound IL-6 was performed. Sequences are assigned to a pool of origin based upon the occurrence of the same sequence in any of the 7 previous selection rounds. Sequences that were not previously found from HTS analysis of previous rounds are labeled as unidentified. Sequences appearing in multiple pools where assignment was ambiguous were removed, but labelled as duplicates, so bars add to 100%.



Supplementary Figure 11. Individual polymer enrichment in IL-6 competition experiment. Each point represents a sequence that could be assigned to a pool and was identified in both the starting and eluted mixes.



Supplementary Figure 12. Individual polymer binding to IL-6 example data. Sample SPR sensorgrams of individual polymers in response to 8.8, 27, 80, and then 240 nM injection of IL-6. Data plotted in Fig. 4b were calculated based upon based on the response at the fourth binding point, at the end of the end the final injection of IL-6. Polymer Polar-E1 is the polymer exhibiting the greatest enrichment from any round in the IL-6 selection with the polar polymer library. Polymers Nonpolar-1, Nonpolar Plus-1, and Nonpolar Plus-2 are the 1st-, 1st-, and 2nd-most abundant sequences from their respective pools following IL-6 selection. Data presented are representative of sensorgrams used to derive data in Fig. 4c and Supplementary Table 4. For this measurement, most polymers were analyzed once. For the above sensorgrams, Nonpolar-1 and Nonpolar Plus 2 were analyzed twice, and Nonpolar Plus-1 was analyzed three times, giving similar results for each measurement.

Supplementary Note 1. Oligos used in selections, HFNAP preparation, and HTS.

Sequences used for selection of PCSK9-binding HFNAPs.

Name	Sequence
Naïve library AZ	CGA ATC AGA TTG GAC CAG YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN GAG TCC AGA TGT AGG TAG
ddC-pp1A	/5phos/GAG TCC AGA TGT AGG TAG/3ddC/
pp2Z	CGA ATC AGA TTG GAC CAG
ExtPrimerA	CTA CCT ACA TCT GGA CTC
BtBt_EPrimA	/52Bio//iSp18/CTA CCT ACA TCT GGA CTC

Sequences used for selection of IL-6-binding HFNAPs.

Name	Sequence
Naïve library BW	CTC GGA TGA ACC TGG ACT YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN YNN GCA TCG AAG CCA AGA TTC
ddC-pp1B	/5Phos/GCATCGAAGCCAAGATT/3ddC/
pp2W	CTC GGA TGA ACC TGG ACT
ExtPrimerB	GAA TCT TGG CTT CGA TGC
BtBt_EPrimerB	/52-Bio//iSp18/GAA TCT TGG CTT CGA TGC

Adapter primers used for Illumina MiSeq analysis of PCSK9 selections.

Name	Sequence
Seq_Adapt_A1	ACA CTC TTT CCC TAC ACG ACG CTC TTC CGA TCT N CTA CCT ACA TCT GGA CTC
Seq_Adapt_A2	ACA CTC TTT CCC TAC ACG ACG CTC TTC CGA TCT NN CTA CCT ACA TCT GGA CTC
Seq_Adapt_A3	ACA CTC TTT CCC TAC ACG ACG CTC TTC CGA TCT NNN CTA CCT ACA TCT GGA CTC
Seq_Adapt_A	ACA CTC TTT CCC TAC ACG ACG CTC TTC CGA TCT NNNN CTA CCT ACA TCT GGA CTC
Seq_Adapt_Z1	TGG AGT TCA GAC GTG TGC TCT TCC GAT CT N CGA ATC AGA TTG GAC CAG
Seq_Adapt_Z2	TGG AGT TCA GAC GTG TGC TCT TCC GAT CT NN CGA ATC AGA TTG GAC CAG
Seq_Adapt_Z3	TGG AGT TCA GAC GTG TGC TCT TCC GAT CT NNN CGA ATC AGA TTG GAC CAG
Seq_Adapt_Z	TGG AGT TCA GAC GTG TGC TCT TCC GAT CT NNNN CGA ATC AGA TTG GAC CAG

Adapter primers used for Illumina MiSeq analysis of IL-6 selections.

Name	Sequence
Seq_Adapt_B1	ACA CTC TTT CCC TAC ACG ACG CTC TTC CGA TCT N GAA TCT TGG CTT CGA TGC
Seq_Adapt_B2	ACA CTC TTT CCC TAC ACG ACG CTC TTC CGA TCT NN GAA TCT TGG CTT CGA TGC
Seq_Adapt_B3	ACA CTC TTT CCC TAC ACG ACG CTC TTC CGA TCT NNN GAA TCT TGG CTT CGA TGC
Seq_Adapt_B4	ACA CTC TTT CCC TAC ACG ACG CTC TTC CGA TCT NNNN GAA TCT TGG CTT CGA TGC
Seq_Adapt_W1	TGG AGT TCA GAC GTG TGC TCT TCC GAT CT N CTC GGA TGA ACC TGG ACT
Seq_Adapt_W2	TGG AGT TCA GAC GTG TGC TCT TCC GAT CT NN CTC GGA TGA ACC TGG ACT
Seq_Adapt_W3	TGG AGT TCA GAC GTG TGC TCT TCC GAT CT NNN CTC GGA TGA ACC TGG ACT
Seq_Adapt_W4	TGG AGT TCA GAC GTG TGC TCT TCC GAT CT NNNN CTC GGA TGA ACC TGG ACT

Barcoding adapters used in Illumina MiSeq analysis.

Name	Sequence
PE-REV1	CAAGCAGAAGACGGCATACGAGAT ATCGTGAT GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV2	CAAGCAGAAGACGGCATACGAGAT ATACATCG GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV3	CAAGCAGAAGACGGCATACGAGAT ATGCCTAA GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV4	CAAGCAGAAGACGGCATACGAGAT ATTGGTCA GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV5	CAAGCAGAAGACGGCATACGAGAT ATCACTGT GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV6	CAAGCAGAAGACGGCATACGAGAT ATATTGGC GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV7	CAAGCAGAAGACGGCATACGAGAT ATGATCTG GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV8	CAAGCAGAAGACGGCATACGAGAT ATTCAAGT GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV9	CAAGCAGAAGACGGCATACGAGAT ATCTGATC GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV10	CAAGCAGAAGACGGCATACGAGAT ATAAGCTA GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV11	CAAGCAGAAGACGGCATACGAGAT ATGTAGCC GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV12	CAAGCAGAAGACGGCATACGAGAT ATTACAAG GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV13	CAAGCAGAAGACGGCATACGAGAT TGTTGACT GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV14	CAAGCAGAAGACGGCATACGAGAT ACGGAACT GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV15	CAAGCAGAAGACGGCATACGAGAT TCTGACAT GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV16	CAAGCAGAAGACGGCATACGAGAT CGGGACGG GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV18	CAAGCAGAAGACGGCATACGAGAT GTGCGGAC GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV19	CAAGCAGAAGACGGCATACGAGAT CGTTTCAC GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV20	CAAGCAGAAGACGGCATACGAGAT AAGGCCAC GTGACTGGAGTTCAGACGTGTGCTCTTC

