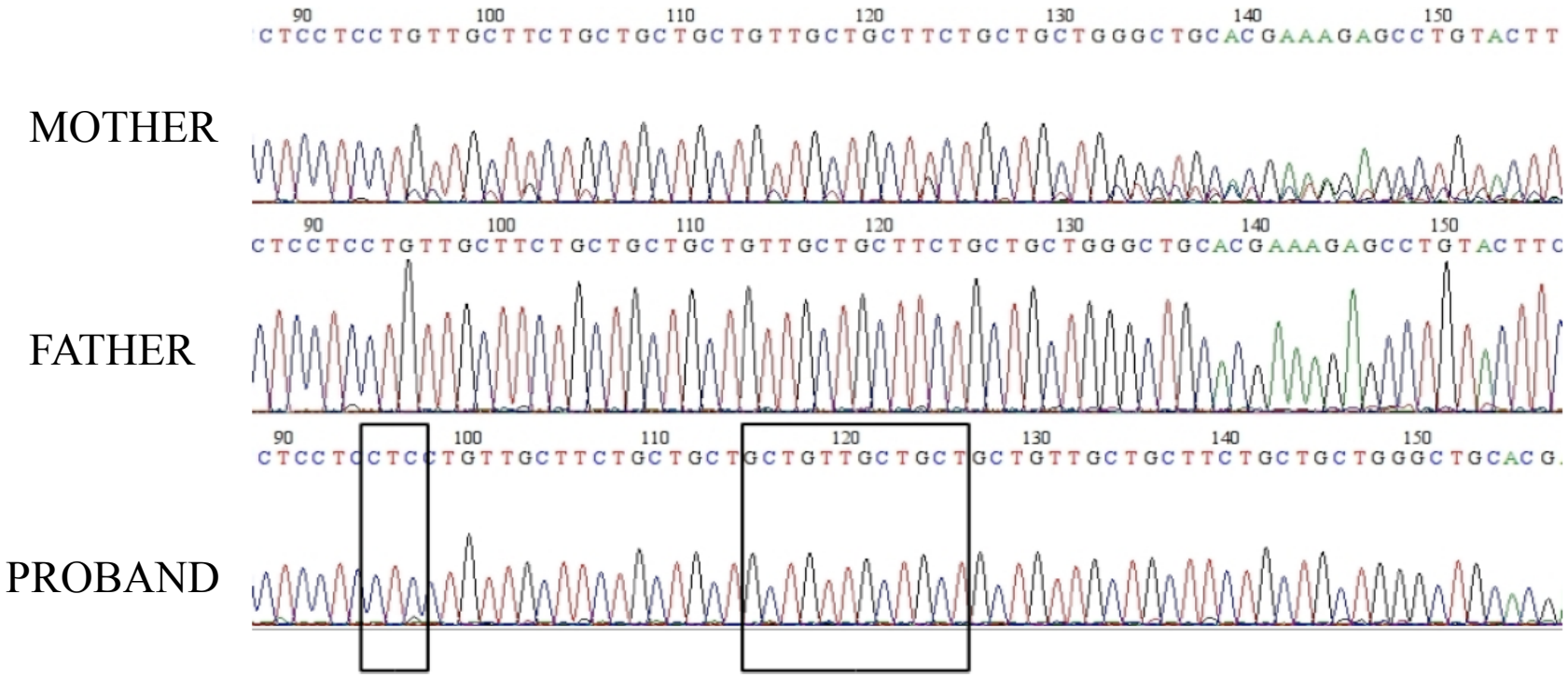
Supplementary Figure 1

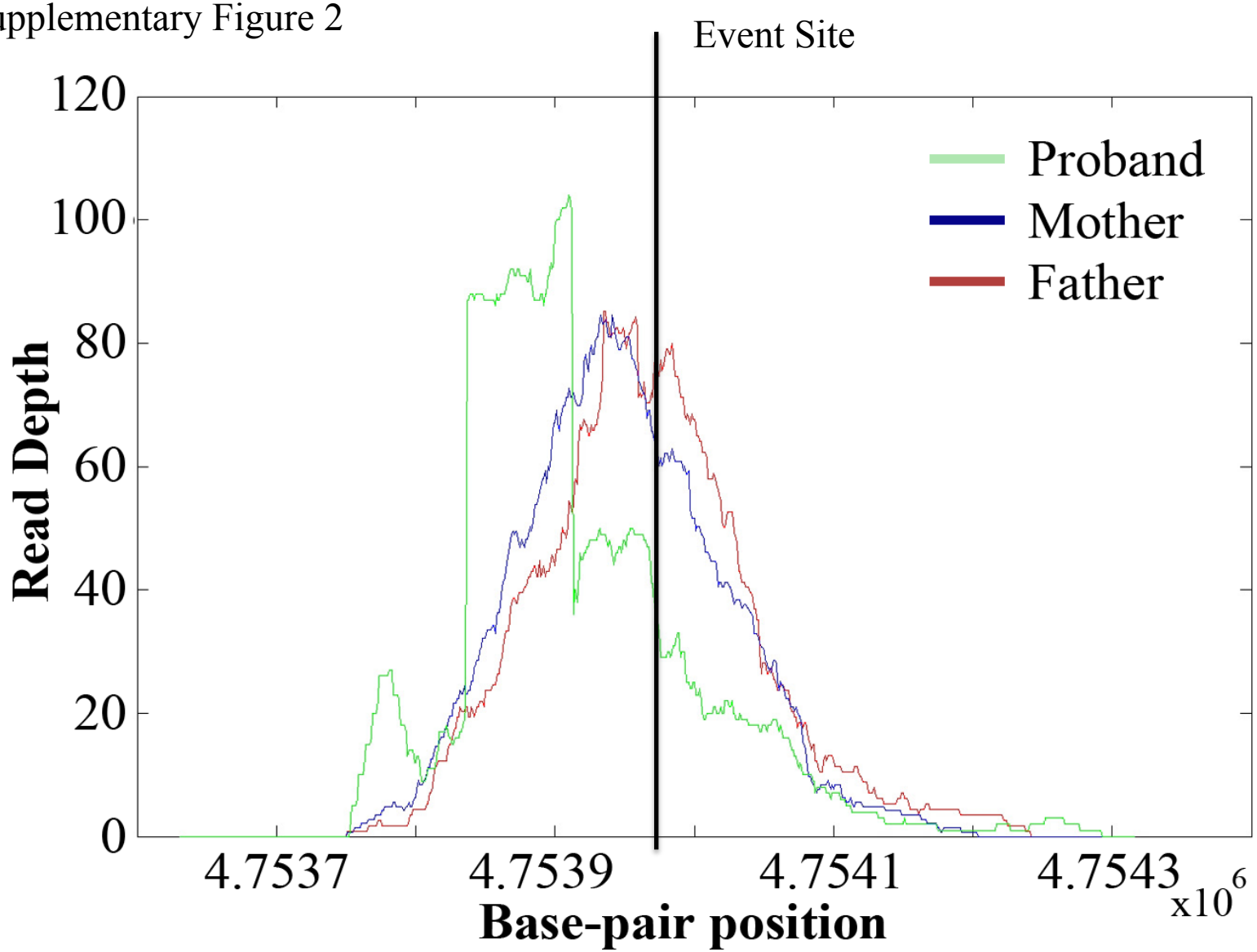
```
CLUSTAL 2.0.12 multiple sequence alignment

Protein SEQ    E  E  E  E  E  E  E  E  E  E  E  Q  Q  K  Q  Q  Q  Q  Q  Q  Q  Q  Q  K  Q  Q  Q  A  A  R  F  L  R  Y  K  E
991555(FA)     TTCTTCTTCTTCCTCCTCCTCCTCCTCCTCCT---GTTGCTTCTGCTGCTGCTGTTGCTGCT------------TCTGCTGCTGGGCTGCACGAAAGAGCCTGTACTTCTC
991556(MO)     TTCTTCTTCTTCCTCCTCCTCCTCCTCCTCCTCCTCCTGTTGCTTCTGCTGCTGCTGTTGCTGCTGCTGTTGCTGCTTCTGCTGCTGGGCTGCACGAAAGAGCCTGTACTTCTC
991557(SB)     TTCTTCTTCTTCCTCCTCCTCCTCCTCCTCCTCCTCCTGTTGCTTCTGCTGCTGCTGTTGCTGCTGCTGTTGCTGCTTCTGCTGCTGGGCTGCACGAAAGAGCCTGTACTTCTC
991558(PR)     TTCTTCTTCTTCCTCCTCCTCCTCCTCCTCCTCCTCCTGTTGCTTCTGCTGCTGCTGTTGCTGCTGCTGTTGCTGCTTCTGCTGCTGGGCTGCACGAAAGAGCCTGTACTTCTC
               ********************************   ****************************           ************************************
```



Supplementary Figure 1. Validation of a complex INDEL. Capillary-based "Sanger" sequencing confirms a complex variant within the protein-coding sequence of the X chromosome gene, *SHROOM4*. A variant haplotype consisting of a 3 bp and a 12 bp insertion (separated by 18 bp) is transmitted from a mother (M) to two of her children (SB-female and PR-male). The father carries a simple allele lacking these insertions and is identical to the reference. Both insertions occur in low complexity repetitive protein-coding tracts and neither were detected by GATK or Pindel.

Supplementary Figure 2



Supplementary Figure 2. Read-depth filtering for *de novo* events. A validated *de novo* insertion of 1 bp within *FOXP1* was detected by both SPLITREAD and BWA/GATK analyses. Sequence read-depth comparison of exon 9 of *FOXP1* for the mother (blue), father (red) and proband (green), shows a significant decrease in the read-depth in the proband when compared to parents corresponding to the insertion size. Note that the distribution of read-depth (mother and father) is not uniform over the exome.

Supplementary Table 1: SPLITREAD Validation for the NA12891 exome.

Gene	Event	Size	Sequence (6bp flanking)	Chrom	Start	End	Perfect Support	Unbalanced Support	Status	Type
*SLC2A7**	Ins	11	CCACCTTGTGCCCACCTTGTGCG	chr1	9007616	9007626	7	5	Conf.	Het.
LCE4A	Ins	18	TCTGGGGGCTGCTGTAGCTCTGGGGGCTGT	chr1	150948305	150948322	2	30	Conf.	Hom.
*KCNMA1**	Ins	1	TGCTTTTTTTTTT	chr10	78399792	78399792	3	11	FP	NA
*MKI67**	Del	2	TGTGTGTGTGTGTG	chr10	129795112	129795115	3	3	FP	NA
*PHF21A**	Del	3	ctgggcgtggtggtg	chr11	45957943	45957947	3	62	FP	NA
*SLC22A9**	Del	1	CAGCACAAAAAAA	chr11	62906256	62906258	2	8	Conf.	Het.
*TDG**	Ins	1	GAAAAAAATTACA	chr12	102897862	102897862	2	3	Conf.	Het.
NCOR2	Ins	3	TGCTGCTGCTGCTGC	chr12	123453033	123453035	2	3	Conf.	Hom.
*TSC22D1**	Del	3	gttgctgctgctgct	chr13	44046694	44046698	2	4	FP	NA
*SLC35F5**	Del	1	AAAAAAAAGCTAA	chr2	114216752	114216754	2	3	Conf.	Hom.
C21orf62	Ins	1	TGATTTTAAGGCT	chr21	33088060	33088060	2	18	Conf.	Het.
*TRAK1**	Ins	6	GAGGAGGAGGAGGAGGAGAG	chr3	42226590	42226595	2	7	Conf.	Hom.
*YEATS2**	Del	3	GCCggaggaggagga	chr3	184976441	184976445	2	19	Conf.	Het.
MAP3K1	Del	3	caacaacaacaacaa	chr5	56213619	56213623	21	5	Conf.	Hom.
*COL14A1**	Del	2	TTTTTTTTTAGGAT	chr8	121362338	121362341	2	3	Conf.	Het.
*UBE2R2**	Del	3	actgttatgatgatg	chr9	33907211	33907215	2	5	FP	NA
*VCP**	Ins	1	TTTTTTTTTGTGG	chr9	35049653	35049653	2	3	FP	NA
HRC	Ins	3	TCATCATCATCATCA	chr19	54349554	54349556	15	11	Conf.	Het.
*HYDIN**	Del	15	ctccaggcgctccttctccgtgcgctc	chr16	69512199	69512215	4	14	Conf.	Het.
*KRTAP5**	Del	30	TAAGCCTTACTGCTGCCAGTCCAGCTGCTGTAAGCCCTACTG	chr11	1608186	1608217	4	5	Conf.	Het.
*WDR66**	Ins	15	AGGAGGAGAAAGAGGAGGAGGGGAAGG	chr12	120843784	120843798	3	16	Conf.	Het.
*MAP3K4**	Del	3	gctgctgctgctgct	chr6	161439369	161439373	2	7	Conf.	Het.
*FERD3L**	Ins	3	cctcttcctcctcct	chr7	19151287	19151289	2	6	Conf.	Het.
*MEOX2**	Del	3	tgatggtggtggtgg	chr7	15692325	15692329	2	3	Conf.	Het.

