Supplementary Methods, Supplementary Figures 1-12 and Supplementary Note

INTEGRATE: Gene fusion discovery using whole genome and transcriptome data

Jin Zhang[1,2], Nicole M. White[2], Heather K. Schmidt[1], Robert S. Fulton[1,3], Chad Tomlinson[1], Wesley C Warren[1], Richard K. Wilson[1,3], and Christopher A. Maher[1,2,4,5*]

[1]The McDonnell Genome Institute, Washington University School of Medicine, St. Louis, MO, 63110, USA.

[2]Department of Internal Medicine, Division of Oncology, Washington University School of Medicine, St. Louis, MO, 63110, USA.

[3]Department of Genetics, Washington University School of Medicine, St. Louis, MO, 63110, USA.

[4]Alvin J. Siteman Cancer Center, Washington University School of Medicine, St. Louis, MO, 63110, USA.

[5]Department of Biomedical Engineering, Washington University School of Medicine, St. Louis, MO, 63110, USA.

*Correspondence and requests for materials should be addressed to C.A.M. (email: cmaher@dom.wustl.edu).

Supplementary Methods

HCC1395 whole genome and transcriptome sequencing data

Cells were purchased from American Type Culture Collection (ATCC, Manassas, VA). The cells were grown at 37°C in 95% O_2–5% CO_2. HCC1395BL cells were cultured in Iscove's Modified Dulbecco's Medium (IMDM) with 20% fetal bovine serum (FBS), and 1% penicillin/streptomycin (P/S). HCC1395 cells were cultured in RPMI with 10% FBS, and 1% P/S. Cells were minimally passaged from time of purchase to reach desired cell numbers and genomic DNA and RNA was isolated from cells of the same passage. RNA was isolated using RNeasy Mini Kit (Qiagen, Valencia, CA) following the manufacturer's instructions with the recommended on column DNase I (Qiagen) digestion. Genomic DNA was isolated with the DNase Blood and Tissue Kit (Qiagen) with a RNase A digestion (40ug/uL). All RNA and DNA were eluted in water. For each Whole Genome Shotgun library, 350ng DNA was fragmented in 5X DNATerminator End Repair Buffer (Lucigen, Middleton, WI) using the Covaris S2 and micro-TUBEs (Covaris, Woburn, MA; using the following settings: volume = 50µL, temperature = 4°C, duty cycle = 5, intensity = 4, cycle burst = 200, time = 90 seconds. The fragmented ends were converted to blunt ends by adding DNATerminator End Repair Enzyme and following the manufacturer's protocol. The blunt ended DNA was then purified using MinElute columns per manufacturers protocol (Qiagen). DNA was eluted with 32 µl 10mM Tris-HCl (pH 8.0). A 3' A overhang was added to the blunt ended fragments by treating with 15 units of Klenow Fragment 3'->5' exo- and 200nM dNTP mix (New England BioLabs, Ipswich, MA) for 30 min at 37°C. Five and three lanes of whole genome HiSeq 2000 (v3 chemistry; Illumina) sequence data consisting of 2x100bp reads data were produced for HCC1395 and HCC1395BL, respectively. The two cell lines were whole genome sequenced, as previously described (Mardis et al. 2009). A single lane of 2x100bp RNA-seq data was

generated, as previously described (Govindan et al. 2012), for HCC1395 and HCC1395BL transcriptomes.

The direction of the read corresponding to the 5' gene is in the same orientation as the 5' gene whereas the orientation of the read aligning to the 3' gene is in the opposite direction to the 3' gene. The weight of the edge correlates with the total number of encompassing reads supporting the gene fusion. Since a graph of putative fusions is more compact than a list of gene fusions, many subsequent operations, such as removing some edges (hence candidate gene fusions), can be performed efficiently on a graph. For instance, if one node or gene is connected to many nodes or genes, then aligning spanning reads to this node is only performed once. At this point the graph can be very dense and INTEGRATE uses a series of filtering steps to remove false positive gene fusion candidates according to the concordant suboptimal alignments and repetitiveness of the paired-end reads in the graph (See Supplementary Methods).

INTEGRATE software

The above methods and algorithms are implemented in the INTEGRATE software following the standards of the C++ Programming Language. An external library, Libdivsufsort (https://code.google.com/p/libdivsufsort), is applied to construct the Burrows–Wheeler transforms of the gene node sequences, and the APIs of SAMtools (http://samtools.sourceforge.net) are called to get aligned reads from the BAM files. The INTEGRATE software can be downloaded from SourceForge (https://sourceforge.net/projects/integrate-fusion/) and the source code is available in the Supplemental Material.

The input files required to run the INTEGRATE software include a text file of gene annotations that can be downloaded from the UCSC Genome Browser (Kent et al. 2002),

mapped and unmapped RNA-seq reads in BAM format, mapped tumor (and normal if applicable) WGS reads with 'soft-clipping' (unaligned region is not trimmed off) in BAM format, the reference genome in FASTA format and its pre-built Burrows-Wheeler transform. INTEGRATE reconstructs the exact fusion junctions (i.e., giving coordinates) of a gene fusion from spanning RNA-seq reads. INTEGRATE can detect multiple transcript isoforms for a fusion gene as well as isoforms utilizing canonical and non-canonical exonic boundaries.

For each predicted gene fusion INTEGRATE reports the exact fusion junctions and genomic breakpoints in bedpe and VCF format. Detailed information of a fusion is also provided, i.e., number of encompassing and spanning RNA-seq and WGS reads, sequences of reads and their mapped coordinates, lengths of the split segments of the spanning reads, whether a fusion is due to a reciprocal translocation, all possible fusion transcript isoforms, and type of fusion (inter-chromosomal, intra-chromosomal, and read-through). For fusions with canonical exonic boundaries, INTEGRATE reports the transcripts, exons and their sequences in addition to the coordinates of the fusion junctions.

As part of the INTEGRATE workflow we have chosen to process the tumor RNA-seq data before the WGS data. Starting with WGS data, as done by some approaches (i.e., BreakTrans) may appear appealing since usually the mapping quality of WGS data is better than RNA-seq data (i.e., percentage of reads mapped, uniqueness the of mappings). However, our observation is that many genomic events may associate with repetitive reads, while expressed fusion transcripts may involve exons that are quite unique. Therefore, RNA-seq reads could lead to higher sensitivity in nominating candidate fusions compared to WGS reads. From the aspect of combining data, analyzing more repetitive WGS reads can be much easier in local regions adjacent to fusion junctions discovered by RNA-seq reads. A third reason is that a genomic event does not necessarily produce a gene fusion. Effort to

find expressed gene fusions in RNA-seq data for these SVs would not yield any true positive gene fusions but would take time for an algorithm to analyze. Furthermore, RNA only chimeras, such as transplicings and read-throughs, would not have any WGS reads and therefore would be filtered despite having supporting RNA-seq reads. For these reasons, INTEGRATE starts with RNA-seq reads to nominate fusion candidates followed by examining their genomic breakpoints.