PE-REV21	CAAGCAGAAGACGGCATACGAGAT TCCGAAAC GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV22	CAAGCAGAAGACGGCATACGAGAT TACGTACG GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV23	CAAGCAGAAGACGGCATACGAGAT ATCCACTC GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV25	CAAGCAGAAGACGGCATACGAGAT AAAGGAAT GTGACTGGAGTTCAGACGTGTGCTCTTC
PE-REV27	CAAGCAGAAGACGGCATACGAGAT ATATCAGT GTGACTGGAGTTCAGACGTGTGCTCTTC
PE_REV-18B	CAAGCAGAAGACGGCATACGAGAT GCGTACGT GTGACTGGAGTTCAGACGTGTGCT
PE_REV-19B	CAAGCAGAAGACGGCATACGAGAT TTTCACCG GTGACTGGAGTTCAGACGTGTGCT
PE_REV-20B	CAAGCAGAAGACGGCATACGAGAT GGCCACAA GTGACTGGAGTTCAGACGTGTGCT
PE_REV-21B	CAAGCAGAAGACGGCATACGAGAT CGAAACTC GTGACTGGAGTTCAGACGTGTGCT
PE_REV-22B	CAAGCAGAAGACGGCATACGAGAT CGTACGTA GTGACTGGAGTTCAGACGTGTGCT
PE_REV-23B	CAAGCAGAAGACGGCATACGAGAT CCACTCAT GTGACTGGAGTTCAGACGTGTGCT
PE_REV-25B	CAAGCAGAAGACGGCATACGAGAT AGGAATAA GTGACTGGAGTTCAGACGTGTGCT
PE_REV-27B	CAAGCAGAAGACGGCATACGAGAT ATCAGTAT GTGACTGGAGTTCAGACGTGTGCT
FWD-2a	AATGATACGGCGACCACCGAGATCTACAC ATTACTCG ACACTCTTTCCCTACACGAC
FWD-2b	AATGATACGGCGACCACCGAGATCTACAC TCCGGAGA ACACTCTTTCCCTACACGAC
FWD-2c	AATGATACGGCGACCACCGAGATCTACAC CGCTCATT ACACTCTTTCCCTACACGAC
FWD-2d	AATGATACGGCGACCACCGAGATCTACAC GAGATTCC ACACTCTTTCCCTACACGAC
FWD-2e	AATGATACGGCGACCACCGAGATCTACAC ATTCAGAA ACACTCTTTCCCTACACGAC
FWD-2f	AATGATACGGCGACCACCGAGATCTACAC GAATTCGT ACACTCTTTCCCTACACGAC
FWD-2g	AATGATACGGCGACCACCGAGATCTACAC CTGAAGCT ACACTCTTTCCCTACACGAC
FWD-2h	AATGATACGGCGACCACCGAGATCTACAC TAATGCGC ACACTCTTTCCCTACACGAC
FWD-2i	AATGATACGGCGACCACCGAGATCTACAC CGGCTATG ACACTCTTTCCCTACACGAC
FWD-2j	AATGATACGGCGACCACCGAGATCTACAC TCCGCGAA ACACTCTTTCCCTACACGAC
FWD-2k	AATGATACGGCGACCACCGAGATCTACAC TCTCGCGC ACACTCTTTCCCTACACGAC
FWD-2l	AATGATACGGCGACCACCGAGATCTACAC AGCGATAG ACACTCTTTCCCTACACGAC

Sequences for SPR analysis derived from the unfunctionalized pool of PCSK9 selection.

Template Name	Template
Rnd12-1_seq1	CGA ATC AGA TTG GAC CAG TTA CCG CCC TCT CCC TTT TGT CCC CTC CGT TGT TTC TCT TTC CCT GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
Rnd12-1_seq2	CGA ATC AGA TTG GAC CAG CCG CTT TTG CCT TCT TAC CCT TTT TGT CAT CCC TGT TGT TTC CCT GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
Rnd12-1_seq3	CGA ATC AGA TTG GAC CAG CAC CCT CCG TCT CCG CTC CCC TCG TCT TGT TGT TTC CCT TTC CCT GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
PCSK9-MOB-1_seq1	CGA ATC AGA TTG GAC CAG CAC TTC TCA TTA TTG TTC TTA TGA CCT TGC CCG TTT TCA CCC CCT GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
PCSK9-MOB-1_seq2	CGA ATC AGA TTG GAC CAG CGC TTA CCA TTG TCT TGT TTG CCT TCC CCT TTG TCC TGT TGT TTA GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
PCSK9-MOB-1_seq3	CGA ATC AGA TTG GAC CAG CGC TTT CTC TTT TTC CCT TTT CCA TGC CCA TGT TTA TCT TTC CGC GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
PCSK9-MOB-1_seq4	CGA ATC AGA TTG GAC CAG CTT TCT TGT CGC CGC CCT CCT TGT TGT TGA TGT CCT TCC TTT CCT GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
PCSK9-MOB-1_seq5	CGA ATC AGA TTG GAC CAG TGC TTT TTC CTC TCT TAT CAC TTC TTG TTG CTT TGT TGT TTC TCT GAG TCC AGA TGT AGG TAG /iSp18//3Bio/

Templates used to synthesize HFNAPs from the charged pool derived from PCSK9 selection for SPR analysis.

Template Name	Template
Rnd12-2_seq1-temp	CTA CCT ACA TCT GGA CTC AGG ACA ACG TGG GGG TGA AGG GGG ACA ACA AGG GTG GCA ACA AAA CTG GTC CAA TCT GAT TCG
Rnd12-2_seq2	CGA ATC AGA TTG GAC CAG TTT CAT CCC TTT TTG CCA CCC CTT TGT TGT CCA CCC CCA TGT CCT GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
Rnd12-2_seq3	CGA ATC AGA TTG GAC CAG TTT CTC CCT CTT CCA CAC CTT TGT CTT TGC CAC CCT CCT TGT CCT GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
Rnd12-2_seq4-temp	CTA CCT ACA TCT GGA CTC CGG TGG AGG GTG CGG TTG GGA GAG TGA AAG ACG TTA GCA AAA TCA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-2_seq1	CGA ATC AGA TTG GAC CAG CAC TAG CCT TTT TGC CCC TTT CTT TGT CGT CTT CCC TGT TTT CTC GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
PCSK9-MOB-2_seq2	CGA ATC AGA TTG GAC CAG CGT TTT TTG CTT CTT CTA TGC CCT TTT CTC TGC CCG TTT CCC CGC GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
PCSK9-MOB-2_temp3	CTA CCT ACA TCT GGA CTC GAG ACA AAA ACA AGG GTG ACA AGG GGA AAA ACG AGG GGG CAA CTA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-2_temp4	CTA CCT ACA TCT GGA CTC AGG GGG TGA AAA GGG GCA CGG AAG GGG TGA AGA ACA AAG AAA AAG CTG GTC CAA TCT GAT TCG
PCSK9-MOB-2_temp5	CTA CCT ACA TCT GGA CTC AGG AGG ACA AAA ACA TGG AAA CCG GGG ATA AAG CGG GGA AAA CGG CTG GTC CAA TCT GAT TCG

Templates used to synthesize HFNAPs from the polar pool derived from PCSK9 selection for SPR analysis.