*Variants were called exclusively by SPLITREAD but not but by either PINDEL v0.2.0 or BWA/GATK. FP: false positive, Conf: confirmed, Hom: homozygous, Het: heterozygous

Supplementary Table 2. The list of all structural variants and the frequency of these variants among 11 samples.

Chromosome	Start	End	Event	Size	Number of Samples	Sample ID	Genes
chr1	7812662	7812716	Deletion	54	1	NA18507	*PER3*
chr1	150938183	150948306	Deletion	10123	8	NA15510,NA18555,NA19240,NA19129,NA18507,NA18956,NA12891,NA18517	*LCE2A,LCE4A*
chr1	150948293	151015622	Deletion	67329	6	NA12891,NA19238,NA12878,NA18517,NA18555,NA19240	*LCE4A,C1orf68,KPRP,LCE1F*
chr1	229539561	229541839	Deletion	2278	3	NA18555,NA19129,NA12878	*EXOC8,C1orf124*
chr10	124321939	124323340	Deletion	1401	1	NA18517	*DMBT1*
chr11	1006690	1007869	Deletion	1179	2	NA12891,NA12892	*MUC6*
chr11	1006701	1008387	Deletion	1686	4	NA18517,NA19238,NA12891,NA12892	*MUC6*
chr11	1006762	1007269	Deletion	507	1	NA12891	*MUC6*
chr11	1006901	1007408	Deletion	507	2	NA12891,NA12892	*MUC6*
chr11	1006991	1007501	Deletion	510	1	NA19238	*MUC6*
chr11	1006994	1008176	Deletion	1182	2	NA19238,NA19240	*MUC6*
chr11	1007030	1007537	Deletion	507	7	NA12878,NA18517,NA18956,NA18507,NA18555,NA19238,NA19240	*MUC6*
chr11	1007035	1007707	Deletion	672	2	NA12891,NA12892	*MUC6*
chr11	1007037	1007544	Deletion	507	2	NA12891,NA12892	*MUC6*
chr11	1007151	1007823	Deletion	672	1	NA12891	*MUC6*
chr11	1007201	1007873	Deletion	672	1	NA12891	*MUC6*
chr11	1007210	1008389	Deletion	1179	2	NA12891,NA12892	*MUC6*
chr11	1007225	1007897	Deletion	672	2	NA12891,NA19238	*MUC6*
chr11	1007300	1007972	Deletion	672	1	NA12891	*MUC6*
chr11	1007312	1008491	Deletion	1179	2	NA12891,NA12892	*MUC6*
chr11	1007367	1008039	Deletion	672	2	NA12891,NA19238	*MUC6*
chr11	1007428	1008100	Deletion	672	1	NA12891	*MUC6*
chr11	1007542	1007707	Deletion	165	2	NA12891,NA12892	*MUC6*
chr11	1007661	1008171	Deletion	510	1	NA19238	*MUC6*
chr11	1007771	1008278	Deletion	507	2	NA12891,NA12892	*MUC6*
chr11	1007903	1008410	Deletion	507	2	NA12891,NA12892	*MUC6*
chr11	1007987	1008494	Deletion	507	2	NA12891,NA12892	*MUC6*
chr11	1082402	1082474	Deletion	72	1	NA18956	*MUC2*

chr11	1082491	1082629	Deletion	138	2	NA12892,NA19129	*MUC2*
chr11	1082510	1082741	Deletion	231	2	NA15510,NA19238	*MUC2*
chr11	1082513	1082627	Deletion	114	1	NA19238	*MUC2*
chr11	1082556	1082622	Deletion	66	3	NA15510,NA19238,NA19240	*MUC2*
chr11	1082646	1082739	Deletion	93	1	NA19129	*MUC2*
chr11	1082675	1082744	Deletion	69	1	NA18507	*MUC2*
chr11	1082905	1083043	Deletion	138	1	NA12892	*MUC2*
chr11	1082957	1083170	Deletion	213	1	NA19238	*MUC2*
chr11	1082961	1083105	Deletion	144	1	NA18555	*MUC2*
chr11	1082961	1083243	Deletion	282	3	NA18956,NA19238,NA18507	*MUC2*
chr11	1082967	1083318	Deletion	351	1	NA19238	*MUC2*
chr11	1083012	1083222	Deletion	210	2	NA19238,NA19240	*MUC2*
chr11	1083027	1083585	Deletion	558	2	NA19129,NA19238	*MUC2*
chr11	1083029	1083104	Deletion	75	1	NA19238	*MUC2*
chr11	1083037	1083319	Deletion	282	1	NA19238	*MUC2*
chr11	1083056	1083539	Deletion	483	6	NA15510,NA18507,NA19238,NA18956,NA19129,NA12878	*MUC2*
chr11	1083101	1083590	Deletion	489	2	NA19238,NA19129	*MUC2*
chr11	1083105	1083318	Deletion	213	1	NA19238	*MUC2*
chr11	1083149	1083221	Deletion	72	1	NA19238	*MUC2*
chr11	1083149	1083290	Deletion	141	1	NA19238	*MUC2*
chr11	1083165	1083510	Deletion	345	4	NA12891,NA12892,NA18555,NA19238	*MUC2*
chr11	1083245	1083320	Deletion	75	1	NA19238	*MUC2*
chr11	1083265	1083541	Deletion	276	2	NA15510,NA19129	*MUC2*
chr11	1083400	1083538	Deletion	138	4	NA18517,NA12878,NA15510,NA19129	*MUC2*
chr11	1083441	1083510	Deletion	69	1	NA12892	*MUC2*
chr11	1083469	1083538	Deletion	69	9	NA19240,NA15510,NA18555,NA18956,NA19129,NA12878,NA18507,NA19238,NA18517	*MUC2*
chr11	1083565	1083634	Deletion	69	4	NA18507,NA18517,NA19240,NA19238	*MUC2*
chr11	7673481	7673797	Deletion	316	10	NA19240,NA19129,NA18517,NA12892,NA12891,NA18507,NA19238,NA12878,NA15510,NA18555	*OVCH2*

chr11	48223585	48303521	Deletion	79936	2	NA12891,NA12892	*OR4X2,OR4X1,OR4S1,OR4C3*
chr11	63082288	63114014	Deletion	31726	1	NA19238	*HRASLS2,PLA2G16*
chr11	123625829	123640235	Deletion	14406	1	NA18517	*OR8G5,OR8G1*
chr11	133656586	133719596	Deletion	63010	2	NA12891,NA12892	*GLB1L2,GLB1L3*
chr12	11311591	11311718	Deletion	127	1	NA12891	*PRB3*
chr12	21087522	21241135	Deletion	153613	1	NA18517	*LST-3TM12,SLCO1B1*
chr12	54956248	54962472	Deletion	6224	1	NA12892	*CS*
chr12	62465118	62482136	Deletion	17018	1	NA19238	*TMEM5*
chr12	102903637	102904856	Deletion	1219	3	NA12891,NA12892,NA19240	*TDG*
chr12	107541480	107541840	Deletion	360	1	NA12891	*SELPLG*
chr12	107541737	107541827	Deletion	90	4	NA12891,NA12878,NA18507,NA19238	*SELPLG*
chr13	20627290	20627832	Deletion	542	4	NA15510,NA18956,NA12878,NA12891	*SKA3*
chr13	20627948	20630058	Deletion	2110	2	NA12878,NA18956	*SKA3*
chr13	20630263	20632039	Deletion	1776	3	NA12878,NA15510,NA18956	*SKA3*
chr13	20632128	20633930	Deletion	1802	1	NA18956	*SKA3*
chr13	20634015	20640127	Deletion	6112	3	NA12878,NA18956,NA12891	*SKA3*
chr13	20640541	20644480	Deletion	3939	1	NA15510	*SKA3*
chr13	20644642	20648512	Deletion	3870	2	NA15510,NA12878	*MRP63,SKA3*
chr13	23278207	23281987	Deletion	3780	2	NA18555,NA19240	*MIPEP*
chr13	44421975	44431517	Deletion	9542	2	NA12891,NA12892	*NUFIP1*
chr14	92530100	92552778	Deletion	22678	3	NA12878,NA12892,NA19129	*ITPK1*
chr16	3194465	3205829	Deletion	11364	1	NA18517	*OR1F2P,OR1F1*
chr17	1358937	1359228	Deletion	291	1	NA12892	*INPP5K*
chr17	1358988	1359223	Deletion	235	1	NA12891	*INPP5K*
chr17	1359104	1359227	Deletion	123	1	NA12878	*INPP5K*
chr17	70424628	71262644	Deletion	838016	2	NA19238,NA18956	*USH1G,ICT1,NT5C,MRPS7,LOC100287042,LOC643008,LOC100130933,OTOP2,OTOP3,C17orf28,CDR2L,ATP5H,KCTD2,SLC16A5,ARMC7,SUMO2,NUP85,GGA3,SLC25A19,KIAA0195,CASKIN2,TSEN54,MYO15B,SAP30BP,ITGB4,HN1,MIF4GD,GRB2,LLGL2,RECQL5*
chr18	74957635	74971340	Deletion	13705	1	NA19238	*ATP9B*
chr19	1562849	1566284	Deletion	3435	1	NA12878	*TCF3*
chr19	4462139	4462931	Deletion	792	1	NA18555	*PLIN4*
chr19	4462506	4462605	Deletion	99	4	NA19238,NA19240,NA18507,NA19129	*PLIN4*

chr19	4462551	4462749	Deletion	198	2	NA12891,NA12892	*PLIN4*
chr19	4463228	4463525	Deletion	297	1	NA18956	*PLIN4*
chr19	8863577	8869249	Deletion	5672	1	NA12891	*MUC16*
chr19	8870620	8879469	Deletion	8849	2	NA12891,NA12892	*MUC16*
chr19	8876428	8879241	Deletion	2813	7	NA18956,NA12878,NA12892,NA15510,NA18555,NA12891,NA18517	*MUC16*
chr19	8882082	8894638	Deletion	12556	1	NA12892	*MUC16*
chr19	14813206	14875502	Deletion	62296	5	NA15510,NA19238,NA18517,NA19240,NA18956	*OR7A10,OR7A17*
chr19	14813472	14852950	Deletion	39478	1	NA12892	*OR7A10,OR7A17*
chr19	15591497	15619058	Deletion	27561	6	NA12891,NA15510,NA18507,NA18555,NA18956,NA19238	*CYP4F3,CYP4F8*
chr19	15594062	15621003	Deletion	26941	1	NA12892	*CYP4F3,CYP4F8*
chr19	15624610	15656855	Deletion	32245	3	NA18956,NA18517,NA19238	*CYP4F3,CYP4F12*
chr19	40694203	40694254	Deletion	51	5	NA12891,NA18956,NA12878,NA18555,NA15510	*DMKN*
chr19	57808757	57808841	Deletion	84	1	NA12891	*ZNF83*
chr2	31214381	31214480	Deletion	99	1	NA18507	*GALNT14*
chr2	111470015	111470069	Deletion	54	2	NA19129,NA18517	*ACOXL*
chr2	179005228	179009118	Deletion	3890	2	NA18956,NA18555	*PRKRA,MIR548N*
chr2	179014678	179016240	Deletion	1562	1	NA18956	*PRKRA,MIR548N*
chr2	179020558	179023213	Deletion	2655	2	NA18956,NA18555	*PRKRA,MIR548N*
chr2	179023385	179023939	Deletion	554	2	NA18956,NA18555	*PRKRA,MIR548N*
chr2	227902745	227903587	Deletion	842	1	NA19238	*MFF*
chr2	240630131	240630859	Deletion	728	2	NA15510,NA19238	*PRR21*
chr2	240630785	240630897	Deletion	112	1	NA19238	*PRR21*
chr2	240630841	240630897	Deletion	56	2	NA18555,NA19238	*PRR21*
chr20	3684250	3687183	Deletion	2933	1	NA18507	*C20orf27*
chr20	23494625	23532272	Deletion	37647	7	NA18507,NA18517,NA18555,NA18956,NA19238,NA15510,NA19129	*CST9,CST9L*
chr20	48034298	48037818	Deletion	3520	4	NA15510,NA18507,NA18555,NA19238	*SNAI1*
chr21	31036424	31041217	Deletion	4793	1	NA18956	*KRTAP21-2*
chr22	28493537	28495666	Deletion	2129	3	NA12878,NA15510,NA19129	*UQCR10*