Filtering false positive encompassing RNA-seq reads

The encompassing reads reported by reads mapping tools contain large amount of false positives. For example, using the HCC1395 data, INTEGRATE creates an initial graph with 23,668 nodes and 124,381 edges using TopHat2 alignments with Ensembl genes. INTEGRATE removes false positive gene fusion candidates from paired-end reads that are considered to be discordant when they actually correspond to a suboptimal alignment that results in a concordant pairing. Since INTEGRATE generates edges within the graph as it encounters discordant mappings during its first pass through a BAM file it is possible that a subsequent mapping that appears later in the BAM file may reveal a concordant mapping. Therefore, INTEGRATE reduces these potential false positives by removing encompassing reads, used as edges, that have a concordant mapping during a subsequent iteration through the BAM file. Each concordant read is checked in a hash table TI1 to determine whether it had been used as an encompassing read in the graph. If so, using the list of ids kept in TI1 the indexes are removed from the edges (each edge that the read contributed to can be found from TR1 (Step 2 in Supplementary Figure 9)). If an edge does not have any TR1 indexes, then the edge is entirely removed. Depending on the alignment tool used it is possible that suboptimal concordant alignments may not have been detected and therefore the discordant reads were not removed in the previous step. Therefore, as a control,

INTEGRATE includes a step where it attempts to realign each encompassing read to both nodes connected to the edge. Given that this is a targeted alignment INTEGRATE allows a higher error rate (i.e., 2 mismatches or gaps in one flanking region) to increase the sensitivity. If both reads can be aligned to a single node then the contribution of the encompassing read is removed from the edge.

INTEGRATE was implemented so that an encompassing read with multiple mappings can contribute to more than one edge on the graph. INTEGRATE assigns a weighted value to each edge corresponding to the sum of the scores of the encompassing reads. INTEGRATE first assigns a weighted score of one to a unique encompassing read alignment, and each occurrence of the non-unique encompassing read alignment is given a weighted value equivalent to one over the number of alignments. A user can define the minimum weight cutoff value. The default cutoff value is 2.0 thereby ensuring two independent reads support an event. Edges with weights less than the minimum value are removed from the graph.

Depending on the alignment tool used, it is possible that not all non-unique encompassing read alignments are reported. Therefore, while the reported alignments of repetitive reads may have contributed to many edges in the graph, the true occurrences across the genome could be even greater. Instead of realigning these repetitive reads (which is not likely to yield gene fusions but will require a longer run time), INTEGRATE estimates the repetitiveness of these reads by focusing on the smallest occurrence out of the four segments of 25-mers (for a 100-mer read) from the 5' and 3' end of each read in a pair. The 25-mers are aligned to the BWT of the whole genome to get the number of exact alignments. For non-repetitive read pairs, the value of occurrence is usually 1 (0 if all the four 25-mers contains errors). For extremely repetitive reads the value of occurrence can be very large. A new weight, one over occurrence, is assigned to each encompassing read when the value of

occurrence is larger than 1. The weights of the edges in the gene graph are then updated, and the graph is reduced with the same cutoff value (default 2.0).

Aligning unmapped reads as spanning RNA-seq reads

INTEGRATE retrieves read pairs that have one aligned read and one unaligned read. Using the aligned read as an anchor, INTEGRATE conducts a targeted alignment to the node corresponding to the mapped gene, or 'anchor node', and all nodes that it is connected to in the graph, or 'neighboring node(s)'. Since most of the previously unaligned reads are not likely to be real spanning reads from gene fusions and by chance large false positives may be among the reads aligned as spanning reads, INTEGRATE also tries to find the suboptimal concordant alignments of these reads.

INTEGRATE first iterates through the BAM file of aligned reads and collects information about the reads with only one read aligning to a gene(s) in the graph. The alignment of these records, and their corresponding genes, are stored in table TR2 (Supplementary Figure 9). TI2 is used to keep a map of the anchor read and its corresponding alignments from table TR2 (Supplementary Figure 9). Next, to identify fusion junction spanning RNA-seq reads INTEGRATE uses the fusion graph to first check if the unaligned read can map to the anchor gene (with less stringent parameters than the initial alignment) thereby representing a concordant pair. This is done by aligning the 3' of the unaligned read to the anchor gene node downstream of the anchor read and on the reverse strand. If the 3' of the unmapped read cannot be aligned INTEGRATE proceeds to the next pair. If the 3' of the unmapped read is mapped, then INTEGRATE tries to map the 5' portion of the unmapped read to the same gene (to test for concordance). If both 3' and 5' can be aligned to the same gene, then this is categorized as a concordant pair and all records of this read are removed. This remove operation is straightforward, since the records are

consecutive at the bottom of TR3 (the table to store mapped spanning RNA-seq reads). If only the 3' portion of the unaligned read has been mapped, then INTEGRATE proceeds with the neighboring nodes and aligns the 5' portion of the unmapped read. If the 5' portion of the unmapped read can be aligned to a neighboring node then INTEGRATE tries to align the 3' portion of the unaligned read to the neighboring node to evaluate whether it is an encompassing read pair. If the 3' of the unmapped read cannot be mapped to the neighboring node but can only be mapped to the anchor gene then INTEGRATE classifies it as a spanning read and adds a record to TR3 (Step 6 of Supplementary Figure 9).

Efficient split-read RNA-seq reads mapping on gene node

Since the length of the current short paired-end reads is still very short (typically less or equal to 150 bps), the current version of INTEGRATE intends to align a split RNA-seq read into two segments. After the fusion graph has been created, BWTs of the gene node sequence (includes exons and introns; Supplementary Figure 1) and its reverse complement are created and suffix array indexes are kept every 32 locations. Then the 3' end of an RNA-seq read can be mapped to the forward strand of some genes and the 5' can be mapped to the reverse complement strand of these same genes. Supplementary Figure 1d shows the prefix trie of the sequence (text) of `GCCGCT'. When the BWT of this sequence has been computed, the prefix trie can be simulated by searching (mapping on BWT) A, C, G and T at each node of the tree. The 3' of a sequence (pattern) can be mapped to the prefix trie. For example, the pattern `CCAC' can be mapped to the path marked as red in the prefix trie. Only tree nodes relevant to the sequence are generated, i.e., tree nodes at the top of the tree and tree nodes along the path where the pattern could be mapped. Supplementary Figure 1e shows the nodes of the tree that are needed when mapping the 3' of 'TTTTCCAC' to the tree with at most one error (only the portion of `CCAC' can be mapped). The nodes

are shown in a line, which correspond to the column of a matrix to store the scores of dynamic programming mapping. The rows of the matrix correspond to the sequence to be mapped. The links are the same as in Supplementary Figure 1d. This is different from the classical Smith-Waterman local alignment algorithm, which only allows item (i,j) in the matrix to come from (i-1,j-1), (i-1,j), or (i,j-1), as an item in the matrix in Supplementary Figure 1e does not necessarily come from the consecutive left column. Instead, (i,j) can come from the column that corresponds to the parent node of the node associated with the current column, i.e., the columns are in the order defined by the subtree. However, an item at row i can come from row i or i-1, which is the same as classical Smith-Waterman local alignment algorithm. Split-read mapping of INTEGRATE is performed as mapping a portion of a read until no more errors are allowed. Since only a certain number of errors are allowed, only a subset of nodes from the tree are needed thereby eliminating the need to compute all the items in the matrix. When a path in the tree can only be mapped using partial reads (with more errors than allowed) then the path dies, hence the nodes in the subtree of the path are not generated. Only item at row r-x, r, and r+x for the $l-r+1^{th}$ character of the sequence are mapped, where l is the length of the sequence, and x is the maximum number of errors allowed. Also, not all the 2x+1 items in a column are used. If an item in the matrix can only be reached by introducing more errors than allowed, then it is not reached. The pseudo code is provided in Supplementary Figure 11. When allowing a small number of errors (one or two mismatches or gaps), and when the sequence of the gene is not very repetitive, it only takes less than several hundred operations to map a read, thus mapping one read using this algorithm is very efficient. Additionally, the alignment of a read to a gene where it did not originate will stop quickly since it only has to align to a smaller space of the gene node.