Template Name	Template
Rnd12-3_seq1-temp	CTA CCT ACA TCT GGA CTC AGG ACA ACA CGG GGA AGG AGG CGG GAG TGG AGG CGA GGG CAA CAA CTG GTC CAA TCT GAT TCG
Rnd12-3_seq2-temp	CTA CCT ACA TCT GGA CTC AGG ACA ACA TGG CAA TAG GGA GAG GCA GGG GGA AGG AAG AAA GGG CTG GTC CAA TCT GAT TCG
Rnd12-3_seq3-temp	CTA CCT ACA TCT GGA CTC AGG GGG ACA ACA AGG AAA GAG GGA AAG AGG CGG TGG TGG CGG AAA CTG GTC CAA TCT GAT TCG
Rnd12-3_seq4	CGA ATC AGA TTG GAC CAG TTT TTT CCT TGT CAG CCT CTC TTG CCC CCG CCT CCT TGT TGT CCT GAG TCC AGA TGT AGG TAG /iSp18//3Bio/
PCSK9-MOB-3_temp1	CTA CCT ACA TCT GGA CTC CTA ACG AGA ACA AGG GGA AAA CGG GAG ATA CCG GGA GGA AAA AGG CTG GTC CAA TCT GAT TCG
PCSK9-MOB-3_temp2	CTA CCT ACA TCT GGA CTC AGG ACA GGG AAA AAG ACA ATA AAA CGG TGG GAG AAA AGG GGA AAA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-3_temp3	CTA CCT ACA TCT GGA CTC CTA ACG AGA ACA AAA AAG ACA GGA TCA GAG AAG AAA TGG GTA CCG CTG GTC CAA TCT GAT TCG
PCSK9-MOB-3_temp4	CTA CCT ACA TCT GGA CTC CTA ACG AGA ACA AGG GGG AAA ACA AGG AAA ATG TGA AGA ATA CTG CTG GTC CAA TCT GAT TCG
PCSK9-MOB-3_temp5	CTA CCT ACA TCT GGA CTC GTA GAG AGG AAG AAA ACA GGG GGA AAA ACG AGG GTA GTG CCG TCA CTG GTC CAA TCT GAT TCG

Templates used to synthesize HFNAPs from the polar plus pool derived from PCSK9 selection for SPR analysis.

Template Name	Template
Rnd12-4_seq1-temp	CTA CCT ACA TCT GGA CTC TAG CAA GGG AGG AAG AAG AAA AGG CAA AAA AGG AAA AAA GGG TGG CTG GTC CAA TCT GAT TCG
Rnd12-4_seq2-temp	CTA CCT ACA TCT GGA CTC TAG TGG AAA GGG ACA GGG ACA AAA GGG GAG AAA AGG AAA TAG CGA CTG GTC CAA TCT GAT TCG
Rnd12-4_seq3-temp	CTA CCT ACA TCT GGA CTC AGG GTG GCG GGG CAA AGG CAA ACG AGG GCA GAG AAA ACA AAA GGG CTG GTC CAA TCT GAT TCG
PCSK9-MOB-4_temp1	CTA CCT ACA TCT GGA CTC AGG AAA AAG GGG GAG AAA AGG GCA AAA AGG CGG AAA AAA GTA CGA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-4_temp2	CTA CCT ACA TCT GGA CTC TGG CGG AAG AGG CAA GGA AGG GTG AAA GGG CAA AAA ACA AAA GCG CTG GTC CAA TCT GAT TCG
PCSK9-MOB-4_temp3	CTA CCT ACA TCT GGA CTC AGG GAG AAG AAA CAA AGG ACA AAA GGG GGG GCG CGA CGA AGG TGA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-4_temp4	CTA CCT ACA TCT GGA CTC GCG GGG CAA ACA AGG AAA AGG CAA AAA ACA GGG AAA AAA TAG AAG CTG GTC CAA TCT GAT TCG
PCSK9-MOB-4_temp5	CTA CCT ACA TCT GGA CTC AGG ACA TGG GTG AAA GGG AAG AAA GTG GGA AAA AAA CAA CAG AGG CTG GTC CAA TCT GAT TCG

Templates used to synthesize HFNAPs from the nonpolar pool derived from PCSK9 selection for SPR analysis.

Template Name	Template
12-5_seq1-temp	CTA CCT ACA TCT GGA CTC CGG AGG TGA ACA ACG ATG CGA GAG GCA AGA TAG ATA TCG AAA CTA CTG GTC CAA TCT GAT TCG
Rnd12-5_seq2-temp	CTA CCT ACA TCT GGA CTC TAG AGA ACA GAA GGA ACA GAA TGG AAA AGA AAG TGA GAA ACG AAA CTG GTC CAA TCT GAT TCG
Rnd12-5_seq3-temp	CTA CCT ACA TCT GGA CTC CGG AGG TGA ACA ACG ATG CGA GAG GCA AGA TAG GTA TCG AAA CTA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-5_temp1	CTA CCT ACA TCT GGA CTC CGG AGG TGA ACA ACG ATG CGA GAG GCA GGA TAG ATA TCG AAA CTA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-5_temp3	CTA CCT ACA TCT GGA CTC CGG AGG TGA ACA ACG ATG CGA GAG GCA AGA TAG ATA TCG TAA CTA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-5_temp4	CTA CCT ACA TCT GGA CTC CGA TGG GTA TAG CAA AGA ACA GAA GGG AAA AGA AAG CAA ACG AAA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-5_temp5	CTA CCT ACA TCT GGA CTC CGG AGG GGA ACA ACG ATG CGA GAG GCA AGA TAG ATA TCG AAA CTA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-5_temp7	CTA CCT ACA TCT GGA CTC CGG AGG TGA ACA ACG ATG CGA GAG GCA AGA TAG ATA TCA AAA CTA CTG GTC CAA TCT GAT TCG

Templates used to synthesize HFNAPs from the nonpolar plus pool derived from PCSK9 selection for SPR analysis.

Template Name	Template
Rnd12-6_seq1-temp	CTA CCT ACA TCT GGA CTC GGA CGG GGG ATA GGG GAA GGA CAG GGG AGG GCA ATG CAA ACG CTG CTG GTC CAA TCT GAT TCG
Rnd12-6_seq2-temp	CTA CCT ACA TCT GGA CTC AGG GAA CAA AGG GGA ACA GGA AAA GGG GGA ACA ACA AGA GAG GGG CTG GTC CAA TCT GAT TCG

Rnd12-6_seq3-temp	CTA CCT ACA TCT GGA CTC TTA ACA GGA ACA CAG GGG GAA AGA TAA GGG GGA CAA ACA GAG GTA CTG GTC CAA TCT GAT TCG
Rnd12-6_seq4-temp	CTA CCT ACA TCT GGA CTC TCA AGG GAA ACA AGG GGG CGA GGA GAG GAA ACA ACA AAA GGG CAA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-6_temp1	CTA CCT ACA TCT GGA CTC GCA AGG GAA ACA AGG GGG CGA GGA GAG GAA ACA ACA AAA GGG CAA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-6_temp2	CTA CCT ACA TCT GGA CTC GAG ATA ACA CAG GCG TGG GCG GTA AGG GGA AAA GGG CAG TCG CGG CTG GTC CAA TCT GAT TCG
PCSK9-MOB-6_temp3	CTA CCT ACA TCT GGA CTC GGA GGG GGA TAA ACA GGA AGA CAA GGA AGG GAA ACA CAA CAG AAA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-6_temp4	CTA CCT ACA TCT GGA CTC TAG TGA GGG TAA ACA GAA GAA AAA CGG GGG AGG AAA AAG AAA GCA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-6_temp5	CTA CCT ACA TCT GGA CTC TAA GGG GGA AGA GGG GGG AAG ACA AAA AGG GAA ACA TGA AGA AAG CTG GTC CAA TCT GAT TCG

Templates used to synthesize HFNAPs from the fully functionalized pool derived from PCSK9 selection for SPR analysis.