						NA12878,NA15510,NA18555,N	
chr22	36449831	36450128	Deletion	297	5	A18956,NA19238	*TRIOBP*
chr3	110529562	110530427	Deletion	865	3	NA18517,NA19240,NA18555	*DPPA4*
chr3	198977721	199046923	Deletion	69202	1	NA19129	*LRCH3,FYTTD1*
chr4	1378351	1379400	Deletion	1049	1	NA12891	*CRIPAK*
chr4	1378351	1378442	Deletion	91	2	NA12891,NA12892	*CRIPAK*
chr4	1378390	1378516	Deletion	126	2	NA18507,NA15510	*CRIPAK*
chr4	1378394	1378612	Deletion	218	1	NA18507	*CRIPAK*
chr4	1378398	1379048	Deletion	650	1	NA19238	*CRIPAK*
chr4	1378544	1379099	Deletion	555	2	NA19238,NA19240	*CRIPAK*
chr4	1378553	1379048	Deletion	495	2	NA19238,NA19240	*CRIPAK*
chr4	1378573	1379279	Deletion	706	1	NA19238	*CRIPAK*
chr4	1378613	1378983	Deletion	370	1	NA12892	*CRIPAK*
chr4	1378638	1379223	Deletion	585	2	NA15510,NA18507	*CRIPAK*
chr4	1378643	1379384	Deletion	741	1	NA19238	*CRIPAK*
chr4	1379073	1379382	Deletion	309	2	NA19240,NA19238	*CRIPAK*
chr4	12979377	12987271	Deletion	7894	1	NA18555	*RAB28*
chr4	88754707	88755496	Deletion	789	1	NA12891	*DSPP*
chr4	88755030	88755783	Deletion	753	1	NA12892	*DSPP*
chr4	88755086	88755485	Deletion	399	2	NA12891,NA12892	*DSPP*
chr4	88755161	88755488	Deletion	327	2	NA12891,NA12892	*DSPP*
chr4	88755174	88755483	Deletion	309	1	NA18555	*DSPP*
chr4	88755376	88756348	Deletion	972	1	NA12891	*DSPP*
chr4	88755467	88755794	Deletion	327	1	NA12891	*DSPP*
chr4	88755489	88756518	Deletion	1029	1	NA18555	*DSPP*
chr4	88755871	88755952	Deletion	81	1	NA19238	*DSPP*
chr4	88756068	88756239	Deletion	171	1	NA12891	*DSPP*
chr4	88756068	88756158	Deletion	90	1	NA12891	*DSPP*
chr4	88756074	88756452	Deletion	378	1	NA12892	*DSPP*
chr4	88756077	88756176	Deletion	99	1	NA12892	*DSPP*
chr4	88756088	88756376	Deletion	288	1	NA19238	*DSPP*
chr4	88756090	88756549	Deletion	459	1	NA18517	*DSPP*
chr4	88756092	88756281	Deletion	189	2	NA12892,NA19238	*DSPP*
chr4	88756092	88756164	Deletion	72	4	NA19238,NA19240,NA18517,NA19129	*DSPP*
chr4	88756093	88756552	Deletion	459	4	NA19240,NA18507,NA19129,NA12892	*DSPP*

chr4	88756095	88756149	Deletion	54	6	NA18507,NA19240,NA15510,NA18517,NA12891,NA12892	*DSPP*
chr4	88756096	88756348	Deletion	252	1	NA19238	*DSPP*
chr4	88756104	88756239	Deletion	135	1	NA12892	*DSPP*
chr4	88756188	88756395	Deletion	207	1	NA12892	*DSPP*
chr4	88756200	88756263	Deletion	63	1	NA12891	*DSPP*
chr4	88756248	88756545	Deletion	297	1	NA12891	*DSPP*
chr4	88756302	88756455	Deletion	153	1	NA12891	*DSPP*
chr5	54599405	54600995	Deletion	1590	3	NA15510,NA18555,NA18956	*DHX29*
chr5	115258757	115266483	Deletion	7726	1	NA12891	*AP3S1*
chr5	115266587	115276956	Deletion	10369	2	NA15510,NA19240	*AP3S1*
chr5	141334703	141338050	Deletion	3347	6	NA12878,NA18517,NA18956,NA18555,NA19129,NA19238	*RNF14*
chr5	172967904	172968849	Deletion	945	1	NA19238	*BOD1*
chr5	172969042	172972738	Deletion	3696	1	NA18517	*BOD1*
chr6	31713081	31713202	Deletion	121	1	NA18956	*BAT2*
chr6	32030339	32030465	Deletion	126	1	NA12878	*RDBP*
chr6	73976167	73990727	Deletion	14560	2	NA18507,NA12878	*KHDC1L*
chr6	136624309	136630993	Deletion	6684	8	NA15510,NA18517,NA18555,NA19129,NA19238,NA19240,NA12892,NA12891	*BCLAF1*
chr6	136631171	136632268	Deletion	1097	6	NA12878,NA12892,NA18555,NA19240,NA18507,NA19238	*BCLAF1*
chr6	136632446	136634828	Deletion	2382	1	NA18555	*BCLAF1*
chr7	23318993	23319666	Deletion	673	1	NA12891	*IGF2BP3*
chr7	44840688	44841654	Deletion	966	1	NA12892	*H2AFV*
chr7	99299326	99301499	Deletion	2173	3	NA19238,NA18517,NA19240	*CYP3A43*
chr7	100467080	100467257	Deletion	177	1	NA12892	*MUC17*
chr7	100468652	100469006	Deletion	354	1	NA12892	*MUC17*
chr8	12884556	12996561	Deletion	112005	1	NA18555	*C8orf79,DLC1*
chr8	23051637	23110581	Deletion	58944	1	NA19238	*TNFRSF10A,TNFRSF10D*
chr8	30040974	30043048	Deletion	2074	1	NA12891	*TMEM66*
chr8	30043198	30043835	Deletion	637	1	NA12891	*TMEM66*
chr8	73130654	73305266	Deletion	174612	1	NA12878	*LOC392232,LOC100132891,TRPA1*
chr9	19050220	19053004	Deletion	2784	7	NA18517,NA19238,NA12878,NA15510,NA18507,NA18956,NA12891	*HAUS6*

chr9	19060300	19066601	Deletion	6301	1	NA19238	*HAUS6*
chr9	19066703	19068174	Deletion	1471	8	NA12878,NA12891,NA12892,NA15510,NA18507,NA18517,NA18555,NA19238	*HAUS6*
chr9	19068296	19070472	Deletion	2176	11	NA15510,NA19238,NA12891,NA18507,NA18555,NA19240,NA12878,NA12892,NA18517,NA18956,NA19129	*HAUS6*
chr9	19070668	19072868	Deletion	2200	8	NA12892,NA12891,NA18507,NA18517,NA12878,NA18555,NA18956,NA19129	*HAUS6*
chr9	19073046	19076736	Deletion	3690	2	NA12891,NA12892	*HAUS6*
chr9	19083300	19084313	Deletion	1013	5	NA12891,NA19129,NA12878,NA19240,NA18555	*HAUS6*
chr9	107496875	107507699	Deletion	10824	1	NA15510	*TMEM38B*
chr9	107523826	107524638	Deletion	812	1	NA15510	*TMEM38B*
chr9	131633169	131633996	Deletion	827	4	NA15510,NA18555,NA18956,NA19129	*C9orf78*
chr9	139892491	139893327	Deletion	836	5	NA12892,NA15510,NA18507,NA18517,NA19240	*CACNA1B*
chr9	139893431	139897015	Deletion	3584	4	NA18956,NA12891,NA12878,NA12892	*CACNA1B*
chrX	56312648	56313340	Deletion	692	1	NA19240	*KLF8*
chr17_random	311953	312091	Deletion	138	1	NA18507	*KRTAP1-1*

Supplementary Table 3: Summary for SPLITREAD analysis of 11 HapMap Exomes

Sample ID	Data	Coverage	Total Number of Calls	INDELs	Structural Variants	1000G Intersection		dbSNP Intersection	
NA12878	50PE	234.23	308	277	31	188	67.87%	211	76.17%
NA12891	76PE	170.7	276	213	63	148	69.48%	154	72.30%
NA12892	76PE	262.84	272	220	52	142	64.55%	155	70.45%
NA15510	50PE	220.76	308	274	34	NA	NA	206	75.18%
NA18507	50PE	247.45	353	324	29	202	62.35%	215	66.36%
NA18517	50PE	244.11	354	324	30	205	63.27%	218	67.28%
NA18555	50PE	300.5	349	313	36	201	64.22%	215	68.69%
NA18956	50PE	215.38	297	264	33	195	73.86%	199	75.38%
NA19129	50PE	228.91	347	323	24	219	67.80%	226	69.97%
NA19238	50PE	268.44	367	302	65	218	72.19%	204	67.55%
NA19240	50PE	216.67	348	320	28	239	74.69%	210	65.63%
Average	9 50PE:2 76PE	237.27	325.36	286.73	38.64	195.7	68.03%	201.18	70.45%

Calls were made only within coding region (CDS) portion excluding duplicated genes and known processed pseudogenes. The indels are required to be within 10bp of each other and the size of the events is required to be exactly the same.

Supplementary Table 4. Analysis of the 63 individuals from autism trio data.

Samples	Coverage	Thresholds (Perfect/Unbalanced)	INDELs (<=50 bp)	Structural Variants (>50 bp)
11048.fa	262.36	2,2	123	13
11048.mo	277.97	2,2	133	13
11048.p1	211.13	1,11	84	7
11307.fa	165.12	1,5	256	5
11307.mo	164.08	1,5	207	22
11307.p1	203.22	1,10	118	11
11580.fa	188.44	1,8	207	20
11580.mo	185.11	1,8	258	29
11580.p1	175.95	1,7	229	16
11666.fa	192.81	1,9	229	443
11666.mo	201.97	1,10	418	675
11666.p1	171.16	1,6	442	405
12325.fa	187.19	1,8	278	25
12325.mo	191.56	1,9	264	17
12325.p1	186.15	1,8	246	20
12499.fa	208.22	1,11	175	32
12499.mo	216.96	1,12	189	33
12499.p1	198.22	1,9	168	41
12575.fa	200.10	1,10	121	15
12575.mo	188.44	1,8	237	24
12575.p1	198.22	1,9	148	23
12647.fa	305.46	2,7	82	7
12647.mo	318.99	2,9	108	14
12647.p1	179.69	1,7	301	24
12680.fa	319.82	2,9	79	4
12680.mo	313.79	2,9	88	8
12680.p1	307.75	2,8	92	7
12681.fa	194.06	1,9	198	27
12681.mo	191.98	1,9	265	34
12681.p1	186.36	1,8	179	22
12817.fa	77.04	1,1	217	15
12817.mo	103.69	1,1	252	20
12817.p1	116.81	1,1	285	23
12974.fa	159.50	1,4	139	16
12974.mo	154.50	1,4	132	13
12974.p1	208.01	1,11	106	12
13095.fa	192.39	1,9	130	20
13095.mo	198.64	1,9	151	15

13095.p1	208.01	1,11	109	9
13253.mo	217.59	1,12	117	10
13253.fa	143.88	1,6	106	16
13253.p1	188.85	1,8	122	11
13284.mo	437.68	3,9	85	41
13284.fa	276.31	2,4	164	75
13284.p1	232.88	1,14	276	29
13683.mo	180.53	1,7	176	11
13683.fa	175.53	1,6	241	14
13683.p1	149.29	1,3	362	16
13466.mo	319.62	2,9	112	189
13466.fa	337.94	2,12	84	18
13466.p1	195.73	1,9	355	629
13708.fa	168.24	1,6	199	27
13708.mo	166.16	1,5	254	25
13708.p1	193.23	1,9	135	16
SAGE4022.mo	117.23	1,1	277	22
SAGE4022.fa	110.15	1,1	243	19
SAGE4022.p1	73.29	1,1	196	18
13970.mo	171.57	1,6	109	15
13970.fa	179.07	1,7	128	10
13970.p1	110.56	1,1	270	18
AVERAGE	200.94	1,10	190.9	56.8

Supplementary Table 5: SPLITREAD Validation from Autism Trios

Sample	Gene	Event	Size	Sequence	Chromosome	Start	End	Perfect Support	Unbalanced Support	Status
11048.p1	*C16orf84*	Deletion	6	TGGGTG	chr16	87308139	87308145	2	1	False Pos.
12575.p1	*ANKRD10*	Deletion	3	TCT	chr13	110330243	110330246	6	7	Confirmed inherited
12575.p1	*MIPOL1*	Insertion	1	A	chr14	37085933	37085934	8	12	Confirmed inherited
12681.p1	*WNT16*	Insertion	4	CCCA	chr7	120752702	120752706	7	11	Confirmed inherited
12817.p1	*TMEM165*	Deletion	2	AA	chr4	55986377	55986379	3	14	Confirmed inherited
12817.p1	*TMPRSS3*	Insertion	2	CC	chr21	42676377	42676379	6	1	Confirmed inherited
13253.p1	*TRPM3*	Insertion	1	A	chr9	72647871	72647872	6	1	Confirmed inherited
13253.p1	*SHROOM4*	Insertion	12	GCTGTTGCTGCT	chrX	50367502	50367514	4	0	Confirmed inherited
13284.p1	*MS4A14*	Deletion	2	TG	chr11	59921931	59921933	8	14	Confirmed inherited
13284.p1	*PRKCSH*	Deletion	3	AGG	chr19	11419365	11419368	6	5	Confirmed inherited
13284.p1	*FOXP1*	Deletion	1	C	chr3	71104272	71104273	1	16	False Pos.
12817.p1	*FOXP1*	Insertion	1	T	chr3	71132860	71132861	1	8	Confirmed *de novo*

Events were selected for validation because they were predicted as *de novo* .