While the classical Smith Waterman local alignment algorithm, which takes O(nm) time, where n is the length of the read and m is the length of the reference (here, gene node), this algorithm takes O(nk) time, where k is the tree nodes simulated. Since the paths of the sub tree are all sequences within allowed differences compared to the read, and these paths can be scattered on the reference, and the shared prefixes of these paths are only compared with the read once. For a gene node containing exons and introns, the length of m can be as high as millions of bases, while k can be as low as hundreds of bases.

Calling gene fusions by clustering spanning RNA-seq reads

Once INTEGRATE identifies spanning reads they are clustered together if multiple spanning reads support the same fusion junction. A single gene fusion can have one or more clusters of spanning RNA-seq reads corresponding to multiple fusion transcript isoforms. By default, if at least one cluster of canonical exonic boundaries with at least one spanning read exists, then INTEGRATE reports a gene fusion. INTEGRATE allows the user to set the threshold of reads needed for reporting a cluster of non-canonical exonic boundaries (default = 3). Just as the encompassing reads were given weighted values to account for multiple alignments, spanning reads mapping to multiple clusters are also weighted. While using a high cut off value can select top candidates, it is possible for true positive gene fusion candidates to be removed in the process. However, being very aggressive for sensitivity in this category may lead to an extremely large set that is not practical to work with, e.g., nFuse reported ~25 thousand fusion candidates in this category in HCC1395 cells.

Finding exact genomic breakpoints supporting gene fusions using WGS data

While existing programs such as BreakTrans and nFuse search for SVs in the entire genome to call genomic breakpoints that support gene fusions, one obvious advantage of

INTEGRATE is that subsequent computational resources reconstructing the fusion junction are only expended for the subset of SVs capable of producing an expressed gene fusion. Since INTEGRATE places an emphasis on detecting expressed gene fusions, it only aligns WGS reads in close proximity to the predicted fusion junctions. Therefore, INTEGRATE assumes that for an SV to potentially generate a gene fusion transcript that the genomic breakpoints reside within a certain distance from the gene (default = 50,000 bps), referred to as the extended gene coordinates.

Since spanning WGS reads can be very repetitive they present a challenging and time consuming process to align to the whole genome. In contrast, INTEGRATE targets the extended gene coordinates therefore repetitive reads will have fewer mappings than they would against the whole genome. For the 5' gene fusion partner the upstream boundary of the extended gene coordinate region is equal to the maximum insert size in length, plus three standard deviations, upstream of the fusion junction whereas the downstream boundary is 50,000 bps downstream of the fusion junction. The extended gene coordinate region for encompassing WGS reads aligning near the 3' fusion partner is defined similarly (Supplementary Figure 10). Encompassing WGS reads aligning to these regions are retrieved from the aligned WGS BAM file. To be an encompassing WGS read, INTEGRATE further requires that (i) a discordant pair has the two reads map within the extended gene coordinates of the 5' and 3' genes respectively (ii) the read within the extended gene coordinates of the 5' gene should have the same strand as the 5' gene whereas the read within the extended gene coordinates of the 3' gene should be in the opposite orientation as the 3' gene. The encompassing WGS reads and the fusion junctions are then used to guide the alignment of spanning WGS reads. For example, at the 5' gene, the genomic breakpoint should be downstream of the fusion junction, and within a maximum insert size downstream of an encompassing WGS read, which we refer to as a 'focal region' (Focal

Region 1 in Supplementary Figure 10). WGS reads are mapped to identify spanning WGS reads with two segments mapped to the two focal regions established by the encompassing reads respectively (Supplementary Figure 10). INTEGRATE uses a semi-global alignment algorithm to align the soft-clipped segment of the spanning WGS reads. By aligning to a smaller focal region INTEGRATE can use less stringent alignment criteria to increase the power of reconstructing genomic breakpoints.

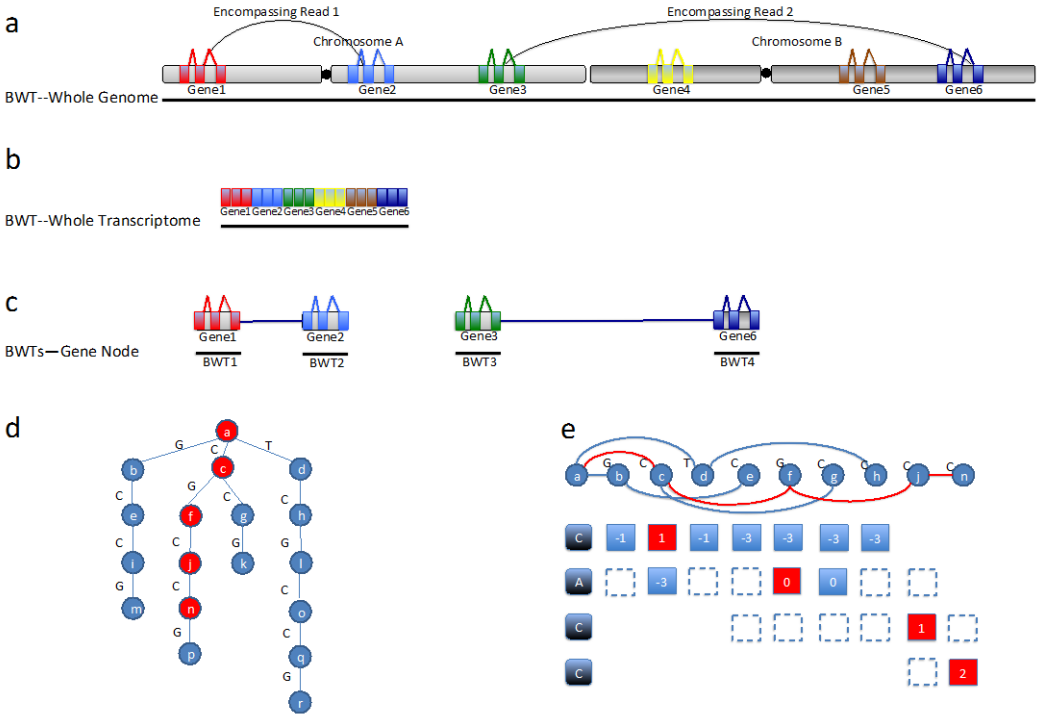
Reference:

Govindan R, Ding L, Griffith M, Subramanian J, Dees ND, Kanchi KL, Maher CA, Fulton R, Fulton L, Wallis J et al. 2012. Genomic Landscape of Non-Small Cell Lung Cancer in Smokers and Never-Smokers. *Cell* **150**: 1121-1134.

Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. 2002. The human genome browser at UCSC. *Genome research* **12**: 996-1006.

Mardis ER, Ding L, Dooling DJ, Larson DE, McLellan MD, Chen K, Koboldt DC, Fulton RS, Delehaunty KD, McGrath SD et al. 2009. Recurring Mutations Found by Sequencing an Acute Myeloid Leukemia Genome. *New Engl J Med* **361**: 1058-1066.

Supplementary Figures 1-12



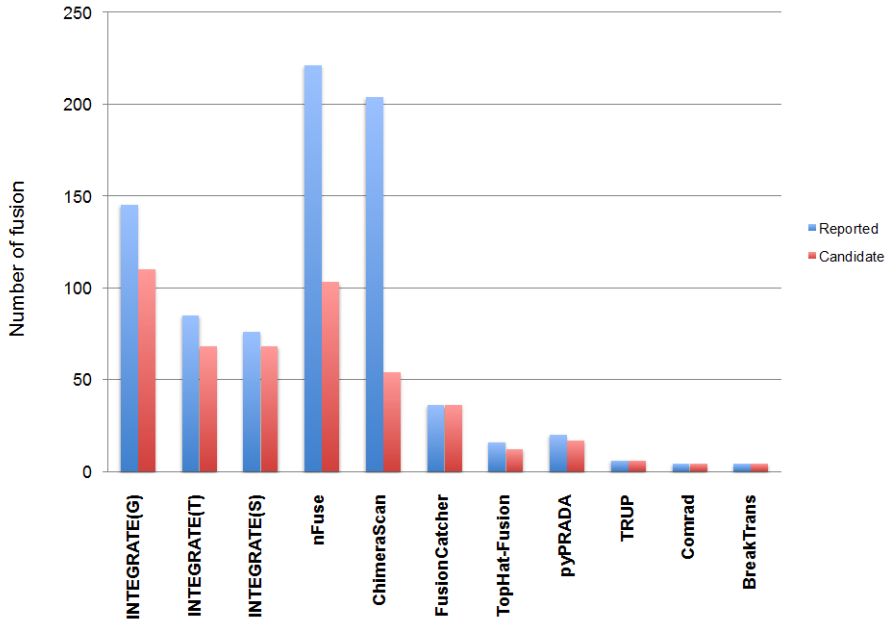
Supplementary Figure 1. Gene node and reads mapping on their BWTs. A Gene Node contains both exons and introns for a gene. A BWT can be constructed for each Gene Node. Compared with using the BWT for whole genome and the BWT for whole transcriptome, mapping on BWTs of gene node have several advantages. (1) Only the subset of genes that are implicated as potential fusions are considered. (2) True positive split reads can be more uniquely mapped to one BWT for a gene node. (3) The alignment algorithm stops faster when mapping false positive split reads. (4) Both canonical and non-canonical junctions can be searched. **(a)** The BWT for whole genome. **(b)** The BWT for whole transcriptome. **(c)** BWTs for gene nodes in a gene network. **(d)** A Prefix trie for sequence "GCCGCT". Prefix trie of a gene node can be simulated on its BWT. The red nodes constitute the path for mapping suffix "CCAC". **(e)** Actual nodes visited when searching for "CCAC" with no more than 1 error and values for dynamic programming when setting the substitute penalty (-1) and gap (-3). Only nodes at the top of the trie for are visited.

Tier	Canonical	EN RNA T	SP RNA T	EN WGS T	SP WGS T	EN & SP WGS N
1	Y	Y	Y	Y	Y	N
2		Y	Y	Y	N	
3		Y	Y	N	N	
4	N	Y	Y	Y	Y	
5		Y	Y	Y	N	
6		Y	Y	N	N	

Canonical	With Canonical Exonic Boundaries
EN RNA T	Encompassing Tumor RNA-Seq Reads
SP RNA T	Spanning Tumor RNA-Seq Reads
EN WGS T	Encompassing Tumor WGS Reads
SP WGS T	Spanning Tumor WGS Read
EN WGS N	Encompassing Tumor WGS Reads
SP WGS N	Spanning Tumor WGS Reads

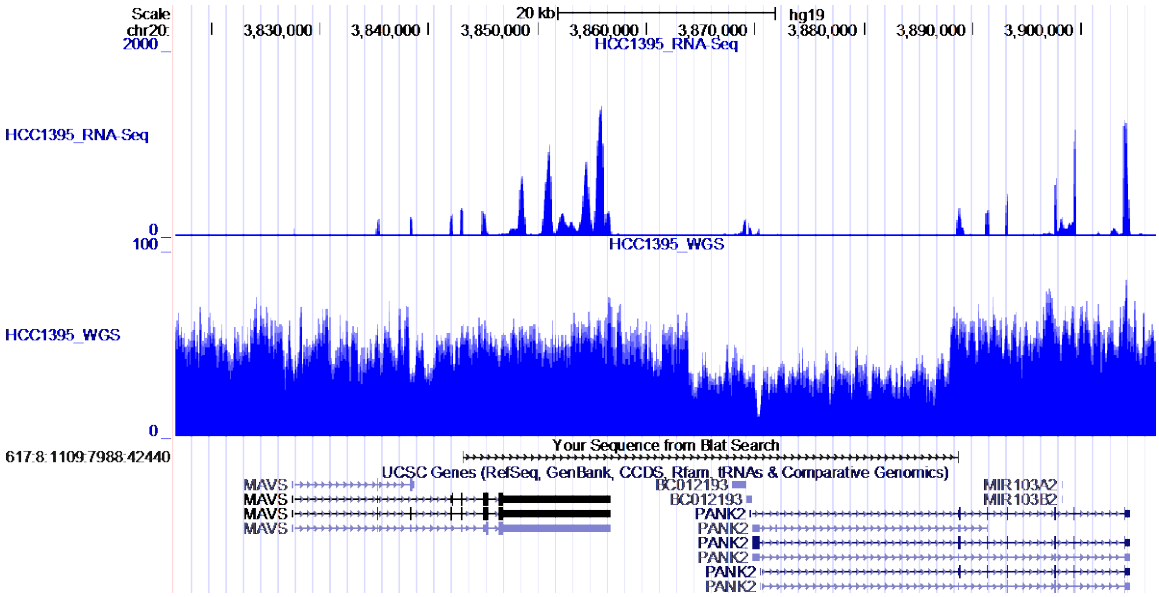
Available data	Report Tiers
RNA T WGS T WGS N	1-6
RNA T WGS T	1-6
RNA T	3,5

Supplementary Figure 2. Definition of fusion Tiers in INTEGRATE output. The gene fusion candidates called by INTEGRATE are categorized into Tiers. Gene fusions with canonical exonic boundaries are included in Tiers 1-3 whereas gene fusions without canonical exonic boundaries are included in Tier 4-6. Tiers 1 and 3 have encompassing and spanning RNA-seq and WGS reads. Tiers 2 and 4 lack spanning WGS reads. Tier 3 and 6 only have encompassing and spanning RNA-seq reads. When only RNA-seq data is available INTEGRATE reports gene fusions with and without canonical exonic boundaries as Tier 3 and 5, respectively.

