Gene Fusion Discovery with INTEGRATE

Jin Zhang and Christopher A. Maher

Abstract

Next-generation sequencing (NGS) has become the primary technology for discovering gene fusions. Decreasing NGS costs have resulted in a growing quantity of patients with whole transcriptome sequencing (RNA-seq) and whole genome sequencing (WGS) data. We developed a gene fusion discovery tool, INTEGRATE, that leverages both RNA-seq and WGS data to reconstruct gene fusion junctions and genomic breakpoints by split-read alignment. INTEGRATE has become widely adopted by the larger cancer research community to discover biologically and clinically relevant gene fusions. Here we explain the rationale driving the development of the INTEGRATE tool and describe the detailed practical procedures for applying INTEGRATE to discover gene fusions using NGS data. INTEGRATE can be applied to both combined data and RNA-seq only data.

Key words Gene fusion, Chimeras, Structural variation, Next-generation sequencing, Whole transcriptome sequencing, RNA-seq, Whole-genome sequencing, Cancer

1 Introduction

Genomic instability and rearrangements are important hallmarks of cancer [1]. Deletions, insertions, inversions, or translocations can generate intrachromosomal or interchromosomal rearrangements, a subset of which brings two genes together, which we refer to as a gene fusion [2]. The subsequent gene fusions can result in the activation of an oncogene, inactivation of a tumor suppressor, or the generation of a new protein product [3]. Some of the most well-studied gene fusions have played critical roles in hematological malignancies. Philadelphia chromosome found in chronic myelogenous leukemia (CML) consisting of the translocation of chromosome 9 and 22 creating the *BCR-ABL* gene fusion, an activated tyrosine kinase that drives CML [4]. Bioinformatics approaches using NGS data accelerate the process of discovering novel gene fusions in both leukemia and solid tumors. The seminal discovery of recurrent gene fusions between *TMPRSS2* and the ETS oncogenic transcription factor *ERG* in ~65–75% of prostate cancer patients demonstrated the significance of gene fusion extends

Hui Li and Justin Elfman (eds.), *Chimeric RNA: Methods and Protocols*, Methods in Molecular Biology, vol. 2079, https://doi.org/10.1007/978-1-4939-9904-0_4, © Springer Science+Business Media, LLC, part of Springer Nature 2020

beyond hematological malignancies [5, 6]. The discovery demonstrated that a gene fusion was a common event acquired early in prostate cancer progression and represents a potential driver of disease progression. Since then, advances in next-generation sequencing have accelerated our ability to discover gene fusions across solid tumors. Notably, one of the first gene fusion discovery tools, called ChimeraScan [7], was used to discover novel recurrent ETS gene fusions, including an RNA chimera between an androgen responsive 5′ partner, *SLC45A3*, and ETS transcription factor family member *ELK4* [5]. Overall, analysis of NGS data revealed mutually exclusive gene fusions in prostate cancer patients [3]. During the past 10 years, tens of thousands of gene fusions predicted by computational tools have been reported in virtually all cancer types, especially with the development of international consortium efforts such as TCGA.

Gene fusion discovery using NGS data is still an extremely challenging computational problem. It is not uncommon that the state-of-the-art gene fusion discovery tools may report hundreds of false positive gene fusion candidates in only one tumor sample in order to detect all of the handful of "gold-standard" gene fusions, whereas other tools failed to detect these gene fusions at all [8]. To address the limitations inherent in the existing methods, we developed a new gene fusion discovery tool, INTEGRATE, which utilizes orthogonal validation from both RNA-seq and WGS data [9]. INTEGRATE achieved significant improvement in sensitivity and accuracy compared to contemporary fusion-calling algorithms. Using INTEGRATE, we facilitated several studies using RNA-seq only data or both WGS and RNA-seq data to discover diver gene fusions in various human cancers. For example, we used INTEGRATE to discover recurrent *ESR1* gene fusions in primary breast cancer patients using both RNA-seq and WGS data [9], and followed up to discover an endocrine-therapy-resistant *ESR1-YAP1* fusion in breast-cancer-derived xenograft models [10]. Recently, we discovered a series of endocrine-therapy-resistant *ESR1* gene fusions in late stage breast cancers, and showed that these *ESR1* fusions also reprogram the ER cistrome to drive EMT and metastasis [11]. We also used INTEGRATE to discover *DNAJB1-PRKACA* as a biomarker in mixed fibrolamellar hepatocellular carcinoma [12], and a highly expressed *EP300-ZNF384* driver gene fusion in a rare case of relapsed adult B-lymphoblastic leukemia [13]. In addition, we helped multiple institutes to deploy INTEGRATE in their pipeline. Researchers have applied INTEGRATE in numerous cancers to discover functionally relevant gene fusions (e.g., *MYBL1* rearrangements in breast adenoid cystic carcinomas [14] and *HMGA2* and *PLAG1* rearrangements in breast adenomyoepitheliomas [15]).