Supplementary Table 6: Copy-number polymorphic processed pseudogenes

Chrom	Start	End	Prediction	Gene	Sample ID	Processed product size	Genomic product size	PCR	GRCb37	Frequency**	Location of Insertion Site***
chr11	123625829	123640235	DEL	*OR8G1*	NA18517	200	13639	-	-	0.03	NA
chr12	54956248	54962472	DEL	*CS*	NA12892	231	6500	+	+	0.72	chr19:18002321-18005644
chr12	**62465118**	**62482136**	**DEL**	***TMEM5***	**NA19238**	**339**	**17357**	+	**-**	**0.03**	chr11:66956724-66956756(8 OEA support)
chr12	11311591	11311718	DEL	*PRB3*	NA12891	868	1721	-	-	0.03	NA
chr13	**20632128**	**20633930**	**DEL**	***C13orf3 ****	**NA18956**	**133**	**1935**	+	**-**	**0.18**	chr15:93975222-93975277(3 OEA Support)
chr13	**20640541**	**20644480**	**DEL**	***C13ofr3 ****	**NA15510**	**333**	**4272**	+	**-**	**0.18**	chr15:93975222-93975277(3 OEA Support)
chr18	**74957635**	**74971340**	**DEL**	***ATP9B***	**NA19238**	**243**	**13948**	+	**-**	**0.03**	chr5:5311964-5311998(4 OEA Support)
chr19	1562849	1566284	DEL	*TCF3*	NA12878	392	3243	-	+	0.28	chr9:5100884-5103421
chr2	227902745	227903587	DEL	*MFF*	NA19238	338	1180	+	+	0.03	chr5:149291121-149292733; chrX:45475129-45476869; chr1:15390840-15392208
chr2	179014678	179016240	DEL	*PRKRA/PACT*	NA18956	135	1697	-	+	0.33	GRCh37-chr6_ssto_hap7:3929822-3931383; chr6_mann_hap4:3945933-3947494

chr20	3684250	3687183	DEL	*C20orf27*	NA18507	209	3137	+	+	0.03	GRCh37-chr12:49193784-49194998; chr16:30831842-30833511
chr4	12979377	12987271	DEL	*RAB28*	NA18555	100	6684	-	+	0.03	chrX:135757222-135757870; chr9:46758141-46759019
chr5	115258757	115266483	DEL	*AP3S1*	NA12891	143	7869	+	+	0.69	chr1:212722185-212723452; chr12:12495380-12496796; chr6:24858504-24859814
chr5	172967904	172968849	DEL	*FAM44B**	NA19238	306	1251	+	+	0.08	chr18:52965372-52966824; chr18:5121192-5123258; chr18:3405055-3406609
chr5	172969042	172972738	DEL	*FAM44B* *	NA18517	261	3959	+	+	0.08	chr18:52965372-52966824; chr18:5121192-5123258; chr18:3405055-3406609
chr6	31713081	31713202	DEL	*BAT2*	NA18956	221	342	-	-	0.03	NA
chr6	136632446	136634828	DEL	*BCLAF1*	NA18555	101	1247	-	-	0.72	chr5:110309875-110314622
chr7	23318993	23319666	DEL	*IGF2BP3*	NA12891	114	791	+	+	0.72	chr6:167032839-167036566
chr7	44840688	44841654	DEL	*H2AFV*	NA12892	244	917	+	+	0.64	GRCh37-chr15:93276846-93277506

chr8	**30040974**	**30043048**	**DEL**	***TMEM66 ****	**NA12891**	**189**	**2190**	+	**-**	**0.05**	chr5:120678054-120678073(6 OEA Support)
chr8	**30043198**	**30043835**	**DEL**	***TMEM66 ****	**NA12891**	**219**	**856**	+	**-**	**0.05**	chr5:120678054-120678073(6 OEA Support)
chr9	19060300	19066601	DEL	*FAM29A*	NA19238	115	6416	+	+	0.72	chr7:53222325-53902081
chr9	**107523826**	**107524638**	**DEL**	***TMEM38B****	**NA15510**	**200**	**1012**	+	**-**	**0.03**	chr3:177314857-177314889(41 OEA Support)
chr9	107496875	107507699	DEL	*TMEM38B**	NA15510	220	11044	-	-	0.03	chr3:177314857-177314889(41 OEA Support)
chrX	56312648	56313340	DEL	*KLF8*	NA19240	242	934	-	-	0.03	NA

Processed pseudogenes initially predicted as deletion events that precisely remove an intron flanked by the coding region; discovery based on analysis of 11 exomes *multiple events from same gene likely correspond to the same processed pseudogene; **Allele frequency determined based on analysis of 51 unrelated exomes; PCR product consistent with processed pseudogene ***If the processed pseudogene is in the reference the location in the reference (build 36 or GRCh37) is given. If it is not in the reference, the insertion location is based on the map locations of one end anchored reads from the first processed exon.

Supplementary Note

Table of Contents

INTRODUCTION

Next-generation sequencing technologies have launched a new era in human genetics with a wide range of possibilities for studies of human disease, evolution, and diversity. It is important to routinely and efficiently detect the full spectrum of genetic variation present in any given genome[1]. This includes single nucleotide polymorphisms (SNPs), small insertions and deletions (INDELs)[2] and larger structural variants (SVs)[3,4] (operationally distinguished from INDELs as events >50 bp in length)[5]. Despite the fact INDELs and SVs contribute significantly to human genetic variation, these forms have been more difficult to detect.

Several algorithms have recently been developed for detecting SVs and INDELs using massively parallel sequencing (MPS) technology. Read mapping methods may be generally classified into three major categories: (i) Read-pair (RP) methods infer variants based on discordancies in the distance and orientation of mate pairs mapped to the reference genome; (ii) read-depth (RD) methods infer copy-number differences based on excess or dearth in the number of reads that map to a given region; while (iii) split-read (SR) methods aim to identify SV breakpoints based on a disruption of sequence alignment continuity between a reference and test genome[6].

Most methods have been designed to handle whole-genome sequencing datasets. Split-read approaches such as Pindel[7], for example, have contributed a large fraction of the INDELs and SVs to the call set from the 1000 Genomes Pilot study. These methods were largely restricted to unique mappings of whole-genome sequencing data. Here, we detail a general combinatorial algorithm (SPLITREAD) and validate its utility to discover INDELs and SVs in exome datasets.

METHODS

Algorithm Notation

We define the set of paired-end reads of the sequenced donor genome as $R = (pe_1, pe_2,..., pe_n)$. Each paired-end read pe_k is composed of two mate-pair reads and we denote $pe_k = (R^k_1,R^k_2)$. Each read pair can be mapped to multiple locations on the reference genome. The set of alignments for a paired-end read is represented as $Al(pe_k) = (A^k_1,A^k_2,..., A^k_m)$ where A^k_m corresponds to a vector of positions and orientations: $A^k_m = [loc(R^k_1),or(R^k_1),loc(R^k_2),or(R^k_2)]_m$. We distinguish six different types of read-pair placements: two by length (concordant or discordant), three by orientation (direct, everted or inverted), and one by chromosomal location (transchromosomal)[8]. For each read pair, a read-depth value is defined for each nucleotide in the reference genome. The read-depth of each base pair *i* on the reference genome, RD (*i*), is defined as the number of the reads that spans this base pair. One-end anchored reads, represented as OEA, are defined as paired-end reads where only one end can be mapped to the reference genome[3,9,10]. OEA reads can also be mapped to multiple locations in the reference genome using the same notation defined above where either the location or orientation value is null. The main assumption of our structural variation detection method is that the unmapped reads of the OEA paired-end reads correspond to the regions harboring insertion/deletion (INDEL) or structural variant (SV) breakpoints. Given an OEA paired-end read pe_k, without the loss of generality, we assume that R^k_2 cannot be mapped to the reference genome within defined error threshold. This unmapped read with length L can be represented as a paired-end read that is split into two subsequences at breakpoint i, $sr(R^k_2,i) = (R^k_2[1:i],R^k_2[i+1,L])$ (Figure 1 in main text). All possible locations of the split-read $sr(R^k_2,i)$ are represented similarly as $Al(sr(R^k_2,i))$.

Mapping and Breakpoint Detection

The set of paired-end reads from the donor exome or genome are mapped to the reference genome using mrsFAST[11]. mrsFAST is a seed-and-extend type algorithm that maps a given read to the reference genome within a small number of errors. mrsFAST reports all possible locations in the reference genome that the given read can be mapped to within the given error threshold.

There are two popular metrics of specifying an alignment error: (i) Hamming distance[12] (number of mismatches between two equal length sequences without any gaps) and (ii) edit distance (minimum number of substitution, deletion and insertion operations to transform one sequence to the other). We used the Hamming distance as our error metric. For detecting insertions and deletions by a split-read approach without limiting detectable INDEL size, it is of utmost importance to use the Hamming distance for read mapping for the following reason. Given two sequences S and R with the same length $|S| = |R| = l$, the Hamming distance is defined as the total number of positions j, such that $S[j] \neq R[j]$. Assume that we have two similar sequences S = "AGATCCTAGC" and R = "AGATGCTAGC" where Hamming distance between these two sequences HD(S,R) = 1. After an insertion of a nucleotide, G after position 4 in sequence S, it converts into "AGATGCCTAGC" where HD(S,R) becomes 4 due to the frameshift after the insertion. Any SV in the form of an insertion or deletion causes extreme changes in the sequence content, which can be captured quite accurately using the Hamming distance criteria. Using Hamming distance ensures that the reads containing breakpoints of the deletions and insertions cannot map to the reference genome within the acceptable error threshold. As a result, the reads that contain insertions and deletions and do not map to the reference end up as OEA reads, and regardless of the event type, at the site of the event there will be a reduction in the number of reads mapping. Another advantage of Hamming distance is the computational efficiency. Optimal Hamming distance can be computed in linear time O (l) with respect to the read length l, whereas optimal edit distance can be computed in polynomial time O (l^2).

We map paired-end reads to the reference genome via the Hamming distance criteria using mrsFAST requiring ≥94% sequence identity in single-end mode. We then process all alignments to match the locations of paired-end reads and keep track of all possible concordant and discordant mappings. In the absence of these events, SPLITREAD reports all possible inverted, everted and transchromosomal mappings. Next, we calculate the read-depth of each base pair. Paired-end reads where only one end can be mapped to the reference genome—OEA reads—are identified from the remaining paired-end reads. As described previously, this guarantees that OEA reads indicate SVs, including deletions and insertions. After determining our candidate set for detecting the breakpoints for insertions and deletions, we aim to split the unmapped ends of these reads and map to the reference genome within a certain Hamming distance threshold (~6% of read length).

Split-Read Definition

We described in the previous section that the unmapped read of the OEA pairs corresponds to the reads that span a simple breakpoint of insertions and deletions and thus cannot be mapped to the reference genome in full length under the Hamming distance metric. If we can correctly identify the breakpoint of these events, we can split the OEA reads from the preprocessing step into two subsequences that will map precisely to the reference genome (Figure 1). We defined these two subsequences as split-reads.

We distinguish two types of split-reads: i) A balanced split is one in which the unmapped read decomposes into two subsequences of equal length, and ii) an unbalanced split partitions into subsequences of unequal length.

However, it is computationally infeasible to test for each possible split for an unmapped read that requires alignments proportional to the length of the read. Regardless of the breakpoint location, we note that for an unmapped OEA read with a given length *L*, there always exists a subsequence with length equal to or larger than *L/2 (*pigeonhole principle). Assuming each short read can contain one insertion or deletion event and the distribution of the breakpoints are uniform within read, the worst case is the split is in the middle of the read and there are two subsequences of length L/2. If the breakpoint is not in the middle, there will be at least one subsequence after the split with length >L/2. We can take advantage of this simple observation for detecting SVs. Given an OEA pe_k, where R^k_2 is unmapped and $|R^k_2| = L$, we define a split-read as $sr(R^k_2,L/2) = (R^k_2[1{:}L/2], R^k_2[L/2{+}1,L])$. If there is an insertion or deletion event spanning this read, at least one of these split-reads is going to map to the reference. This significantly eliminates the number of the alignments we have to examine. If we assume the split-reads as paired-end reads, a split-read with insert size 0 corresponds to a read with no SVs in it and is defined as a concordant mapping of the split-reads. All discordant mappings and OEA mappings of the split-reads indicate a structural variation. Because Hamming distance estimates are used, our approach is sensitive to even insertions and deletions with size 1.

In the case of a deletion, the distance between the split subsequences corresponds to the size of the deletion event. In the case of an insertion, the location will be flanked by the OEA mappings of the split-reads. In the case of repeat expansions (microsatellites, etc.), the mappings of split-reads may overlap. By transforming the unmapped mates of the OEA pairs into split-reads—paired-end reads with half of the original length, we can efficiently map these sequences back to the reference using mrsFAST. The main problem with this approach is that split-reads are shorter than the original read and can be mapped to multiple locations in the reference genome. Notice that split-reads consistent with a single variation in the form of an insertion or deletion will map to the exact same location on the reference genome when they traverse or split across a breakpoint. We can identify real insertions and deletions by solving the problem through clustering split-read mappings that support the same variation.