Template Name	Template
Rnd12-7_seq1	CTA CCT ACA TCT GGA CTC TGG AGG GTA GAG CAG GCA GAG AGG GAG GAG AGG GCA CAA CAA ACG CTG GTC CAA TCT GAT TCG
Rnd12-7-seq2	CTA CCT ACA TCT GGA CTC TGG AGG GTA GAG CAG GCA GAG CGG GAG GAG AGG GCA CAA CAA ACG CTG GTC CAA TCT GAT TCG
Rnd12-7-seq3	CTA CCT ACA TCT GGA CTC ACA ATA AGG GGG ACG TGG GGG AAA ACA GGG AAG AAA AAA CGG AGG CTG GTC CAA TCT GAT TCG
Rnd12-7-seq4	CTA CCT ACA TCT GGA CTC AGG GGG GAG ACA ACA GGA GGG AAA ACA GGG GGA TGG ATA ACA AAA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-7_temp1	CTA CCT ACA TCT GGA CTC TAA CAA AGG GGG AAG ACA GGG AAG AAA ACA GGG AGG GGA AAG CGA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-7_temp2	CTA CCT ACA TCT GGA CTC GAG AAG GGG AGG GAG AGG CAA GGG GGA AAG GCA AAA CGA CAA AAA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-7_temp3	CTA CCT ACA TCT GGA CTC TGG AAA ATA AGG GCA ATG GGA AAA ACA ACA AAG AAA AGG GGG CGA CTG GTC CAA TCT GAT TCG
PCSK9-MOB-7_temp4	CTA CCT ACA TCT GGA CTC TGA AAG GGG ACA AGG ACA AGG GAG AGG AAA ACA GGG AAG GCA AAG CTG GTC CAA TCT GAT TCG
PCSK9-MOB-7_temp5	CTA CCT ACA TCT GGA CTC TGG TCA AAG GGG CAG GGG CAA ACA AGG GCA GGG AAA AAG CGG AAA CTG GTC CAA TCT GAT TCG

Sequences for SPR analysis derived from the unfunctionalized pool of IL-6 selection.

Template Name	Template
IL6-R7-1_seq1	CTC GGA TGA ACC TGG ACT TTA TTG CAC TGG CTG TAC TTT TCG TAT CCA TGG TCA TCT TTT CTA GCA TCG AAG CCA AGA TTC /iSp18//3Bio/
IL6-R7-1_seq2	CTC GGA TGA ACC TGG ACT TTA CTG CGG CGC CAG CCG TGC CCA TCC TTT CTT CAT CCG TGC CCA GCA TCG AAG CCA AGA TTC /iSp18//3Bio/
IL6-R7-1_seq3	CTC GGA TGA ACC TGG ACT CTG TTT TTA CTA CCG CGC TAG CCG CCT CAT CCG TTT TAG CGC CAT GCA TCG AAG CCA AGA TTC /iSp18//3Bio/
IL6-R7-1_seq4	CTC GGA TGA ACC TGG ACT TAA TGA TGC TCA TGG TGC TCT TCA TCA CCA TCT TCG TTC CAT CCT GCA TCG AAG CCA AGA TTC /iSp18//3Bio/
IL6-R7-1_seq5	CTC GGA TGA ACC TGG ACT TCT TCA TTG CGC TGG TGC CTG CCA TCC TTG TTC CAT CCC CTT TGC GCA TCG AAG CCA AGA TTC /iSp18//3Bio/
IL6-MOB-1_seq1	CTC GGA TGA ACC TGG ACT TTA TTG CAC TGG CTG TAC TTT TCG TAT CCA TGG TCA TCT TTC CTA GCA TCG AAG CCA AGA TTC /iSp18//3Bio/
IL6-MOB-1_seq2	CTC GGA TGA ACC TGG ACT TTA CTG CGG CGC CAG CCG TGC CCA TCC TTT CTC CAT CCG TGC CCA GCA TCG AAG CCA AGA TTC /iSp18//3Bio/
IL6-MOB-1_seq3	CTC GGA TGA ACC TGG ACT TTA TTG CAC TGG CTG TAC TTT TCG TAT CCA TGG TCA TCC TTT CTA GCA TCG AAG CCA AGA TTC /iSp18//3Bio/
IL6-MOB-1_seq4	CTC GGA TGA ACC TGG ACT TTA TTG CAC TGG CTG TAC TTC TCG TAT CCA TGG TCA TCT TTT CTA GCA TCG AAG CCA AGA TTC /iSp18//3Bio/
IL6-MOB-1_seq5	CTC GGA TGA ACC TGG ACT TTA TTG CAC TGG CTG TAC TTT TCG TAT CCA TGG TCA TCT TTT CTG GCA TCG AAG CCA AGA TTC /iSp18//3Bio/

Templates used to synthesize HFNAPs from the charged pool derived from IL-6 selection for SPR analysis.

Template Name	Template
IL6-R7-2_seq1-temp	GAA TCT TGG CTT CGA TGC CCG GTG GAA GCA ACA ACA CTA ACG CCA GGA CCA TAA GCA TTG CAG AGT CCA GGT TCA TCC GAG
IL6-R7-2_seq2-temp	GAA TCT TGG CTT CGA TGC GAG CCG CTA GCG CGG TAA TAG AAA TCG GGA GCA CGA CCA TCG CGA AGT CCA GGT TCA TCC GAG
IL6-R7-2_seq3-temp	GAA TCT TGG CTT CGA TGC CCG GTG GAA GCA ACA ACA CTA CCG CCA GGA CCA TAA GCA TTG CAG AGT CCA GGT TCA TCC GAG

IL6-R7-2_seq4-temp	GAA TCT TGG CTT CGA TGC ACG AAG AGG GCG GGG TGA CAA GCG CAG GCA GCA TAG AAG CCA CTA AGT CCA GGT TCA TCC GAG
IL6-MOB-2_temp2	GAA TCT TGG CTT CGA TGC GAG CCG CTA GCG CGG TAA TAG AAA TCG GGA CCA CGA CCA TCG CGA AGT CCA GGT TCA TCC GAG
IL6-MOB-2_temp3	GAA TCT TGG CTT CGA TGC GAG CCG CTA GCG CGG TAA TAG AAA TCG GGA GTA CGA CCA TCG CGA AGT CCA GGT TCA TCC GAG
IL6-MOB-2_temp5	GAA TCT TGG CTT CGA TGC CCG GTG GAA GCA ACA ACA CTA ACG CCA GGA CCA TAA TCA TTG CAG AGT CCA GGT TCA TCC GAG
IL6-MOB-2_temp6	GAA TCT TGG CTT CGA TGC GAG CCG CTA GCG CGG TAA TAG AAA TCG GGG GCA CGA CCA TCG CGA AGT CCA GGT TCA TCC GAG
IL6-MOB-2_temp7	GAA TCT TGG CTT CGA TGC GAG CCG CTA GCG CGG TAA TAG AAA TCG GGA TCA CGA CCA TCG CGA AGT CCA GGT TCA TCC GAG

Templates used to synthesize HFNAPs from the polar pool derived from IL-6 selection for SPR analysis.