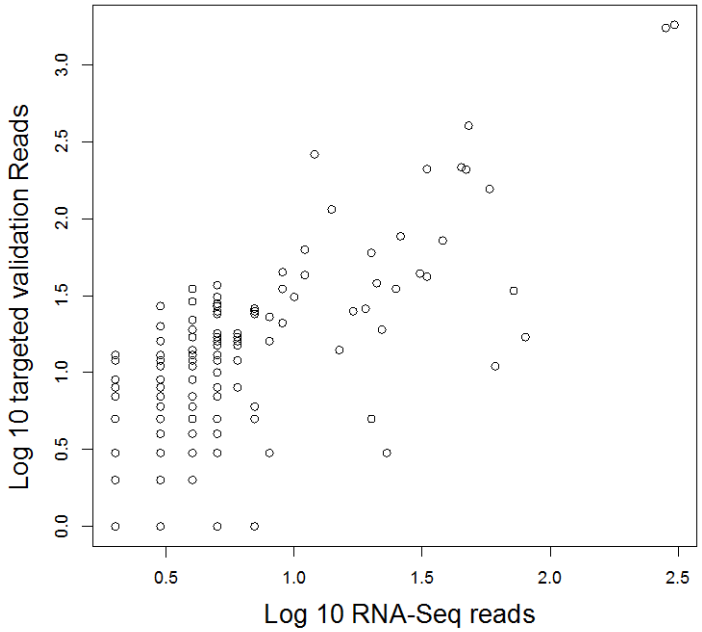


Supplementary Figure 3. The quantity of fusions with canonical exonic boundaries reported by 9 gene fusion detection tools and the quantity of candidate gene fusions attempted by validation. INTEGRATE is run using alignments from GSNAP (G), TopHat2(T), and STAR(S) separately with default parameters. BreakTrans is supplied with alignments from BreakDancer. All tools provide coordinates of predicted fusion junctions except Comrad and BreakTrans. For Comrad, 4 gene fusions with predicted junctions within 15bps of exons are considered as with canonical boundaries. These 4 are also called by other methods. For BreakTrans, all 4 called fusions are all called by other methods, thus included.

Supplementary Figure 4. Coverage map view for *KCNQ5* and *RIMS1* of HCC1395 cell line in UCSC Genome Browser. *RIMS1* and *KCNQ5* are on the forward strand of region 6q13. *KCNQ5* is downstream of *RIMS1*. Vertical viewing ranges are set to 500 (quantity of reads) for WGS and 100 (quantity of reads) for RNA-seq. This view highlights the lack of *RIMS1* expression thus supporting the lack of encompassing or spanning RNA-seq reads supporting a fusion between the two genes in our RNA-seq data.



Supplementary Figure 5. A highly expressed gene fusion discovered only by INTEGRATE. *MAVS-PANK2* has 38 RNA-seq reads in our HCC1395 data sets. The coverage map of WGS data shows a deletion between *MAVS* and *PANK2* which may cause the fusion. Blat search results of a spanning RNA-seq read shows the fusion junction.

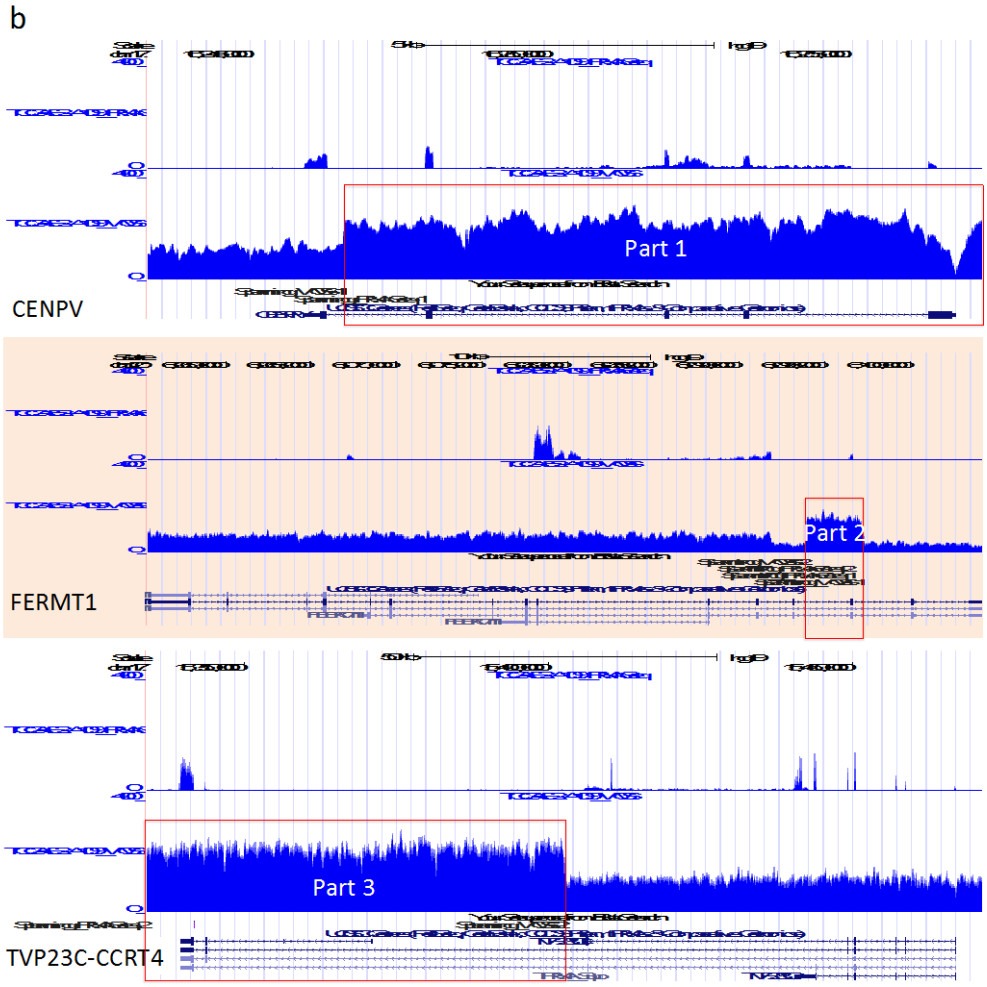


Supplementary Figure 6. Correlation of read support for gene fusions between RNA-seq and targeted validation. Scatterplot of the log transformed total encompassing and spanning reads generated from targeted validation and RNA-seq for 138 HCC1395 gene fusions. Scatterplot highlights the consistency of expression levels between the two data sets (correlation=0.95). The lower expressed gene fusions typically have very few supporting sequence reads and therefore require very sensitive gene fusion calling tools to be detected.