Split-Read Clustering

We also map split-reads using mrsFAST with a Hamming distance threshold at half of the initial mismatch threshold. All possible mappings of the splits are reported where there is a proper anchored read (under the assumption that the insert size between the anchored read and the split-read is within 3 standard deviations of the initial distribution). For exome data, the insert size distribution is usually dictated by the size distribution of the exons. After the capture process, no size selection is applied in exome resequencing, thus the distribution is wider compared to a Gaussian distribution usually with a maximum threshold of 300 bp. Since OEA reads are only small subset of the total paired-end read set, the running time for the split-read mapping is significantly faster than the initial mapping.

Here, we formally describe an algorithm to identify clusters of the split-read mappings where each cluster supports a certain insertion or deletion event. For each split-read, $sr(R^k_2,L/2)$ is represented similarly as $Al(sr(R^k_2,L/2)) = (al^{sr\_k}_1,al^{sr\_k}_2,\ldots,al^{sr\_k}_m)$. Each alignment $al^{sr\_k}_j$ is defined as $(loc(sr_k),or(sr_k),loc(pe^F_k),or(pe^F_k),loc(pe^R_k),or(pe^R_k))_i$. Given the set of split-read mappings, a minimum number of clusters are determined where at least one balanced split supports the insertion or deletion

event. Balanced splits can map to different locations and the same balanced split can represent multiple events. For duplicated regions and simple repeat regions, balanced splits that support the same event can map to slightly different locations. These balanced splits are also clustered together to the left-most mapping position. Given the two balanced alignments, $al^{sr\_k}_i$ and $al^{sr\_m}_j$, where the split mapping overlaps with each other, the sequences of $Ref[loc(pe^F_k)_i+1,loc(pe^R_k)_i-1]$ and $Ref[loc(pe^F_m)_i+1,loc(pe^R_m)_i-1]$ are inspected in cyclic fashion. If two sequences are similar, these balanced splits are assigned to the same event, which is the left-most location. Each cluster is represented as clu_i = (loc_start,loc_end).

OEA split-reads are inserted into this initial clustering clu_i according to the following rules:

- OEA split-read with alignment $(loc(sr_k),or(sr_k),loc(pe^F_k),or(pe^F_k),loc(pe^R_k),or(pe^R_k))_i$ where $loc(pe^F_k)$ < start_loc or $loc(pe^R_k)$ > end_loc. This criterion guarantees that the OEA split-read can be used as evidence of the corresponding event.
- $|loc(r_k) - loc(pe^F_k)|$ < average insert size + 3X standard deviations. This guarantees that the split-read maps to the anchored read concordantly.
- The sequence of the remaining split matches to the reference with respect to the insertion or deletion event.

OEA split-read mappings are assigned to any cluster that satisfies the above constraints. Each split-read pair can be mapped to multiple clusters indicating different events in the reference. The remaining set of OEA split-pairs are clustered together based on their orientation, which corresponds to the insertion of retrotransposons. Notice that the search space for OEA split-reads is limited by the balanced splits. Due to limited search space, the performance of the algorithm for cluster generation is efficient in practice.

INDEL and SV Detection with Set Cover Approximation

Each cluster clu_i is associated with a set of OEA split-reads (SR^i_1, SR^i_k, SR^i_j,…,SR^i_m). We define the detection of the structural variation as selecting a set of clusters such that the majority of the split-read mappings can be explained. We define the structural variation detection problem as the computation of the minimum number of clusters such that all split-reads are assigned to a unique cluster and the total number of the support for each cluster is maximized. Each cluster can be defined as a set of unique split-reads and the cost of the set is defined as a function of the number of elements in the set. It is possible to use any type of function, and in our method, we use the number of elements as the cost. This problem is equivalent to the weighted set cover problem for which a simple greedy algorithm provides an *O(logn)* approximation[13]. The greedy algorithm works iteratively: at each iteration it simply selects a set where the cost per uncovered element is minimal. After selecting the best set, all the split-reads that belong to the selected set are removed from the remaining sets. The costs are updated after the removal of the optimal set and iterated over the remaining sets. The algorithm terminates when all split-reads are covered. SVs or INDELs represented by these clusters are reported with their support value and the actual reads that map to their correct location in the reference genome. The main advantage of this framework is that the cost function can be defined in different ways. An alternative cost function can be defined as a combination of the split-read support and the read-depth.

Trio-Aware Read-Depth Filter for SV and INDEL Detection

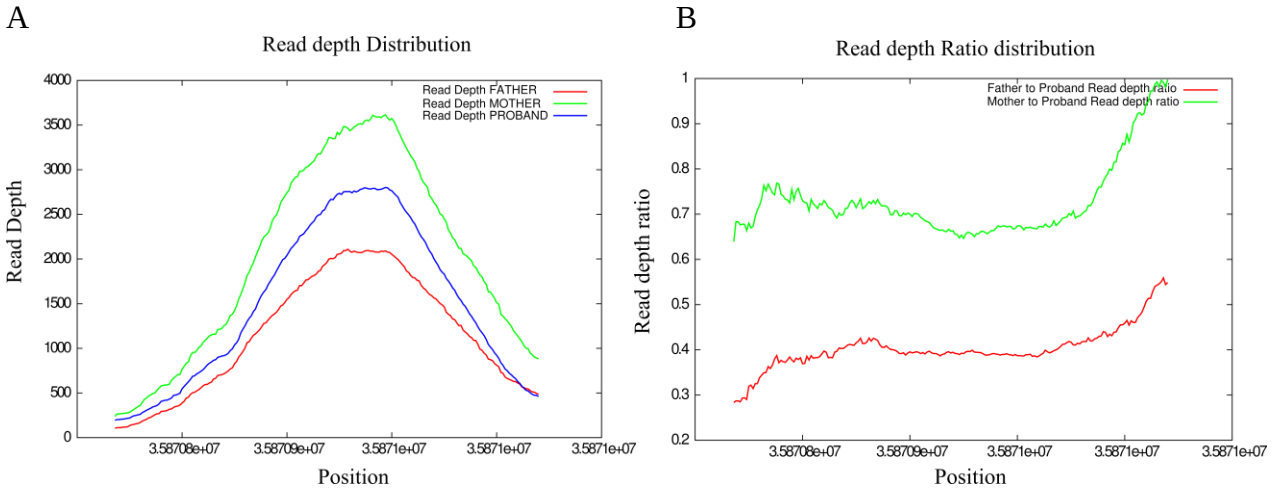
As described previously, SV and INDEL events may radically alter the number of reads mapping to the breakpoints of the event when Hamming distance is used. Although the read-depth is quite uniform for the whole-genome sequencing, there are major problems for using read-depth in exome sequencing due to variability in capture efficiency at the edges of the coding regions and the non-uniform distribution of the read-depth over the exons. Read-depth is usually distributed as a Gaussian distribution over the coding region and two-copy exons do not necessarily have the same read-depth due to different capture efficiencies. However, given a set of exomes, it is possible to use the read-depth information for verifying the insertions and deletions—the simplest case being the trio exome sequencing data of the father, mother and proband.

Although the read-depth is not distributed uniformly among the coding regions for the capture-based sequencing, there is a good correlation between the read-depth of the same exon between multiple samples. This gives us an opportunity to use the read-depth for detecting an increase or decrease in the copy number of the exons as well as detecting reduction in the read-depth at the breakpoints of the events. *de novo* INDEL detection is one of the most difficult analyses for the sequencing data. There are still numerous challenges for detecting INDELs with the most significant being false positives especially among *de novo* events. The read-depth information may be used as a filter to help eliminate these false positives and inherited events missed in the parents. As described previously, the read-depth at the breakpoints is reduced compared to the reference state.

Given all possible mappings of the exomes of the mother(M), father(F) and proband(P), the read-depth at position i is represented respectively as $RD_M(i)$, $RD_F(i)$ and $RD_P(i)$. The differences at the coverage of the samples are eliminated by normalizing the read-depth with respect to the proband using the sequence coverage ($RD'_M(i)$ = $RD_M(i)$ * Coverage(P) / Coverage(M); and $RD'_F(i)$ = $RD_F(i)$ * Coverage(P) / Coverage(F)). We define the coverage of an exome as the total number of the all possible mappings of the short reads. For an SV/INDEL predicted at position i with an exome dataset of read length l, reduction at the read-depth is expected between i-l and i+l. We compare the read-depth values for all candidate *de novo* events detected by split-reads in the proband with the parents' read-depth values and identify events that have a reduction compared to both parents. While this simple procedure works for many samples, on occasion we encountered exomes where simple normalization was less effective due to biases at the capture efficiencies. For such samples we developed a second filtering process based on the ratio of the read-depths. We calculated two ratios for parents: $RD_M(i)$/ $RD_P(i)$ and $RD_F(i)$/ $RD_P(i)$. For a true *de novo* event there should be an increase of this ratio near the breakpoint but for false positives this ratio will be constant (Supplementary Note Figure 1). Using the distribution of these two ratios, we identified events that have a local maximum around the event site and a ratio of at least 1.25 times more than the normal flanking regions. Based on these two methods, we eliminated inherited or potential false positives events. Using read-depth methods, it is possible to increase our confidence of the *de novo* candidates and reduce the number of events for validation. The normalization methods for exome sequencing are not available for more specific event detection especially for short variants, but they can be used in conjunction with multiple samples to determine evidence for these events.

The same approach also applies to the whole-genome sequencing, which is continuous and consistent such that the same copy regions have similar read-depth throughout the genome. Given single whole-genome sequencing data, the read-depth distribution within the genome can be used to determine regions with increased or decreased read-depth. Although it is sufficient to use a single genome to

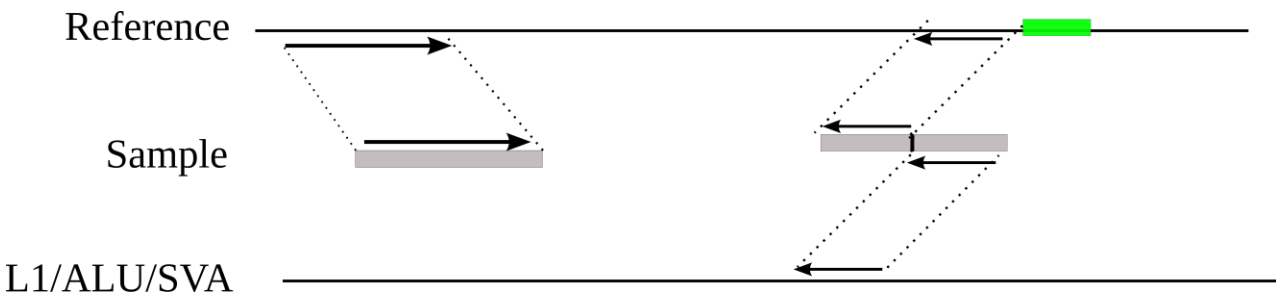
determine read-depth cost, one can apply this method similarly between exomes and genomes.



Supplementary Note Figure 1: A) The normalized read-depth distribution of trio (11666) exome data around an INDEL event in *CLPS* located at chr6. The figure represents the INDEL (middle position) and 100 bp upstream and downstream of it. B) The ratio of the read-depths of the mother and father with respect to the proband. When the normalization is not sufficient to make a call, we use the ratios. For a real event we expect to see a local maxima at the middle position, which is missing in this case meaning that this event is a false positive call.

Repeat Element Insertion Discovery Using Split-Reads

The split-read approach may be readily extended to identify common repeat element insertions such as *Alu*s, L1s and SVAs. All reads processed in the step above are mapped to the genome or exome where we define an artificial chromosome, chrN, defined in this case as consensus sequences of all common repeat elements. We then track all "transchromosomal" paired-end read mapping where one end maps to a normal chromosome and the other end maps to chrN (Supplementary Note Figure 2) repeat consensus delineating a potential repeat insertion site for the corresponding repeat element[14]. After detecting the INDELs, the remaining reads are searched for such mappings in both initial mapping and the split mappings. The possible insertion sites are clustered based on the breakpoint of the insertion on the reference genome. The minimum number of insertion sites with the maximum support is determined using a similar weighted set cover approximation described in the previous section.

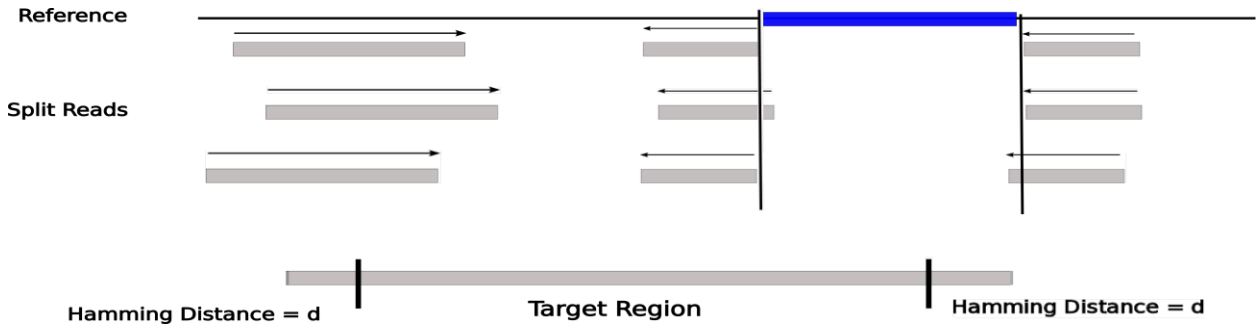


Supplementary Note Figure 2: This schematic represents a transchromosomal mapping where one end maps to the reference exome and the other end cannot be mapped to the reference. The unmapped read subsequently split into two sequences where one maps to the reference insertion site and the other end maps to a repeat element implying an insertion of this repeat element to the candidate region represented as green.