Template Name	Template
IL6-R7-3_seq1-temp	GAA TCT TGG CTT CGA TGC GGG CGG GCA CCG AGA TCA TGA GAG GTA ACA ACA TTA AGG GGG CCA AGT CCA GGT TCA TCC GAG
IL6-R7-3_seq2-temp	GAA TCT TGG CTT CGA TGC GGA TGG GGG CAA GGG AAG GAA CAG TAA GAA CGA AAG AAA TTG CCG AGT CCA GGT TCA TCC GAG
IL6-R7-3_seq3-temp	GAA TCT TGG CTT CGA TGC CGG AGG ATG GAA CAG TAG GAA CGA AGG ACA TCA GAG CCA ACA TGA AGT CCA GGT TCA TCC GAG
IL6-MOB-3_temp1	GAA TCT TGG CTT CGA TGC GGA TGG GGG CAA GGG AAG GAA CAG TAA GAA CGA GAG AAA TTG CCG AGT CCA GGT TCA TCC GAG
IL6-MOB-3_temp2	GAA TCT TGG CTT CGA TGC CGG AGG GTG GAA CAG TAG GAA CGA AGG ACA TCA GAG CCA ACA TGA AGT CCA GGT TCA TCC GAG
IL6-MOB-3_temp4	GAA TCT TGG CTT CGA TGC GGG CGG GCA CCG AGA TCA TGA GAG GTA ACA ACA TTA GGG GGG CCA AGT CCA GGT TCA TCC GAG
IL6-MOB-3_temp5	GAA TCT TGG CTT CGA TGC CGG AGG ATG GAA CAG TAG GAA CGA AGG ACA TCA GAG CCA ACA AGA AGT CCA GGT TCA TCC GAG
IL6-MOB-3_temp7	GAA TCT TGG CTT CGA TGC CGG AGG ATG GAA CAG TAG GAA CGA AGG GCA TCA GAG CCA ACA TGA AGT CCA GGT TCA TCC GAG

Templates used to synthesize HFNAPs from the polar plus pool derived from IL-6 selection for SPR analysis.

Template Name	Template
IL6-R7-4_seq1-temp	GAA TCT TGG CTT CGA TGC CAG TCG ACA TCA TGA ACG GGG ATG ACA GCA CAA TGG CAA TCG AAA AGT CCA GGT TCA TCC GAG
IL6-R7-4_seq2-temp	GAA TCT TGG CTT CGA TGC CAG GGG ATG AGA TCA CAA GCA AAA ATA CAG CAA CAG TTA AGG GAA AGT CCA GGT TCA TCC GAG
IL6-R7-4_seq3-temp	GAA TCT TGG CTT CGA TGC ACG GGG GAA AGG GTG GAA CAG TAA GAA CGA AAA AAG CCG TGA TGG AGT CCA GGT TCA TCC GAG
IL6-MOB-4_temp1	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCG CCA CAG CAA AGA TCA CCA GTA TTA TGA TAA AGG AGT CCA GGT TCA TCC GAG
IL6-MOB-4_temp2	GAA TCT TGG CTT CGA TGC ACA GTG GAA ACA CAG TCG GGA GGA CCG TAG TCG ACA CAG TTA TGG AGT CCA GGT TCA TCC GAG
IL6-MOB-4_temp3	GAA TCT TGG CTT CGA TGC CAG TCG ACA TCA TGA GCG GGG ATG ACA GCA CAA TGG CAA TCG AAA AGT CCA GGT TCA TCC GAG
IL6-MOB-4_temp4	GAA TCT TGG CTT CGA TGC GCG ACG GAG AAG GAA CAG TAA GAA CGA AAA CAA AAG CAG CAA CCG AGT CCA GGT TCA TCC GAG
IL6-MOB-4_temp7	GAA TCT TGG CTT CGA TGC GGG TGA GGG GTG AGG CAG AGA TGG AGA TTG AAG CCA ATA CAG GGA AGT CCA GGT TCA TCC GAG

Templates used to synthesize HFNAPs from the nonpolar pool derived from IL-6 selection for SPR analysis.

Template Name	Template
IL6-R7-5_seq1-temp	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCG CCA CAG CAA AGA TCA CCA GTA TTA TGA TAA AGG AGT CCA GGT TCA TCC GAG
IL6-R7-5_seq2-temp	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCG CCA CAG CCA ACA TAA ACG CAA AGG GTA TAA GTA AGT CCA GGT TCA TCC GAG
IL6-R7-5_seq3-temp	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCG CCA CAG CCA TAG ATA GGA CCA AAA CCA TGG GAG AGT CCA GGT TCA TCC GAG
IL6-MOB-5_temp1	GAA TCT TGG CTT CGA TGC CAG TAG TAA ACA AAA GCA CAG ATA TCG CCA GCA AAG AGA ATA GAA AGT CCA GGT TCA TCC GAG
IL6-MOB-5_temp2*	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCG CCA CAA CCA AGG GAA CAA GGA AGA ACA CTA AAA AGT CCA GGT TCA TCC GAG
IL6-MOB-5_temp4	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCG CCA CAG CCA ACA TAA ACG CAA AGG GTA CAA GAA AGT CCA GGT TCA TCC GAG
IL6-MOB-5_temp5	GAA TCT TGG CTT CGA TGC TCG GGA GAG TGA GGG GCA CGG GGA GAG GGA CAA CGA ATA CCA CAG AGT CCA GGT TCA TCC GAG
IL6-MOB-5_temp6	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCG CCA CAG CAA AGA TCA CCA GCA TTA TGA CAA AGG AGT CCA GGT TCA TCC GAG

* Translation did not yield sufficient full-length product

Templates used to synthesize HFNAPs from the nonpolar plus pool derived from IL-6 selection for SPR analysis.