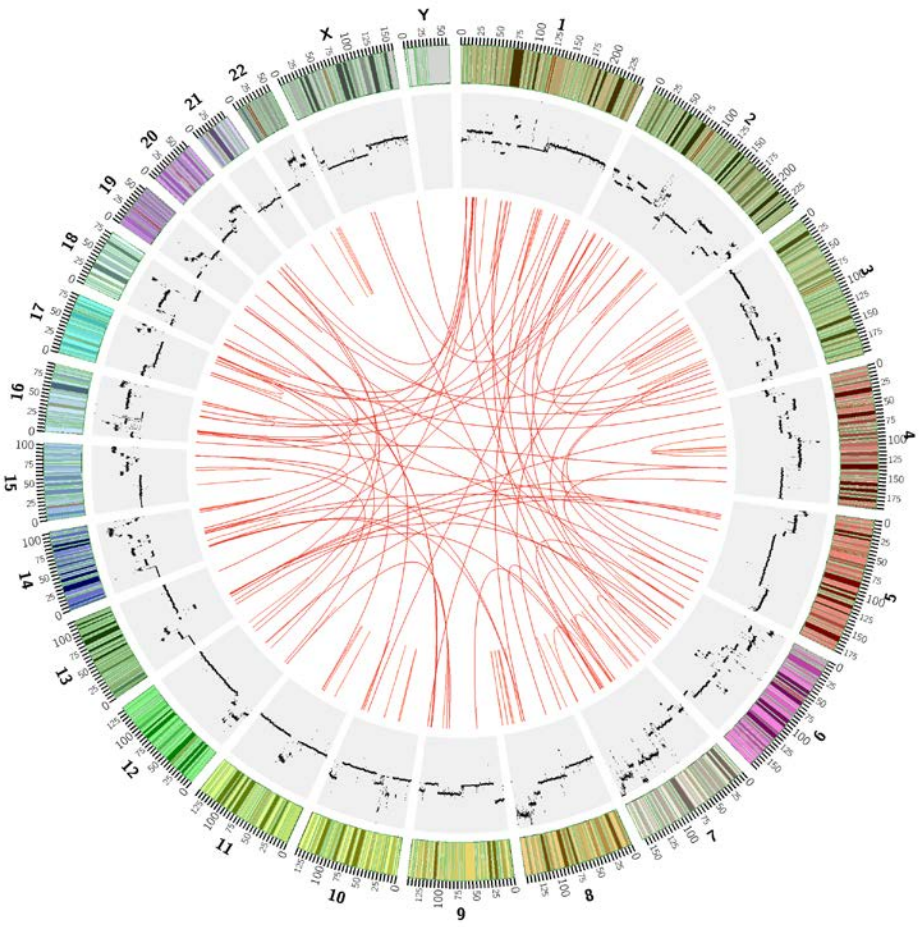
a

Gene1	Gene2	Gene3	Chr1	Pos1	Chr2	Pos2	Chr3	Pos3	Chr4	pos4	sample
DCAF7	VMP1	RAD51C	17 +	61,628,176	17 +	57,808,782	17 +	57,816,308	17 +	56,787,220	BH-A18R
BCAS3	RNF169	CADM2-AS2	17 +	58,786,686	11 +	74,521,229	11 +	74,528,759	3 -	85,850,101	BH-A18R
ACER3	UVRAG	PAK1	11 +	76,730,820	11 +	75,826,968	11 +	75,827,059	11 -	77,034,405	E2-A109
CENPV	FERMT1	TVP23C-CDRT4	17 -	16,247,911	20 -	6,096,691	20 -	6,096,691	17 -	15,341,517	E2-A109
FXN	SMC5	IMMP2L	9 +	71,679,951	9 +	72,879,220	9 +	72,882,891	7 -	110,603,621	AO-A124
RARA	NARS2	PLXDC1	17 +	38,487,648	11 -	78,282,489	11 -	78,239,888	17 -	37,296,085	A1-A0SM
CHD4	VWF	LOC338817	12 -	6,715,440	12 -	6,184,717	12 -	6,155,889	12+	11,706,500	B6-A0RE

b



Supplementary Figure 7. Chains of gene fusions discovered by INTEGRATE from 62 TCGA breast cancer patients. (a) Genes for chains and the coordinates of fusion junctions. **(b)** *Chain CENPV-FERMT1-TVP23C-CCRT4*. The first 4 exons of CENPV fuses with the 3rd exon of FERMT1, which is on a different chromosome, then fuses with the last exon of *TVP23C-CCRT4*. The coverage maps match very well with the WGS spanning reads, while the spanning RNA-seq reads show the exons involved.



Supplementary Figure 8. Copy number variations identified by CNVHMM in HCC1395 cell line and Circos plot of 138 validated gene fusions in HCC1395 cell line called by INTERGRATE and other 8 gene fusion detection methods. 223 amplicons (copy number ≥3 for tumor, and 2 for normal) are among the copy number variations regions, and 106 out of 138 (77%) gene fusions are having one or both genes in the amplicons. 143 (59%) out of 276 genes of the 138 validated fusions are in the amplicons, which percentage is significantly higher (Pearson's chi-squared test, p-value=2.35e-06) than the expressed genes in HCC1395 (6871 (38%) out of 18235).

Annotation of Genes and BAM of Mapped RNA-Seq Reads

1. Build Gene Fusion Graph (G) with Encompassing (e) RNA-Seq Reads

Gene Fusion Graph (G)

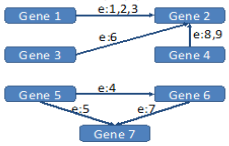


Table Indexing encompassing reads (TI1)

Hash value	En_ids
Hash(Read_A)	1
Hash(Read_B)	2
Hash(Read_C)	3,6
Hash(Read_D)	4,5
Hash(Read_E)	7
Hash(Read_F)	8
Hash(Read_G)	9

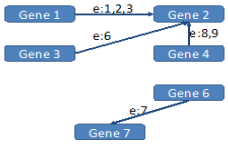
Table of encompassing reads (TR1)

En_id	Read Name	5p	3p
1	Read_A	1	2
2	Read_B	1	2
3	Read_C	1	2
4	Read_D	5	6
5	Read_D	5	7
6	Read_C	3	2
7	Read_E	6	7
8	Read_F	4	2
9	Read_G	4	2

Go through the BAM: for each pair of encompassing reads mapped to two genes, keep a record in TR1 (above), and keep the id in TI1 (left) and G.

2. Remove Encompassing RNA-Seq Reads that Have Been Mapped to One Gene

G after the edges between genes 5,6 and genes 5,7 removed

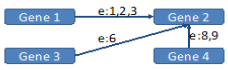


This step goes through the BAM and removes false positives according to reads mapped as normal pairs.

Suppose Read_D has also been mapped to Gene 8 not as encompassing reads but as normal alignments in the BAM, find ids 4 and 5 by name "Gene 8" using TI1, and find edges (5,6) and (5,7) by records 4 and 5 in TR1. Remove the reads on the edges, and hence the edges in this case.

3. Make Sure Encompassing RNA-Seq Reads on the Edges Can NOT Mapped to One Gene

G after the edge between genes (6,7) removed

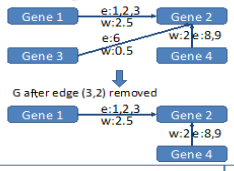


For each pair on an edge, try map them to one of the two nodes connect the edge, if mapped (Thus, normal reads but not encompassing reads.), remove the pair from the edge.

Suppose Read_E can be mapped to Gene 6, so remove it from the graph, and hence the edges in this case.

4.Remove Edges with Small Weights (w)

G after weights computed



Compute weights for each edge, remove the ones with small weights.

E.g. Read_C (records 3 and 6 in TR1) can be on two edges (1,2) and (3,2), contributing weight 0.5 to each. Remove the edge with weight less then 2 , which is edge (3,2). Hence remove the node of Gene 3 in this case.