Sensitivity and Specificity of SPLITREAD

The possible alignment of the split-reads to a deletion event can be seen in Supplementary Note Figure 3. The edges of the original full-length reads can be mapped into the event within the Hamming distance limitation d. This will leave the middle portion of the read as a target region. Given a read with length |L|, the target region will be of size |T| = |L|-2d. The possible mappings of the split-read are shown in Supplementary Note Figure 3. The splits can be mapped perfectly at the center of the split. These splits can also map into the deletion event with d/2 Hamming distance from both sides. This will result in d+1 possible breakpoints in the target region around the center point. Given the target region with size |T|, there are |T|-1 possible breakpoints for a split to occur. Assuming the distribution of the reads covering the deletion event is uniform, we can safely assume that the possibility of each read split at these positions is the same. Thus, the possibility of a split occurring in the target region is (d+1)/(|L|-2d-1).

Note: Each read is generated independent from each other and the probability of the obtaining splits can be represented by a binomial distribution. For an INDEL event the probability of observing at least one balanced split in a region with N coverage can be calculated using the binomial distribution as 1-P(not balanced split)^N. For 20X sequence coverage with 76 bp reads and a 4 Hamming distance mapping threshold, the probability of detecting a heterozygous event is dependent on the coverage and at 20X this is only 55% but rises to >90% when the coverage rises to 60X. This sensitivity estimate increases to 79% for homozygous events at 20X, and to 98% at 60X coverage. Such median sequence coverages are not uncommon in many exome sequencing projects and will likely continue to rise as sequencing costs diminish. This probability can be adjusted using different Hamming distance thresholds and different read lengths. The probability of obtaining a balanced split is significantly higher than a random read mapping to the INDEL event and creating a balanced split.

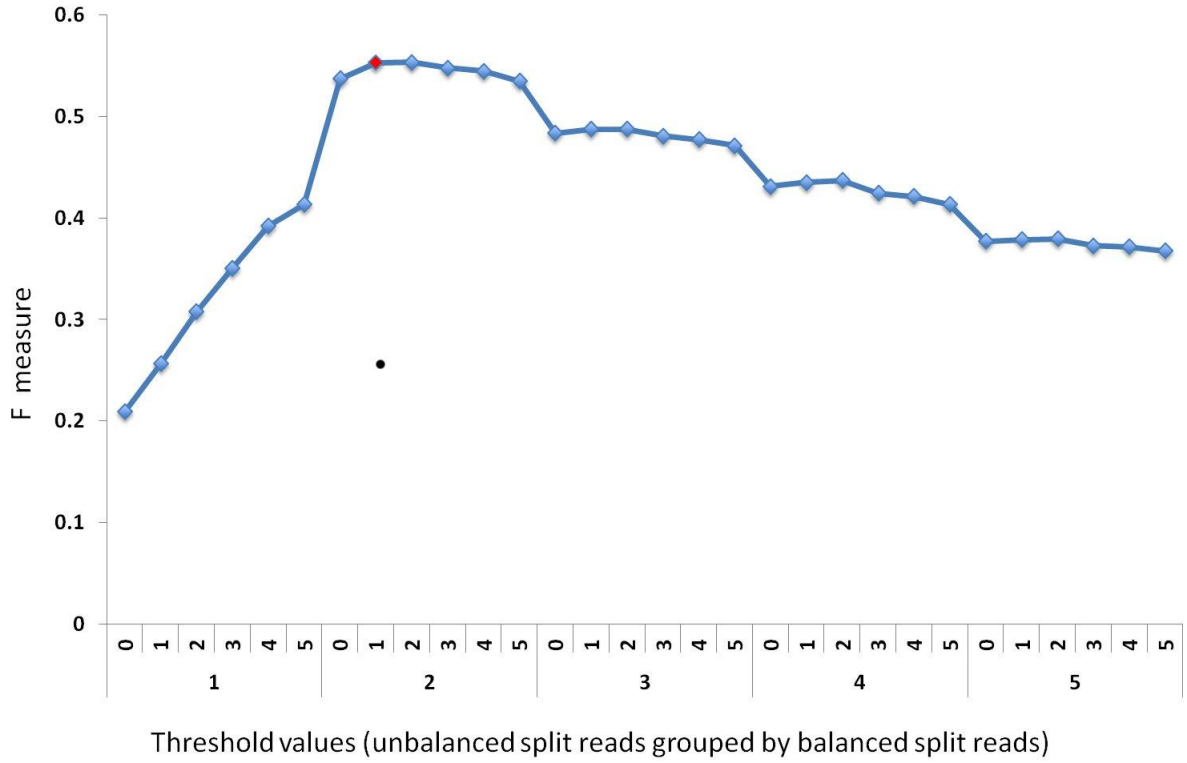


Supplementary Note Figure 3: This schematic described the possible mappings of reads to an INDEL with a given Hamming distance threshold of d. We used this information for estimating our sensitivity for detecting INDELs given the coverage of the region.

Each INDEL/SV detected by the SPLITREAD approach is reported with the number of balanced and unbalanced splits supporting each event. SPLITREAD reports all putative events necessary to reconcile all possible split-reads. False positives, however, occur because of random splits and sequencing errors. In order to establish appropriate thresholds for the number of balanced and unbalanced split-reads, we initially analyzed exome sequence data generated from a single reference sample NA12891(Supplementary Table 1,Fig. 1), which is a part of the 1000 Genomes Project [5,15]. There is an extensive amount of validated SVs and a well-established INDELs predicted by multiple, different

approaches for NA12891. It is fair to assume these events have a lower false positive rate and can be used to configure our SPLITREAD approach.

We applied SPLITREAD using different threshold values with varying numbers of balanced and unbalanced splits required to support a call. For each configuration, we compare the number of predicted events with the proportion intersecting from the 1000 Genomes Project for sample NA12891 (Figure 2A). Assuming that validated calls from this call set are correct, the slope provides the positive predictive value (PPV) of our method. We maximize the sensitivity (number of events recovered from 1000 Genomes Project) without any loss of specificity by selecting the local maxima of this line. We determine the maximum point as two balanced split-reads and two unbalanced split-reads. This approach aims to determine the ROC curve without using true negatives. To more formally address the threshold issue, we generated an F measure plot (harmonic mean of sensitivity and PPV) which agrees with the previous analysis. As evident from Supplementary Note Figure 4, the optimal threshold value is two balanced split-reads and two unbalanced split-reads.



Supplementary Note Figure 4: F measure plot to assess sensitivity and specificity of SPLITREAD.

Samples

We tested the SPLITREAD method on two sample sets: (i) exome sequencing data for 11 HapMap samples and (ii) exome sequencing data for 20 simplex families with children diagnosed with Autism Spectrum Disorder (ASD). The first dataset was generated specifically for this study while the second was published previously.

The first dataset includes 11 HapMap exomes: NA12891, NA12892, NA19238, NA12878, NA15510, NA18507, NA18517, NA18555, NA18956, NA19129 and NA19240. All samples were sequenced

using targeted in-solution capture for protein-coding sequences as described previously[16] (NimbleGen EZ Exome SeqCap v2 spanning 44 Mbp /36.5 Mbp coding region including most RefSeq gene models and several noncoding RNA regions). The post-enrichment libraries for NA12891 and NA12892 were sequenced on Illumina GA2x platform with 76 bp paired-end reads. The remaining samples were sequenced using an Illumina HiSeq platform with 50 bp paired-end reads. We generated, on average, 100 million 50 bp paired-end reads resulting in approximately 60-fold coverage (for NA12891 and NA12892). For the remaining samples, we generated 100 million 50 bp paired-reads with average 113X coverage. ~92% percent of the targeted coding regions are covered with at least 30X coverage. The genomes of all samples (except NA15510) were also sequenced as part of the 1000 Genomes Project[15] and analyzed for INDELs and SVs using numerous mapping and calling algorithms. These genomes are particularly useful for evaluating the performance of our method because most calls from whole-genome shotgun sequence data have been validated by high-density array CGH analysis, PCR and sequence analysis through fosmid end sequence mapping[3].

The second dataset consists of 20 families where there was a single child with ASD and was obtained primarily from Simon Simplex Collection[16]. All children have a normal sibling and the parents have no indication of ASD. The probands were screened for large CNVs. Exome sequencing was performed separately on each member of the family by subjecting genomic DNA derived from whole blood to in-solution hybrid capture. These samples were captured using NimbleGen EZ Exome SeqCap v1 probes. Captured libraries were sequenced on the Illumina GA2x platform with a target of 76 bp paired-end reads. All samples were sequenced with an average coverage of 200X. ~90% of the primary target was captured with at least 8-fold coverage.

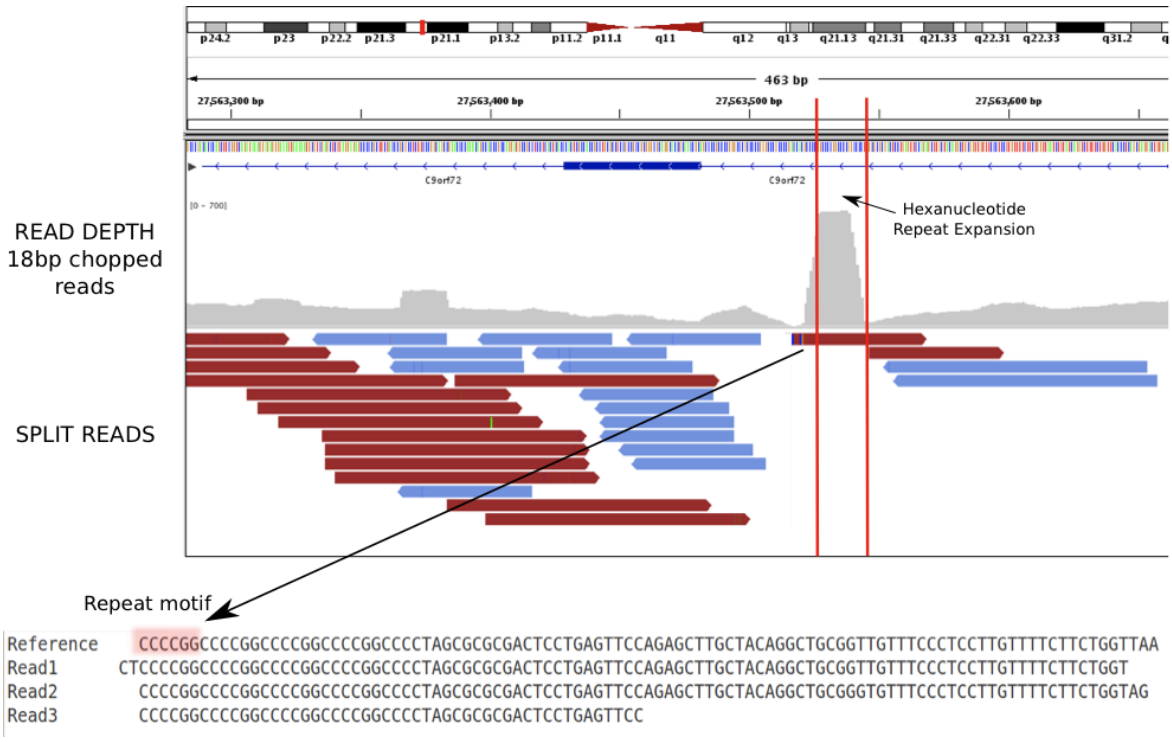
HapMap samples raw sequence data have been deposited into SRA (SRA039053).

Application of SPLITREAD to Whole-Genome Datasets

To demonstrate its applicability to whole-genome sequence, we performed two additional sets of experiments. First, we tested the efficiency of our method using the genome sequence data from ALS-FTD (amyotrophic lateral sclerosis-frontotemporal dementia) patients recently reported to be associated with 23% of familial ALS and 12% of FTD patients[17]. A region in 9p21was identified and the majority of the cases were linked to this region. Chromosome 9 was specifically isolated from a patient and sequenced using the Illumina platform with average sequence coverage of 170X. The large GGGGCC repeat expansion in the case sample was missed by the GATK INDEL calling software and the BWA alignment method was not able to map reads at the site of the repeat expansion. The variant was detected only by using a visualization tool that allowed for manual inspection of read mapping. When we apply SPLITREAD, we find that this large repeat expansion is predicted accurately. The split-reads that support the repeat expansion can be seen in Supplementary Note Figure 5. The event is detected using the unbalanced reads, which indicates that the expansion is larger than the read length, 100 bp. Due to the repetitive nature of the region, it is hard to assemble the sequence at the insertion site. However, using the split-reads we were able to detect the extended hexanucleotide repeat motif GGGGCC. The other ends of the reads deteriorate as they enter into the expansion. The average number of the motif detected is three, which corresponds to 18 bp. In order to quantify the amount of the expansion, we chopped the chr9 sequence dataset into 18 bp long subsequences and mapped these back to the insertion site using mrsFAST, which records all possible mappings. As can be seen from the figure, the read-depth at the expansion site indicates a 10-fold increase in the coverage with respect to the reference genome. The 18 bp repeat motif exists only at the insertion site in chr9 so there are no

paralogs that can interfere with the read-depth. Based on read-depth, we estimate at least 30 copies of the hexanucleotide repeat, which is consistent with repeat PCR experiments performed on this case[17].