Template Name	Template
IL6-R7-6_seq1-temp	GAA TCT TGG CTT CGA TGC CCA TAG TCA TCA CTA AAA GCG CGA GAA GAG CCG TTG TGG CGG TGA AGT CCA GGT TCA TCC GAG
IL6-R7-6_seq2-temp	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCA GAA GCA GGA AGA GAA AAA CAA CAA GGG TCG GCG AGT CCA GGT TCA TCC GAG
IL6-R7-6_seq3-temp	GAA TCT TGG CTT CGA TGC GAA GCA TCA TGA AGG AGG GTA TCA GCG CGG GCG CGA CAA CCA CCA AGT CCA GGT TCA TCC GAG
IL6-R7-6_seq4-temp	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCA TCA CCA CAA GGA AGA ATA TAA GCA ATA GAA TAG AGT CCA GGT TCA TCC GAG
IL6-MOB-6_temp1	GAA TCT TGG CTT CGA TGC CCA TGG ACG ACA CCA CAA TAG CAA GTA GCA AAG AAA TCA GCG CCA AGT CCA GGT TCA TCC GAG
IL6-MOB-6_temp2	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCA GAA GCA TGA GTA TGA AGA CAA AAA AAA CGG CGG AGT CCA GGT TCA TCC GAG
IL6-MOB-6_temp3	GAA TCT TGG CTT CGA TGC GCA GCG TGG TAG ATA TGA CAG GAG AGG CCA CTG AAA GCG CCA CCA AGT CCA GGT TCA TCC GAG
IL6-MOB-6_temp5	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCA GAA GCA GGA AGA GAA AAA CAA CAA GGG GCG GCG AGT CCA GGT TCA TCC GAG
IL6-MOB-6_temp6	GAA TCT TGG CTT CGA TGC CCG GAG CAA TCA TCA CCA CAA GGA AGA ATA TAA GCA ACA GAA TAG AGT CCA GGT TCA TCC GAG

Templates used to synthesize HFNAPs from the fully functionalized pool derived from IL-6 selection for SPR analysis.

Template Name	Template
IL6-R7-7_seq1-temp	GAA TCT TGG CTT CGA TGC GCA TAG GTA CAA TGG GCG CGA GGG GCA CAG TCG AGA AGG CCG TGA AGT CCA GGT TCA TCC GAG
IL6-R7-7_seq2-temp	GAA TCT TGG CTT CGA TGC GCG GAG ACG CAA CAG CAA AAA GCA CAA CCA CAA ACA AGA ACA CGA AGT CCA GGT TCA TCC GAG
IL6-R7-7_seq3-temp	GAA TCT TGG CTT CGA TGC GCA TAG GTA CAA TGG GCG CGA GGG GCA CAG TCG AGA AGG TCG TGA AGT CCA GGT TCA TCC GAG
IL6-R7-7_seq4-temp	GAA TCT TGG CTT CGA TGC GCA TAG GTA CAA TGG GCG CGA GGG GCA CAG TCG AGA AGG CCG CGA AGT CCA GGT TCA TCC GAG
IL6-MOB-7_temp1	GAA TCT TGG CTT CGA TGC GCA TAG GTA CAA TGG GCG CGA GGG GCA CAG TCG AGA CGG CCG TGA AGT CCA GGT TCA TCC GAG
IL6-MOB-7_temp5*	GAA TCT TGG CTT CGA TGC TGG GTA AAA TGG ATG GAG TAA ACG ACG CCA AGA AGG TGA TAG AGA AGT CCA GGT TCA TCC GAG
IL6-MOB-7_temp6	GAA TCT TGG CTT CGA TGC GCA TAG GTA CAA TGG GCG CGA GGG GCA CAG TCG CGA AGG CCG TGA AGT CCA GGT TCA TCC GAG
IL6-MOB-7_temp7*	GAA TCT TGG CTT CGA TGC TGG GTA AAA TGG ATG GAG TAA ACG ACG CCA AGA AGG AGA TAG AGA AGT CCA GGT TCA TCC GAG
IL6-MOB-7_temp8	GAA TCT TGG CTT CGA TGC CCA AGA GGA CCG CGG GGA TAA CCG GAA GCA TAA CCG AAA ACG CGA AGT CCA GGT TCA TCC GAG

* Translation did not yield full-length product

Supplementary Note 2. Description of pool assignment protocol for HTS sequence data from head-to-head competition. Filtered sequences of the appropriate length were obtained from HTS of the mixed samples and the eluted material. These data were used as input files containing a list of the DNA sequences, ordered by relative abundance, along with the number of reads for each sequence in the sample. Each of the unique sequences from these lists were matched first against the HTS data gathered after each of the individual pools was subjected to the final round of parallel selection (e.g., round 12 for the PCSK9 selection). Another data file was produced from this analysis containing each of the unique sequences from the head-to-head competition sample, ranked by abundance, along with a number corresponding to the pool to which it was assigned, the number of reads in the sample, and a list of the pools from which the sequence appeared. Of note, some of the most abundant sequences also appeared at low frequency in other pools, presumably because the probability of contamination increases with higher abundance. At the end of output file is a list of the total number of reads assigned to each starting pool and another list containing the fraction of the total ascribed to each pool. Also included is the percentage of sequences discarded because they appeared in relatively low abundance in several pools and the percentage of sequences that were unmatched. These unmatched sequences were also output in a separate file.

For the bulk library comparison, to make additional assignments, the unmatched sequences from the head-to-head-competition samples were then matched against a much larger list of polymers from each pool containing the entire list of sequences that had been observed in any round of selection (i.e., sequences that were also found from rounds 1-11 in the PCSK9 selection).

All code was run using Python 2.7.10.