5.Get Anchors (a) for Unmapped RNA-Seq Reads

G after adding anchors to the nodes

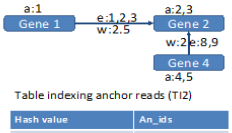


Table indexing anchor reads (TI2)

Hash value	An_ids
Hash(Read_H)	1,5
Hash(Read_I)	2
Hash(Read_J)	3
Hash(Read_K)	4

Table of anchor Reads (TR2)

An_id	Read Name	Gene_id
1	Read_H	1
2	Read_I	2
3	Read_J	2
4	Read_K	4
5	Read_H	4

For each gene node in the graph, find anchor reads, keep a record in TR2 (above), and keep the id in TI2 (left) and G.

BAM of Unmapped RNA-Seq Reads

6. Map the Unmapped RNA-Seq reads as Split Reads

For all the unmapped RNA reads:

Map unmapped read as split read to the gene with anchor. (⟶ to Gene 2)

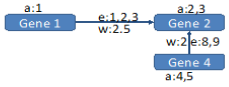
Note: Genes are found by looking at the table in step 5.

Is mapped?

Read_J:Unmapped — Read_J:Anchor at Gene2

Y

Map the other portion to the same gene (⟶ to Gene 2).



Is mapped?

Y

N

Map the other portion to the neighbor genes (⟶ to Gene 1 and 4).

Is mapped?

e.g. ⟶ Mapped to Gene 1 not Gene 4

Y

Map the portion with the anchor to the neighbors where the other portion mapped (⟶ to Gene 1).

Is mapped?

Y (encompassing)

N

Make a record of the split-read mapping

Remove all split-read alignments of the read

Note: Split reads that can be mapped to the same gene are not used by any fusion candidates.

7. Cluster (c) Spanning (s) RNA-Seq Reads and Annotate Types of Fusion

G after edge (4,2) removed because not having enough spanning RNA-seq reads

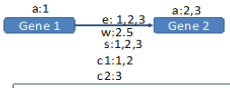


Table of Spanning RNA-seq reads (TR3)

Sp_id	Read Name	5p	3p
1	Read_H	1	2
2	Read_I	1	2
3	Read_J	1	2

e.g. Read H, I, and J are mapped to Gene 1 and 2. Read K is not mapped. Read H is not mapped to Gene 2 and 4.
Read H and I support one ISO form while Read J support another.

Types include:
Inter_Chromosomal, Intra_Chromosomal, and Read_Through

BAM of Tumor (and Normal) WGS Reads with soft-clip

8. Find WGS Encompassing Reads and Map WGS Spanning reads

For each Fusion candidate supported by RNA-Seq reads:

(1) Decide Regions Very Likely to Contain Encompassing WGS Reads Near Fusion Junctions.

(2) Find Encompassing WGS Reads Support RNA-Seq Fusion Junctions in the Above Regions from WGS BAM.

(3) Find WGS Reads Mapped with Soft-clip Near One read of a WGS Encompassing Pair.

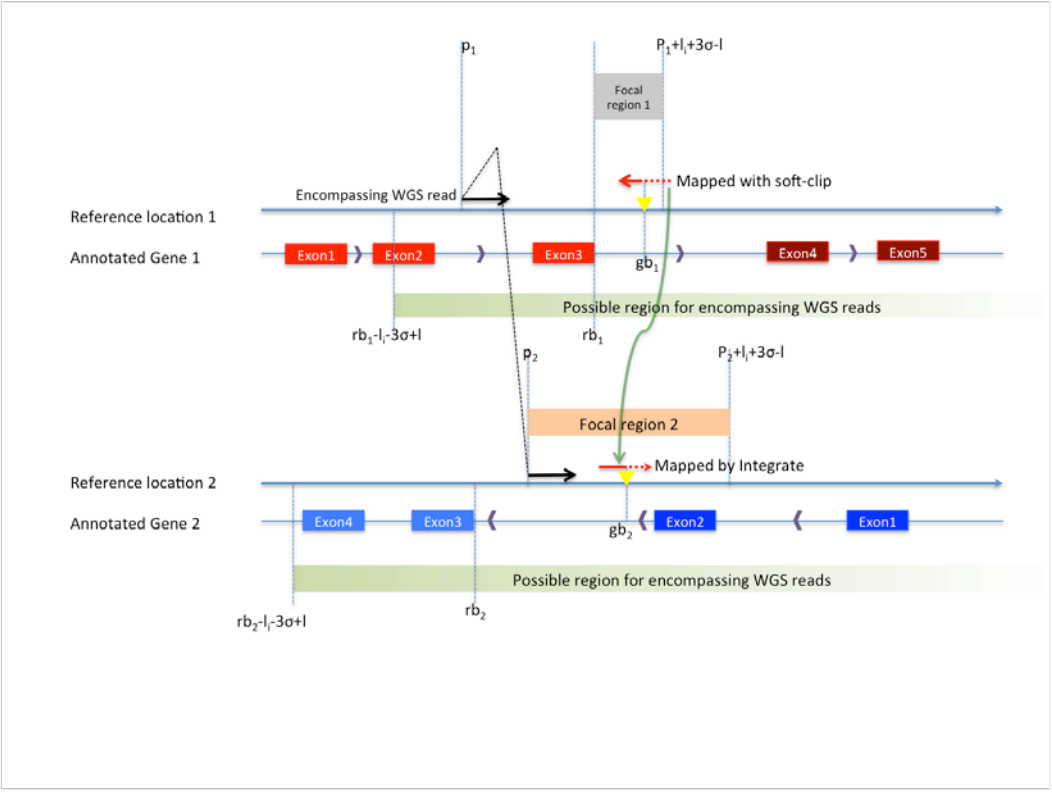
(4) Align Unmapped Portion of WGS Split Reads to the Appropriate Region Based on WGS Encompassing Pair

Note: the regions in (3) and (4) are of length $l_i+3\sigma$, where l_i and σ are the insert size and its standard deviation of an encompassing WGS read.

9. Update Fusion Type and 10. Categorize Fusion candidates in Tiers

Fusion candidates previously annotated as RNA only (i.e. Read_Through) are updated to Intra_Chromosomal if WGS evidences are found.
Fusion candidates are categorized by Tiers, e.g. :
Tier 1: Tumor only, Canonical exon, with DNA and RNA encompassing and spanning reads.

Supplementary Figure 9. Detailed INTEGRATE workflow. Key steps of INTEGRATE are illustrated with an example, which include filtering encompassing RNA-seq reads, mapping spanning RNA-seq reads, and integrating with WGS reads. Key data structures and steps for filtering encompassing RNA-seq reads are shown in detail. Method for handling spanning RNA-seq reads is given as a flow chart. Refer to Supplementary Figure 1 for more details of data structure and algorithm for split-read mapping on BWTs for gene nodes. Refer to Supplementary Figure 10 for more details of finding focal regions for WGS reads mapping.