In the second experiment we extended our analysis to one of the best characterized genomes (NA12878)[18] from the Pilot 2 project from the 1000 Genomes Project; however, we used a dataset generated by others[18] at a higher coverage with longer read length (101 bp at >80X) from the same genome. We used the INDEL calls and the SV calls from the 1000 Genomes for comparison. There are 328,527 INDELs reported by the 1000 Genomes (that intersect with exons); we were able to detect 60% of them. A total of 427,763 INDELS were predicted by SPLITREAD, where 75% intersect with either 1000 Genomes predictions or dbSNP release 130. Of the remaining events, 15% are predicted to map within segmental duplications, which were generally excluded/filtered by the INDEL callers applied to the 1000 Genomes datasets. To test the accuracy of SVs, we used 1000 Genomes[15], fosmid end sequence analysis[3], and CNV datasets generated using an ultra-dense array CGH platform[4]. We detected 42% of 1000 Genomes calls, 15% of the fosmid calls, and 51% of the array CGH calls. We predicted a total of 10,335 SVs where 29% intersect with the combined 1000 Genomes calls, Kidd et al. calls, and Conrad et al. calls. Of SV calls predicted by the SPLITREAD method, 4202 are between 50 bp and 1 kbp. This event range between 50 bp and 1 kbp is underrepresented in the available SV calls in these studies due to the difficult nature of validation. We acknowledge the limitations of our method with respect to coverage and more importantly run time. We estimate a 300-fold increase for whole-genome data compared to exome datasets. Notice that the reference sequence is not repeat masked for read mapping and all possible mappings are considered in our predictions. For whole-genome datasets, a practical implementation may be to consider use of SPLITREAD when all other mapping algorithms have failed to discover the pathogenic variant (as in the case of the ALS example).



Supplementary Note Figure 5: Discovery of a repeat expansion associated with ALS-FTD using SPLITREAD with whole-genome sequence data. Secondary read-depth mapping confirms an increase in read-depth at the junctions predicted by SPLITREAD.

Reference Sequences

We compared three different references for detecting INDELs (<=50 bp) and SVs (>50 bp). The first reference (RefSeq coding sequence) uses the defined coding sequence and 300 bp flanking sequence. In order to detect putative mobile element insertions, we also included consensus sequences of LINEs, SINEs and SVAs (as defined above). The second reference contains the first reference plus duplicated genes and processed pseudogenes. BLAT sequence similarity searches of RefSeq coding regions (50% score and length threshold) were used to define these. The whole genome (build36) defined the third reference.

Comparison to Other Methods

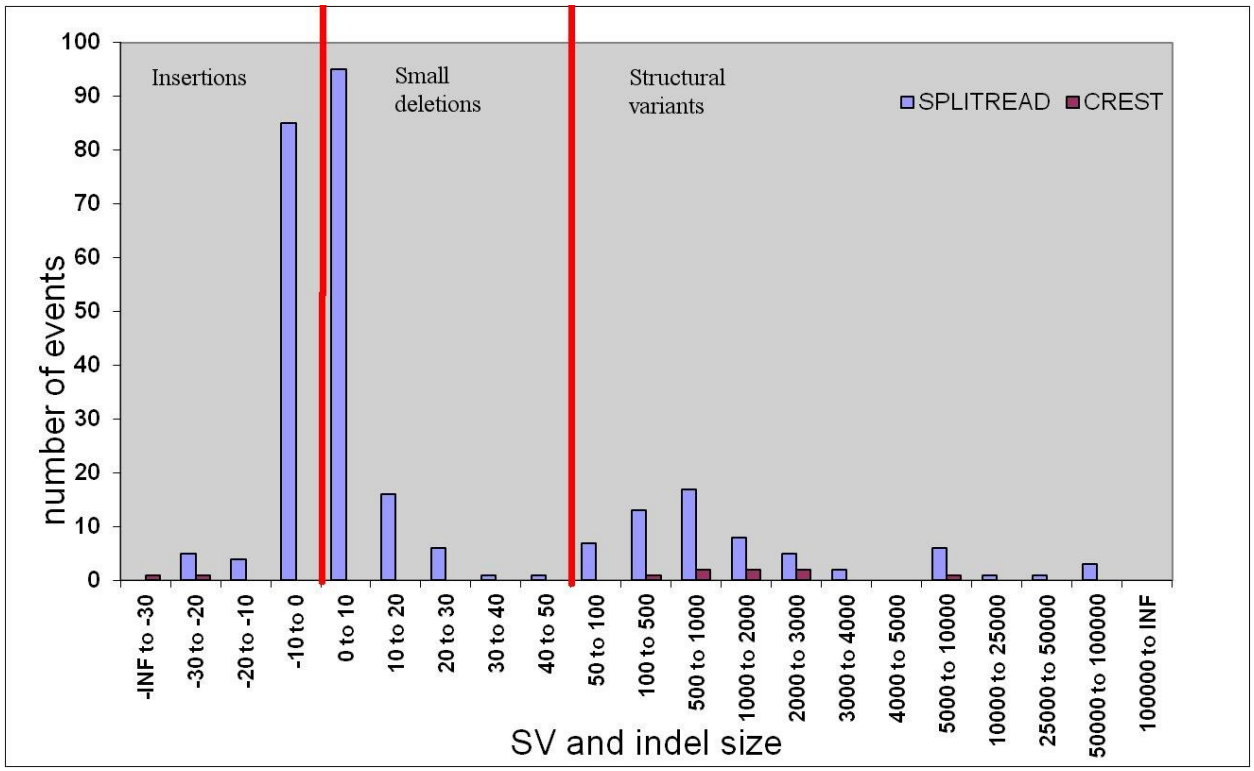
We compared our methods with two alternative INDEL detection methods. The first method, Pindel[7], is a split-based pattern growth approach for detecting the breakpoints of insertion and deletion events using short paired-end data. This method also uses the OEA pairs and is based on the insert size estimating a target alignment region. The unmapped read is aligned to this region, growing patterns from the prefix and suffix of the read. Unlike our method, Pindel computes the exact alignment with the target region and the read. Another important difference is that Pindel requires unique mappings and is not designed to consider highly duplicated regions or low complexity regions. Pindel uses BWA[19] as its mapping method and optimizes mapping to the whole-genome sequence results. For comparisons in this study, we ran Pindel v 0.2.0 using insert size=30, without BreakDancer results, and the maximum event size index set to 5 (8092 bp) as recommended on NCBI human genome reference build36.

We also compared our method to a more general pipeline used for detecting INDELs from exome sequencing data, namely BWA alignments processed with GATK suite[18]. BWA creates a local sequence alignment (pileup), which is processed by GATK for realignment. Corrections of the remaining set are filtered with various filtering options in the GATK suite. This method is limited to small INDELs rarely reporting events >15-20 bp in size. GATK 1.0.5299 is used for INDEL calling using UnifiedGenotyper -glm DINDEL option.

The third method we compared our method to was CREST[20], an algorithm that uses the next-generation sequencing reads with partial alignments to a reference genome to directly map SVs at the nucleotide level of resolution. BWA version 0.5.9-r16 is used and we used the version data 10/15/07 for CAP3 and Standalone BLAT v. 34. CREST is also run on the build36 of the human genome. Similarly, CREST is applied using the exome sequencing data. We further investigated the SVs predicted with CREST and SPLITREAD.

We plotted the predicted SV size between CREST and SPLITREAD and there is a difference in the size of events called as suggested (Supplementary Note Figure 6). We observe that CREST focuses on larger events while SPLITREAD explores a wider spectrum of genetic variation with the bulk of events (similar to the mutational spectrum) occurring within INDEL range. We suspect that the range difference between CREST and SPLITREAD may be due to the nature of exome sequence datasets. The exome data are limited to the coding regions that are, on average, 200 bp in length. The range for possibly mapping these events is short and BWA usually tries to align these small events although they are aligned incorrectly. For small events (INDEL range <50 bp) BWA does not generate sufficient clipped reads. Due to the nature of the exome data, CREST is limited to detect large events

(Supplementary Note Figure 6). CREST only predicts events for large SVs, whereas SPLITREAD has a wider range. The intersection between the two methods in the large deletions is quite good. Based on the observation on the predictions of these methods using the sample NA12891, we observed that there is not a single method that predicts all the events. Each method complements each other and it is quite important to use different methods together to see the whole range of SVs and INDELs. SPLITREAD adds considerable value in the discovery of underrepresented but biologically important classes of INDELs and structural variation.

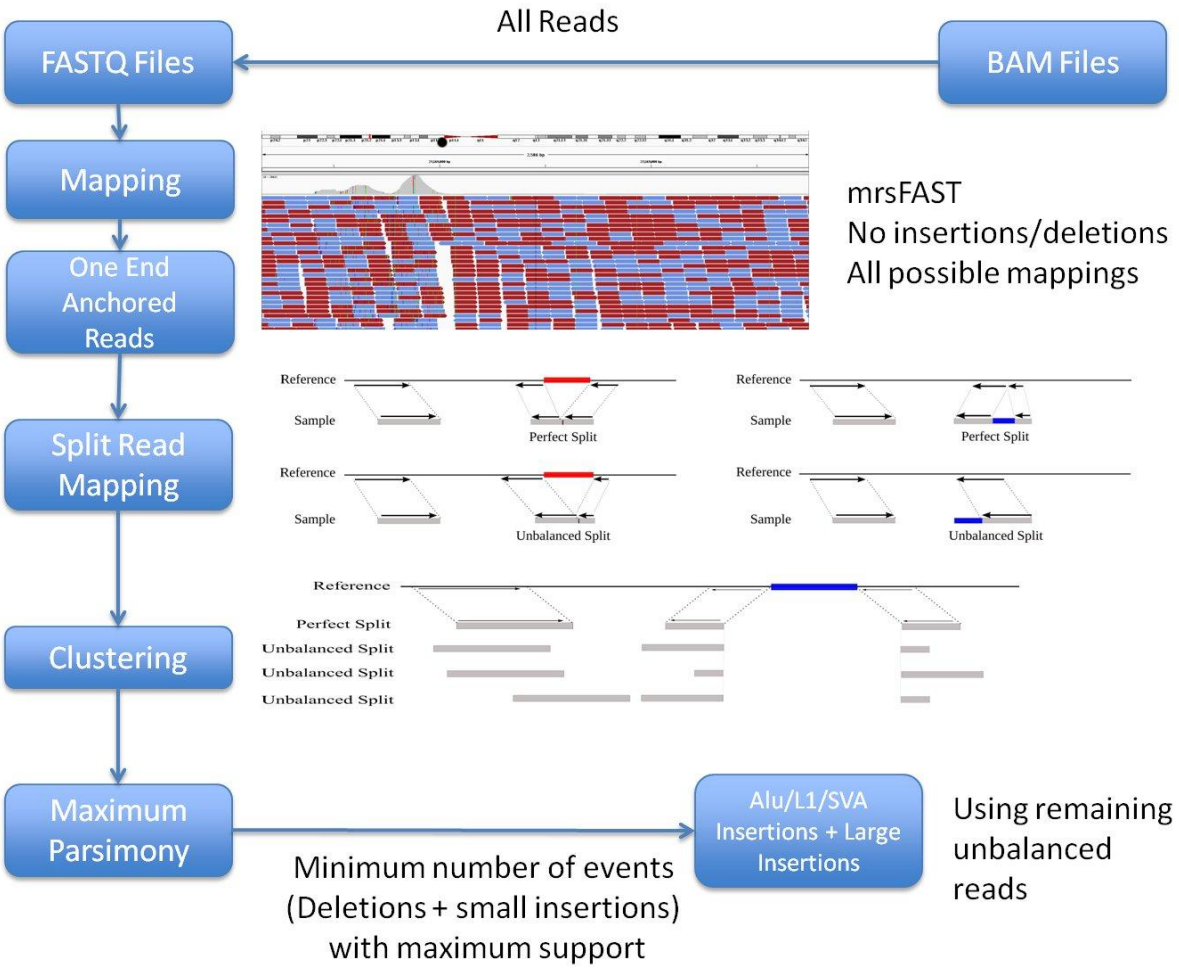


Supplementary Note Figure 6: Size spectrum of INDELs and SVs discovered by CREST and SPLITREAD. We observe that CREST detects primarily larger SVs, while SPLITREAD shows a wider range of detection ranging from INDELs to SVs. Here we show only the insertions fully characterized by SPLITREAD (exact breakpoint and the inserted sequence) as opposed to novel insertions detected by OEA strategies of indeterminate size.