Supplementary Note 3. Code used for pool assignment of sequences from head-to-head competition data.

```
#!/usr/bin/env python
import sys, math, collections, time
from numpy import *
from collections import Counter
from decimal import *

# to run this program, type something like python tester.py List_of_sequences output_file

try:
    infilename = sys.argv[1]        # This will input the file containing the big list of sequences to be analyzed
    infilenametwo = sys.argv[2]     # This inputs the file where the list of sequences will be pulled, i.e. the file with sequences from an individual pool
    outfilename = sys.argv[3]       # This inputs the file where the matches will go
    outfilelast = sys.argv[4]    # This is where the record will be created

except:
    print "Usage:",sys.argv[0], "1_sequence_to_analyze 2_Filelist_file 3_Output_of_matches 4_record_file"; sys.exit(1)

# Sample input:
# python assignall6_mob.py Eluted_data_seqinterp allpools_combined Eluted_data_MOBcomplete_NoDup assignmentrecordedXXMix-NoDup

ifile = open(infilename, 'r')  # open sequence list file from combined pools
ifile2 = open(infilenametwo, 'r') #opens the list of sequence files to look at
recordfile = open(outfilelast, 'a') # open a file where info will be written for a number of sequences
matchfile = open(outfilename, 'w')

unmatchedfilelist = [outfilename, 'unmatched']
unmatchedfilename = '_'.join(unmatchedfilelist)
unmatchedfile = open(unmatchedfilename, 'w')

results = ifile.readlines(); ifile.close() #opens combined pool
files = ifile2.readlines(); ifile2.close()
timestr = time.strftime("%Y-%m-%d %H:%M:%S")

matchfile.write("> %s %s: %s %s %s %s %s %s\n" % (timestr, sys.argv[0], "ln", len(results)-1, "sequences of ", infilename, "searching for matches from",
files))       # stamps the date and time on the output count file
matchfile.write("%s\n \n" % ("index, sequence, best rank, pools in which sequence appears, (rank in pool different pools)"))

matches = []

pool1 = []
pool2 = []
pool3 = []
pool4 = []
pool5 = []
pool6 = []
pool7 = []

print files
templist = {}
readlist = {} # dictionary that ranks which pool that sequence likely came from, updated over time
reads = {} # dictionary that will hold the number of the sequence reads
# Calculating the total reads in the file
totalcount = 0 # this will eventually hold the total number of sequence reads from the mix

duplicates = {} #opens a dictionary of duplicates

# results contains the sequencing reads in the following format:
# > 2016-07-07 15:12:33 processfull237.py PL237-PElute1
# 1 CGTTTGTTGTGCCCTCTCCTCCCTCTCTGCCTGCTCTACCCTCCA 226396
# 2 TTTCGTTTCTCACTTTCTTTTCCATTCTGTTCCTTCTGTTCTCTA 218770
# 3 CAGCGTTTGCATTGCCCTCCCCTGTCCTTCCCCTATCCCCCGTCC 192318
# 4 TAGTTTCGATATCTATCTTGCCTCTCGCATCGTTGTTCACCTCCG 152445
# 5 TAGTTTCGATACCTATCTTGCCTCTCGCATCGTTGTTCACCTCCG 88141

for row in results[1:]:
        row = row.split()
        sequence = row[1]  # depending on the input file, this may need to be column 2
        seq_count = int(row[2])
        totalcount = totalcount + seq_count
        reads[sequence] = seq_count
        templist[sequence] = 0
```

```
		readlist[sequence] = (0,0,0,0,0,0,0) # sequence rank in each pool, if found


print totalcount

poolsumlist = []

poolcount = 0
for pool_db in files:
		pool_db = pool_db.split()[0] # removes the /n from the line
		ifile3 = open(pool_db, 'r') # opens the pool hit file
		print pool_db
		pool_db = ifile3.readlines(); ifile3.close() # pool lines defined
		poolhits = []
		poolcount = poolcount + 1
		for line in pool_db[1:]: #CHANGE BACK FOR ALL
			entry = line.split() #should split into 0:rank, 1:interpreted sequence, 2:DNA sequence, 3: reads
			tup = (entry[0], entry[2], entry[3], pool_db) # enter data as rank, sequence, reads, pool
			poolhits.append(tup)

		checksum = 0

		for ele in poolhits:
			sequence = ele[1]
			poolrank = ele[0]
			y = 0
			tuplst = []
			if sequence in templist:
				if templist[sequence] == 0:
					templist[sequence] = poolcount
				else:
					temp = []
					temp.append(str(templist[sequence])); temp.append(str(poolcount))
					templist[sequence] = ','.join(temp[0:])
				tuplst = list(readlist[sequence])
				tuplst[poolcount - 1] = int(poolrank) + 1
				readlist[sequence] = tuple(tuplst)
			checksum = checksum + 1

i = 1
unmatched = 0
unmatchedreads = []
unmatchedsequences = []
poolreadtotal = {1:0, 2:0, 3:0, 4:0, 5:0, 6:0, 7:0, 8:0}

for row in results[1:]:
		row = row.split()
		sequence = row[1] # pulls sequence
		seq_reads = int(row[2])
		ranks = list(readlist[sequence])
		if sum(ranks) > 0:
			toprank = min(i for i in ranks if i > 0)
			if toprank >= 1000 and toprank != sum(ranks): # <-- Threshold for duplicate exclusion here
				assigned_pool = 8
			else:
				assigned_pool = ranks.index(toprank) + 1
			poolreadtotal[assigned_pool] = int(poolreadtotal[assigned_pool]) + seq_reads

#create new dictionary with key = number of reads for the pool, update the key with new sum
		else:
			assigned_pool = 0
			unmatched = unmatched + 1
			unmatchedreads.append(int(row[2]))	# adds to list containing the read number of unmatched sequences
			unmatchedsequences.append(sequence) # adds unmatched sequences to a list
		matchfile.write("%s %s %s %s %s %s\n" % (i, sequence, assigned_pool, reads[sequence], templist[sequence], readlist[sequence]))
		i = i+1

print "\n \n"
print poolreadtotal

poolpercentages = []
####  calculate the percentage of each of the read totals:

matchfile.write("###\n \n %s\n" % ("The summed reads are"))
```

```
for i in range(1, 8, 1):
        matchfile.write("%s\n" % (poolreadtotal[i]))

matchfile.write("\n %s\n" % ("The pool percentages are"))
for i in range(1, 8, 1):
        poolper = float(poolreadtotal[i])/float(totalcount)
        poolpercentages.append(poolper)
        matchfile.write("%s %s\n" % (i, poolper))
print poolpercentages

matchfile.write("\n %s\n" % ("The duplicate reads and fraction of whole are"))
matchfile.write("%s\n" % (poolreadtotal[8])) # prints duplicate reads
matchfile.write("%s\n\n" % (float(poolreadtotal[8])/float(totalcount))) # prints fraction total

matchfile.write("\n %s %s\n" % ("The percentage of total reads accounted for above is", sum(poolpercentages)))

totalunmatched = sum(unmatchedreads)
print totalunmatched
percentunmatched = 100*float(totalunmatched)/float(totalcount)

matchfile.write("\n \n %s %s %s %s %s %s %s\n \n" % ("There are", unmatched, "unidentified sequences, comprising", totalunmatched, " reads, or",
percentunmatched, "% of total"))
timestr = time.strftime("%Y-%m-%d %H:%M:%S")
matchfile.write("> %s %s" % (timestr, sys.argv[0])) # stamps the date and time on the output count file

unmatchedfile.write("> %s %s: %s %s %s %s %s %s\n" % (timestr, sys.argv[0], "In", len(results)-1, "sequences of ", infilename, "searching for matches
from", files))          # stamps the date and time on the output count file


for j in range(0, len(unmatchedsequences), 1):
        unmatchedfile.write("%s %s %s\n" % (j, unmatchedsequences[j], unmatchedreads[j]))       # stamps the date and time on the output count file

recordfile.write("%s, %s %s %s %s %s\n" % (timestr, sys.argv[0], sys.argv[1], sys.argv[2], sys.argv[3], sys.argv[4]))
```