Supplementary Figure 10. Detailed schematic of focal regions. When fusion junctions are supported by spanning RNA-seq reads, encompassing WGS reads are very likely to be in regions near the fusion junctions. The upstream boundary at the 5' gene is maximum insert size in length upstream (in regard to the strand of the 5' gene) to the 5' fusion junction (rb_1-l_i-3σ+l, where l_i is library insert size and σ is its standard deviation), and the downstream boundary at the 3' gene is maximum insert size in length downstream (in regard to the strand of the 3' gene) to 3' fusion junction. The other boundary of encompassing reads is one extended gene coordinate (default 50,000bps). When encompassing reads are retrieved from these focal regions. Spanning WGS reads can be mapped within the length of maximum insert size of the encompassing WGS reads. For example, the encompassing read at p_1, gives the boundaries of (p_1, p_1+l_i+3σ-l) for gb_1, and the fusion junction rb_1 makes focal region 1 smaller, which is (rb_1, p_1+l_i+3σ-l). Spanning WGS reads may be repetitive when mapped to whole genome but can be very unique in the their focal regions.

```
INEXACTSPLITALIGN
  //initialization of level 0 of prefix trie (also column 0 of dynamic programming)
  Add one SimulationNode sn0 for '$'
  Add 2* MAX_ALLOWED_DIFF+1 scoreNodes

  // initialization of level 1 of prefix trie
  foreach c in {'A','C','G','T'}
      k', l' = nextKL(sn0, c) // search on BWT
      Add one SimulationNode

  //breadth first search
  while next simulationNode sn

      for x  in sn.level ± MAX_ALLOWED_DIFF
          //find the scoreNodes that can lead to the current scoreNode
          sc_m from (x-1, sn.pId) //note: parent not necessarily the left column
          sc_d from (x, sn.pId)
          sc_i from (x-1, current column number)

      //dynamic programming, max score
      ms=max(sc_m.score + MATCH or MISSMATCH, sc_d.score + DEL, sc_i.score +INS)
      Add one scoreNodes with ms and diff = diff_with_ms

      //update min_diff of sn
      if diff_with_ms < sn.min_diff
          sn.min_diff=diff_with_ms

      if  sn.min_diff < MAX_ALLOWED_DIFF
          foreach c in {'A','C','G','T'}
              k', l' = nextKL(sn, c) // search on BWT
              Add one SimulationNode

SimulationNode
  pid        // parent column id
  level      // level in prefix trie
  k,l        //  range in BWT
  min_diff   //min differences (mismatchs+deletions+insertions) for the ScoreNodes in the column
  c          // character of the node

ScoreNode
  ms         // score
  diff       // difference

NodesFinder //store ScoreNodes in a vector and search in constant time by hash
NodesFinder.insert(row, column, index_in_vector)
index_in_vector = NodesFinder.find(row, column)
```

Supplementary Figure 11. Pseudo code of aligning a segment of reads to a gene node. The dynamic programming keeps track of mapping scores of the current subsegments. Search space is the paths with smaller than allowed number of differences (mismatches and indels) compared to the segments of the read. BWT is used to simulate the search paths in a prefix trie. The columns of the score matrix are connected according to the subtree searched. Differences are introduced according to the maximum score of a sub segment, which in turn affect the search space. Since the matrix is sparse, the cells of the matrix are stored in a vector. The codes for checking whether to store, inserting and finding scoreNodes are omitted.

Method	Run Time (d)	Memory (G)
INTEGRATE	0.3	31
Comrad	36	142
BreakTrans	0.5	5
nFuse	272	210
TopHat-Fusion	10	55
ChimeraScan	10	6
FusionCatcher	1	90
pyPRADA	9	26
TRUP	34	9

Supplementary Figure 12. Run time and memory usage of INTEGRATE and other 8 fusion calling methods applied to HCC1395 cell line. INTEGRATE is provided with alignment from TopHat2. BreakTrans is provided with alignments from BreakDancer. Run time is measured in days in sequential time. All the jobs was submitted to a big memory blade with 24 Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz, and 400G of memory.

Supplementary Note. Parameters for applying publicly available gene fusion prediction tools on the HCC1395 whole genome and transcriptome sequence data.

BreakTrans-0.0.6 is run using the predictions of BreakDancer for both RNA-seq and WGS data sets. The command lines are as follows:

bam2cfg hcc1395.tumor.rna.bam > rna.cfg

breakdancer-max rna.cfg > bkd.rna.bed

Bam2cfg hcc1395.tumor.dna.bam > dna.cfg

breakdancer-max dna.cfg > bkd.dna.bed

awk '{print $1"\t"$2"\t"$4"\t"$5"\tBreakDancer"}' bkd.rna.bed > input.rna.txt

BreakTrans.pl bkd.dna.bed input.rna.txt

Comrad-0.1.3 is provided with FASTQ files for tumor RNA-seq and WGS samples rna.1.fq, rna.2.fq, dna.1.fq, and dna.2.fq. The command is as follows:

analyze.pl -c config.txt -d ./hcc1395/dna/ -r ./hcc1395/rna/ -o ./hcc1395_comrad_out

nFuse-0.2.1

nfuse.pl -c config.txt --rnafq1 rna.1.fastq --rnafq2 rna.2.fastq --dnafq1 dna.1.fastq --dnafq2 dna.2.fastq -o hcc1395_nFuse_out

TopHat-Fusion-2.0.8

Commands are as follows:

tophat -o tophat_hcc1395 --fusion-search --keep-fasta-order --bowtie1 --no-coverage-search -r 0 --mate-std-dev 80 --max-intron-length 100000 --fusion-min-dist 100000 --fusion-anchor-length 13 --fusion-ignore-chromosomes chrM path_to_bowtieIndex rna.1.fq rna.2.fq

tophat-fusion-post --num-fusion-reads 1 --num-fusion-pairs 2 --num-fusion-both 5 path_to_bowtieIndex

ChimeraScan-0.4.5a

Command is as follows:

chimerascan_run.py -v path_to_chimeraScan_index rna.1.fq rna.2.q

FusionCatcher_v0.99.3e

fusioncatcher -d ./fusioncatcher_data/ -i ./hcc1395/rna/ -o ./out_dir

TRUP

Commands are as follows:

RTrace.pl --runlevel 1 --sampleName HCC1395 --seqType p --readpool ./hcc1395/rna/ --root ./ --threads 12 --anno ./TRUP_ANNOTATION/

RTrace.pl --runlevel 2 --sampleName HCC1395 --seqType p --readpool ./hcc1395/rna/ --root ./ --anno ./TRUP_ANNOTATION/ --WIG --gf pdf

RTrace.pl –runlevel 3 --sampleName HCC1395 --seqType p –readpool ./hcc1395/rna/ --root ./ --anno ./TRUP_ANNOTATION/ --RA 1

RTrace.pl --runlevel 4 --sampleName HCC1395 --seqType p --readpool ./hcc1395/rna/ --root ./ --anno ./TRUP/TRUP_ANNOTATION/

pyPRADA_1.2

Commands are as follows:

prada-preprocess-bi -conf conf.txt -inputdir ./hcc1395/rna/ -sample hcc1395 -tag HCC1395 -platform illumina -step 2_e1_1 intermediate no -pbs fromfp -outdir out_preprocess -submit no

prada-fusion -bam out_preprocess/hcc1395.withRG.GATKRecalibrated.flagged.bam -conf ./conf.txt -tag HCC1395 -mm 1 -junL 100 -minmapq 30 -outdir ./out_fusion/

Number of fusions called by each tool is counted without repeats, i.e. different isoforms and reciprocals for two genes of a fusion are counted as one fusion.