SPLITREAD Program

SPLITREAD is implemented in C (available at http://splitread.sourceforge.net) and requires as input paired-end mapping information generated by mrsFAST from underlying raw sequence data (FASTQ format). The current version of SPLITREAD is designed for reads generated by the Illumina platform. It is possible to use other mapping methods that can be set up not to allow insertions and deletions and reporting all possible mapping positions for each read. Standard output includes the base pair resolved location of the insertion/deletion, level of support (number of reads supporting each event), and the total Hamming distance of the read mappings. The deletion events and small insertions are processed first. We can detect any deletion from 1 bp to 10 Mbp. Insertions less than the size of the read are also detected. The remaining reads are used for identifying the insertions and repeat expansions that are larger than the read length. It is possible to use methods such as NovelSeq[10] to identify the insertions using OEA reads around the insertion site. We include the mobile elements (*Alu*s, L1s, SVAs and

processed pseudogenes) in our reference, so we can detect these insertions without any size limit. The most important advantage is that our method can handle the events in the low complexity regions or duplicated regions such as segmental duplications. Final call sets can be filtered for the support and Hamming distance adjusted based on exome/genome sequence coverage. SPLITREAD may be used as a standalone program on a single CPU or it can be run on a cluster with multiple nodes. It is possible to generate custom reference sequences for better performance or better sensitivity. The process flowchart of the SPLITREAD method can be seen in Supplementary Note Figure 7.



Supplementary Note Figure 7: Flowchart for data processing using SPLITREAD.

Many methods for detecting structural variation using high throughput sequencing data are currently available[5,6]. Read-pair (RP) based methods such as VariationHunter[21], BreakDancer[22] and MoDIL[23] have limited power in analyzing exome sequencing data, and in most cases exact breakpoints cannot be defined. Exome capture protocols typically do not involve size selection and, as a result, cannot capture the smaller structural variants and INDELs due to a wider variance in insert size. SPLITREAD does not depend on the insert size of read pairs for detecting events. Moreover, RP-based methods rely on read pairs that span the event site, yet SPLITREAD depends only on a read traversing the breakpoint and its length for alignment accuracy increasing SV detection sensitivity. There are, however, limitations related to sequence coverage and the properties of the underlying sequence in the breakpoint, as in any sequence analysis algorithm**.** Another limitation of SPLITREAD is the dependence on a balanced split to seed an event. This is directly dependent upon coverage. Given 76 bp

reads, the chance of detecting a heterozygous event is dependent on the coverage and at 20X this is only 55% but rises to >90% when the coverage rises to 60X. This sensitivity estimate increases to 79% for homozygous events at 20X, and to 98% at 60X coverage. Such median sequence coverages are not uncommon in many exome sequencing projects and will likely continue to rise as sequencing costs diminish**.** A recently developed method, CREST[20], is based on local assembly of soft-clipped (partially aligned) reads identified by the BWA[19] mapper, and its performance depends on the read aligner. As described by Wang *et al.*[20], CREST was not designed for small INDEL detection due to the lack of soft-clipping signatures for events <50 bp. Most SV detection algorithms utilize only uniquely mapped reads, which limits the use in relatively less complex areas of the genome. In contrast, SPLITREAD performs a combinatorial analysis of split-read (SR) alignments, which is tolerant to the alignment errors while still using ambiguously mapping reads. This makes it possible for SPLITREAD to discover INDELs in repeat-rich regions including microsatellites at exact breakpoint resolution, with no theoretical upper or lower bounds on detectable event size.

SPLITREAD can detect insertions and deletions without any size limitation. The size spectrum of the insertions that can be accurately characterized by SPLITREAD is bound by the read length; however, it is possible to detect approximate breakpoints of larger insertions, although the content and the full extent of the inserted sequence will remain unknown. Such larger insertions are detected by identifying clusters of OEA reads[9]; i.e. reads proximal to the insertion locus will map to the forward strand where the distal reads will map to the reverse strand, and the unmapped reads will not be split into two (balanced or unbalanced). Moreover, in the case of an insertion, the distance between the "proximal cluster" and the "distal cluster" will be smaller than the insert size of the library. Note that this "cluster distance" can be much larger for deletions, and the split reads can be detected in both breakpoints of the deletion event, although they do not need to be balanced splits. It is possible to use alternative approaches such as NovelSeq[10] as a post-processing step to fully characterize the larger insertions.

PCR Validation of Processed Pseudogenes

Pseudogenes were validated using PCR amplification and primers specific to flanking exons of the predicted intronic deletions (Supplementary Note Table 1). Pseudogene presence was tested by amplification only in the HapMap individual in which the deletion was detected using manufacturer's protocol [PCR Master (Roche)]. For the two genes (*MFF* and *TMEM66*) that were genotyped in multiple HapMap samples, the presence of both the pseudogene and the original gene were detected using long-range PCR amplification following manufacturer's protocol [Expand Long Template PCR System (Roche)].

For the PCR amplifications using PCR Master (Roche) kit, we performed reactions in 12.5 µl volumes with 1X PCR Mastermix, 20 ng of HapMap DNA, and 0.4 µM of primers. The thermocycler program used is as follows: (1) 94°C for 4:00, (2) 94°C for 0:30, (3) 55°C-58°C for 0:30, (4) 72°C for 1:30, (5) steps 2 through 4 repeated 35 times, and (6) 72°C for 7:00. For the long-range PCR amplifications using Expand Long Template PCR System (Roche), we performed reactions in 15 µl volumes with 1X Expand Long Template buffer 1, 350 µM dNTPs, 20 ng of HapMap DNA, 0.3 µM of primers, and 2.25 U of Expand Long Template Enzyme mix. The thermocycler program used is as follows: (1) 94°C for 2:00, (2) 94°C for 0:10, (3) 55°C-60°C for 0:30, (4) 68°C for 1:30, (5) steps 2 through 4 repeated 10 times, (6) 95°C for 0:15, (7) 55°C-60°C for 0:30, (8) 68°C for 1:30+0:20/cycle, (9) steps 6 through 8 repeated 25 times, and (10) 68°C for 7:00. All primer sequences can be found in the Supplementary Note Table 1.

Supplementary Note Table 1. Primers used in PCR validation of processed pseudogenes.

Chromosome	Start	End	Size	Sample ID	Gene	Forward	Reverse
chr11	123625829	123640235	14406	NA18517	*OR8G1*	TTGCAGCCATCTTCAATCA	TCTGCTGCCATTCTTTGATG
chr12	11311591	11311718	127	NA12891	*PRB3*	TGATTACTGGGGAGGCTGTC	TGTCAGCCAGGAAGAATCTC
chr12	54956248	54962472	6224	NA12892	*CS*	TTGCTGCAACACAAGGTAGC	CAAAAGAGTGGGCAAAGAGG
chr12	62465118	62482136	17018	NA19238	*TMEM5*	CAGCGATGTGACTGCTCAAT	TCATTAATCCAGGGGCTGTC
chr13	20632128	20633930	1802	NA18956	*C13orf3*	GATGGAATTTTCAAACCAGGAG	CAGAATCCAGGCTCAATGAT
chr13	20640541	20644480	3939	NA15510	*C13ofr3*	CCTGTGGAGGGTTTGGTAGA	TGGAAAATCAAGAAGGCATTG
chr18	74957635	74971340	13705	NA19238	*ATP9B*	GAGGATGAGTCTGCGCATTT	TTCAGGACATCCAAGCCATA
chr19	1562849	1566284	3435	NA12878	*TCF3*	GCTTTGTCCGACTTGAGGTG	AGACGAGGACGAGGACGAC
chr2	179014678	179016240	1562	NA18956	*PRKRA/PACT*	TTCCTTTTGGCTTGCTTTTT	TGGCTGGAGACTTCCTGAAT
chr2	227902745	227903587	842	NA19238	*MFF*	GGAAAAGCAGTGTCCGTGTT	TGGAATCCTTGTTCCAGGTC
chr20	3684250	3687183	2933	NA18507	*C20orf27*	GACATCCTTGCTCAGCCTGT	GAGTCCGGAGTATCCGCTTT
chr4	12979377	12987271	7894	NA18555	*RAB28*	TTGATTTTCTTCTTCCGGGTA	GTTGCTGCTGAAATCCTTGG
chr5	115258757	115266483	7726	NA12891	*AP3S1*	TGAAAATGTCTGTGAGCTGGA	TTTCCAGCTTATTTTGTGCATC
chr5	172967904	172968849	945	NA19238	*FAM44B*	CCCTGGGTTGCTGTAGTGTT	CTCCAGCTCCATCTCAGGAC
chr5	172969042	172972738	3696	NA18517	*FAM44B*	GTCCTGAGATGGAGCTGGAG	ATCTGGACAAGCAGGAATGG
chr6	31713081	31713202	121	NA18956	*BAT2*	CACGCCTTCCACCTACAGTG	GTAGGGGGCAAGAGGAACTC
chr6	136632446	136634828	2382	NA18555	*BCLAF1*	TGACCACCTTCTTCCAATGTC	GACAGCCTCCCCAGTAATCA
chr7	23318993	23319666	673	NA12891	*IGF2BP3*	CATCAGGTGTCTGGTCACGA	ATCAGAGTGCCATCCTTTGC
chr7	44840688	44841654	966	NA12892	*H2AFV*	GGCAAGCATAGAAGTGACCAG	GCTCAGGGAAGAATTTATGGAA
chr8	30040974	30043048	2074	NA12891	*TMEM66*	CTTTCTACTTTATCGTCTCCTGGT	GGGCTTACTCACCCCTTCAT
chr8	30043198	30043835	637	NA12891	*TMEM66*	CCTCCATGAAGGGGTGAGTA	TGGTGCAACTTCTGGTTTTG
chr9	19060300	19066601	6301	NA19238	*FAM29A*	CACTGTCTCCTCTGCAACCA	TTTGATCCTGCCTCAGAAGAA
chr9	107496875	107507699	10824	NA15510	*TMEM38B*	GCCCTCTCCTACTCCTCACC	AGGATTCTTCCATGCCAATG
chr9	107523826	107524638	812	NA15510	*TMEM38B*	CAACTACTGGCTTCGGGAAT	AGCCATTCATCACCTTCTGG
chrX	56312648	56313340	692	NA19240	*KLF8*	AAAGTTGACCCCACCTCCAT	ATTCTGCGGTGAGCTTTCAG

REFERENCES

1. Eichler, E.E. Widening the spectrum of human genetic variation. *Nature genetics* **38,** 9-11 (2006).
2. Mills, R.E. *et al.* An initial map of insertion and deletion (INDEL) variation in the human genome. *Genome research* **16,** 1182-1190 (2006).
3. Kidd, J.M. *et al.* Mapping and sequencing of structural variation from eight human genomes. *Nature* **453,** 56-64 (2008).
4. Conrad, D.F. *et al.* Origins and functional impact of copy number variation in the human genome. *Nature* **464,** 704-712 (2010).
5. Mills, R.E. *et al.* Mapping copy number variation at fine scale by population scale genome sequencing. *Nature* **470,** 59-65 (2011).
6. Alkan, C., Coe, B.P. & Eichler, E.E. Genome structural variation discovery and genotyping. *Nature reviews* **12,** 363-376 (2011).
7. Ye, K., Schulz, M.H., Long, Q., Apweiler, R. & Ning, Z. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics (Oxford, England)* **25,** 2865-2871 (2009).
8. Tuzun, E. *et al.* Fine-scale structural variation of the human genome. *Nature genetics* **37,** 727-732 (2005).
9. Kidd, J.M. *et al.* Characterization of missing human genome sequences and copy-number polymorphic insertions. *Nature methods* **7,** 365-371 (2010).
10. Hajirasouliha, I. *et al.* Detection and characterization of novel sequence insertions using paired-end next-generation sequencing. *Bioinformatics (Oxford, England)* **26,** 1277-1283 (2010).
11. Hach, F. *et al.* mrsFAST: a cache-oblivious algorithm for short-read mapping. *Nature methods* **7,** 576-577 (2010).
12. Hamming, R.W. Error-detecting and error-correcting codes. *Bell System Technical Journal* **29,** 147-160 (1950).
13. Chvatal, V. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research* **4,** 233-235 (1979).
14. Hormozdiari, F. *et al.* Next-generation VariationHunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics (Oxford, England)* **26,** i350-357 (2010).
15. 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. *Nature* **467,** 1061–1073 (2010).
16. O'Roak, B.J. *et al.* Exome sequencing in sporadic autism spectrum disorders identifies severe de novo mutations. *Nature genetics* **43,** 585-589 (2011).
17. Renton, A.E. *et al.* A Hexanucleotide Repeat Expansion in C9ORF72 Is the Cause of Chromosome 9p21-Linked ALS-FTD. *Neuron* (2011).
18. Depristo, M.A. *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature genetics* **43,** 491-498 (2011).
19. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)* **25,** 1754-1760 (2009).
20. Wang, J. *et al.* CREST maps somatic structural variation in cancer genomes with base-pair resolution. *Nature methods* **8,** 652-654 (2011).
21. Hormozdiari, F., Alkan, C., Eichler, E.E. & Sahinalp, S.C. Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome research* **19,** 1270-1278 (2009).

22. Chen, K. *et al*. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nature methods* **6,** 677-681 (2009).
23. Lee, S., Hormozdiari, F., Alkan, C. & Brudno, M. MoDIL: detecting small indels from clone-end sequencing with mixtures of distributions. *Nature methods* **6,** 473-474 (2009).