Supplementary Note 4. Code used to calculate individual sequence enrichments of assigned polymers in head-to-head competition. To assess the enrichment of individual sequences in the head-to-head competition that had been assigned to their pools of origin, the output file generated above was amended with the following code to also output a percentage of the total sample for each sequence.

```
#!/usr/bin/env python
import sys, math, collections, time
from numpy import *
from collections import Counter

try:
    infilename = sys.argv[1]        # This will input the file containing the list of sequences to be analyzed
    outfilenametwo = sys.argv[2]  # This will input a file where all of the values will be stored

except:
    print "Usage:",sys.argv[0], "1_sequence_list 2_record_file"; sys.exit(1)

outputfilename = '_'.join([infilename, 'amend'])

ifile = open(infilename, 'r')  # open sequence list file
outputfile = open(outputfilename, 'w') # open file where amended information will be kept
recordfile = open(outfilenametwo, 'a') # open a file where record will be kept

rank = []
DNAseqlist = []
HFNAPseqlist = []
readcount = []

print ifile.readline() # removes header line and prints it in the real-time display
print ifile.readline() # removes header line and prints it in the real-time display
print ifile.readline() # removes header line and prints it in the real-time display

for line in ifile:    # scans sequence list file line by line
    datalist = line.split() # split up the entries in the file
    if datalist[0] == '###':
            break
    rank.append(datalist[0]) #adds rank list
    DNAseqlist.append(datalist[1]) # adds to growing list of DNA sequences
    HFNAPseqlist.append(datalist[2]) # adds to the growing list of HFNAP sequences
    readcount.append(int(datalist[3])) #adds reads

readtotal= sum(readcount)

timestr = time.strftime("%Y-%m-%d %H:%M:%S")
print "> %s %s %s" % (timestr, sys.argv[0], infilename) #Prints out the date and time for logging of multiple runs
print infilename

print "There are", readtotal, "total reads"
print "and", len(rank), "unique DNA sequences"

recordfile.write("%s, %s\n" % (timestr, infilename))

outputfile.write("> %s %s %s\n" % (timestr, sys.argv[0], infilename))      # stamps the date and time on the output count file
outputfile.write("> %s\n" % (readtotal))    # stamps total number of sequences
for i in range(0, len(rank), 1):
    outputfile.write("%s %s %s %s %s\n" % (rank[i], DNAseqlist[i], HFNAPseqlist[i], readcount[i], 100*float(readcount[i])/float(readtotal)))
```

The amended files containing the list of assigned sequences from samples before and after the selection were then compared against each other. Sequences that were found in both samples were compiled into a list ordered by their relative enrichment in the head-to-head competition.

```
#!/usr/bin/env python
import sys, math, collections, time
from numpy import *
from collections import Counter

# The goal of this program is to take in the combined list of sequences and to track how well each of those sequences
# perform in the head-to-head competition experiment - basically want to compare enrichment of sequences in selection round

try:
    infilename = sys.argv[1]        # This will input the file containing the big list of sequences from each pool
    infilenametwo = sys.argv[2]     # This inputs the file where the list of sequences from individual round/pool will be pulled
    outfilename = sys.argv[3]        # This inputs the file where the matches will go
```

```
    outfilelast = sys.argv[4]    # This is where the record will be created

except:
    print "Usage:",sys.argv[0], "1_combined_sequence_list 2_file_containing_list_of_all_seqinterp_files_from_pool 3_Output_of_matches 4_record_file";
sys.exit(1)

ifile = open(infilename, 'r')  # open sequence list file from combined pools
ifile2 = open(infilenametwo, 'r') #opens the list of sequence files to look at
recordfile = open(outfilelast, 'a') # open a file where info will be written for a number of sequences

fulllist = ifile.readlines(); ifile.close() #opens combined list of all sequences

subfiles = ifile2.readlines(); ifile2.close() #reads the lines of the file containing all of the sequence files from the rounds

timestr = time.strftime("%Y-%m-%d %H:%M:%S")

# Format of the data in fulllist is index, DNA sequence, DNA sequence, total counts

percentpoollist = {}
readroundlist = {}
lastranklist = {}
pooloforigin = {}

for line in fulllist[2:]: # This will be the loop that extracts all of the sequneces that are to be examined
    sequencedata = line.split()
    sequence = sequencedata[1] #extracts sequence from the big list of sequences
    percentpoollist[sequence] = (0,0) # defines tuple that will include an entry for each round for each sequence's percentages
    readroundlist[sequence] = (0,0) # defines tuple that will include an entry for each round for each sequence's total reads
    lastranklist[sequence] = 'X'
    pooloforigin[sequence] = sequencedata[2]

roundcount = 0 # defines a value for indexing the round count
for subfil in subfiles:
    subfile = subfil.split()[0] # removes the /n from the line
    ifile3 = open(subfile, 'r') # opens the data file from the round
    print "subfil is", subfil # should print the round number?
    print "subfile is", subfile
    lines = ifile3.readlines(); ifile3.close() # reads the sequence data from each of the rounds

    roundcount = roundcount + 1
    print "roundcount is", roundcount

    poolhits = [] #creates list where each sequence will go
    readcount = lines[1].split()[1] #[1]picks out total number of reads in the second line of document
    print "readcount is", readcount
#    print lines
    for line in lines[2:]:  # Loop parsing data from the amended _seqinterp files
        entry = line.split()  # should split into 0:rank, 1:DNA sequence, 2:pool origin, 3: reads, 4: percentage of total
        #       print entry[2]
        if entry[1] in percentpoollist.keys():
                    tup = (entry[0], entry[1], entry[2], entry[3], entry [4], subfile)  # enter data as rank, sequence, pool origin, reads, pool
                    poolhits.append(tup)

        checksum = 0

    for i in range(0, len(poolhits), 1):
        sequence = poolhits[i][1]
        seqorigin = poolhits[i][2]
        seqreads = poolhits[i][3]
        seqpercent = poolhits[i][4]
        if roundcount == 2:
                    lastrank = poolhits[i][0]
                    lastranklist[sequence] = lastrank

        #for each sequence in the individual round, want to create tuple that has the reads for each round
        #also want to create a tuple that has the percentage for each read

        # want to output the tuple to a list and then modify the particular entry
        temppercentlist = list(percentpoollist[sequence])
        tempreadslist = list(readroundlist[sequence])
        temppercentlist[roundcount-1] = seqpercent
        tempreadslist[roundcount-1] = seqreads

        percentpoollist[sequence] = tuple(temppercentlist)
        readroundlist[sequence] = tuple(tempreadslist)
```

```
        pooloforigin[sequence] = seqorigin


countfilename = '_'.join([outfilename, 'count'])
countfile = open(countfilename, 'w')
countfile.write("> %s %s\n" % (timestr, sys.argv[0])) #Header to output file
counter = 0

mobilityfilename = '_'.join([outfilename, 'mobility'])
mobilityfile = open(mobilityfilename, 'w')
mobilityfile.write("> %s %s\n" % (timestr, sys.argv[0])) #Header to output file

percentagefilename = '_'.join([outfilename, 'percent'])
percentfile = open(percentagefilename, 'w')
percentfile.write("> %s %s\n" % (timestr, sys.argv[0])) #Header to output file
counter = 0

newdictionary1 = {}
newdictionary2 = {}

for w, z in percentpoollist.items():
        counter += 1
        percentfile.write("%s %s %s\n" % (counter, w, z))   #
        templist = []
        for i in range(1, len(z), 1):
                if z[i-1] > 0:
                        templist.append(round(float(z[i])/float(z[i-1]),2))
                elif z[i-1] and z[i] == 0:
                        templist.append(0)
                else:
                        templist.append(-0.01)
        total = sum(templist)
        newdictionary1[w] = max(templist) # adds total
        newdictionary2[w] = templist # adds templist

counter = 0
for k, v in sorted(newdictionary1.items(), key=lambda p:p[1], reverse = True):
        counter +=1
        if v > 1:
                mobilityfile.write("%s %s %s %s %s %s\n" % (counter, pooloforigin[k], lastranklist[k], k, v, newdictionary2[k])) # prints index,
sequence, total, tuple

recordfile.write("%s, %s %s %s %s %s\n" % (timestr, sys.argv[0], sys.argv[1], sys.argv[2], sys.argv[3], sys.argv[4]))
```