Over the past few years since the release, and publication, of INTEGRATE we have continued to update our tool based on the needs of the cancer research community. We initially set up a code depositary at SourceForge (https://sourceforge.net/projects/integrate-fusion/) to include the updated source code, sample input files, test cases, and Wiki pages. The Wiki pages include the details of how to download, install, build index files, and run INTEGRATE, as well as explanations of the input/output formats, options, and release notes. In the supplemental materials of the manuscript, we provided command lines of preparing input BAM files using different read-alignment tools. In addition to the INTEGRATE gene fusion discovery tool, we have expanded the INTEGRATE series tools, including the first gene fusion neoantigen discovery tool, INTEGRATE-Neo [16], and a comprehensive gene fusion visualization tool, INTEGRATE-Vis [17]. These new tools were shared at GitHub (https://github.com/ChrisMaherLab/), which include companion examples and command lines to execute the INTEGRATE series of tools. While organizing the ICGC-TCGA DREAM Somatic Mutation Calling RNA Challenge (SMC-RNA), a community-based collaborative competition of researchers from across the world to help improve the standards adopted across the cancer research community for RNA-seq analyses, we adapted INTEGRATE to support the input/output format defined by the challenge. Recently, the INTEGRATE paper was selected by the International Medical Informatics Association (IMIA) as one of the four best articles in the "Bioinformatics and Translational Informatics" subfield of medical informatics literature published in 2016. Given the widespread adoption of the INTEGRATE suite of tools by the cancer research community, this chapter focuses on design rationale and optimal use of INTEGRATE.

Our initial INTEGRATE paper largely focused on introducing the workflow and algorithms of the INTEGRATE tool, which covered the nitty-gritty of a dedicated gene fusion discovery tool of more than 10,000 lines of C++ code. In Subheading 2, we will outline the rationale and philosophies behind the implementation of INTEGRATE. We will explain why the INTEGRATE tool processes RNA-seq data first instead of processing WGS data first. We will discuss the difference between using combined WGS and RNA-seq and using only RNA-seq data, and how this will affect the INTEGRATE's model of Tiers for gene fusion candidates. We will introduce the philosophy behind the INTEGRATE's model of controlling false positives, while keeping high sensitivity. We will discuss the ways of representing split-reads in BAM files, and how this will affect the successful run of INTEGRATE, in terms of not only executing the tool but also generating meaningful results. A better understanding of the rationale driving the development of INTEGRATE will also help the readers understand the major procedures and considerations in executing INTEGRATE, which will be the next major topic after the rationale.

In Subheading 3, we will outline the detailed practical procedures for applying INTEGRATE to discover gene fusions using either combined WGS and RNA-seq data or using only RNA-seq data. In this section, we will walk through the steps of installing INTEGRATE to a typical Linux system, building indices for the Burrows–Wheeler transform of the Human reference genome, preparing a gene annotation file as an input to INTEGRATE, and executing INTEGRATE to discover gene fusions. We will provide detailed information on the formats of the input/output files of INTEGRATE and how to connect the output files to other INTEGRATE series tools and third-party tools. We will walk through multiple key points for quality control purposes when executing INTEGRATE. For more advanced users, we will discuss how to set certain parameters in different situations to achieve better results instead of using default parameters. We will also provide suggestions for using matched tumor and adjacent normal samples.

In Subheading 4, we will include notes that cover multiple troubleshooting aspects for the readers and alternative strategies. We will also provide key points for the users regarding to the computational resources and timescale they should anticipate, and options that are available to prioritize gene fusion candidates and set up validation experiments.

2 Rationale Driving the Development of INTEGRATE

2.1 Using RNA-Seq Data Before WGS Data

To discover gene fusions, it may appear appealing to start with WGS data and follow with RNA-seq data, since this represents conceptually that a gene fusion is the result of a certain structural variation, and practically the mapping quality of WGS data is usually better than RNA-seq data (i.e., percentage of reads mapped, uniqueness the of mappings, etc.) As part of the INTEGRATE workflow, we have chosen to process the tumor RNA-seq data before the WGS data for a couple of reasons. First, we observed that many genomic events may associate with repetitive reads, while expressed gene fusion transcripts may involve exons that are quite unique. Therefore, RNA-seq reads could lead to higher sensitivity in nominating candidate gene fusions compared to WGS reads. From the aspect of combining WGS and RNA-seq data, analyzing more repetitive WGS reads can be much easier in local regions adjacent to fusion junctions discovered by RNA-seq reads (as is implemented in the INTEGRATE tool). Second, not necessarily all genomic events produce a gene fusion. Structural variations in the intergenic regions are unlikely to produce any transcripts, less likely functionally related gene fusion transcripts. Structural variations with junctions involving one or two genes may not all produce expressed gene fusions, either. Therefore, effort to discover expressed gene fusions in RNA-seq data for these structural

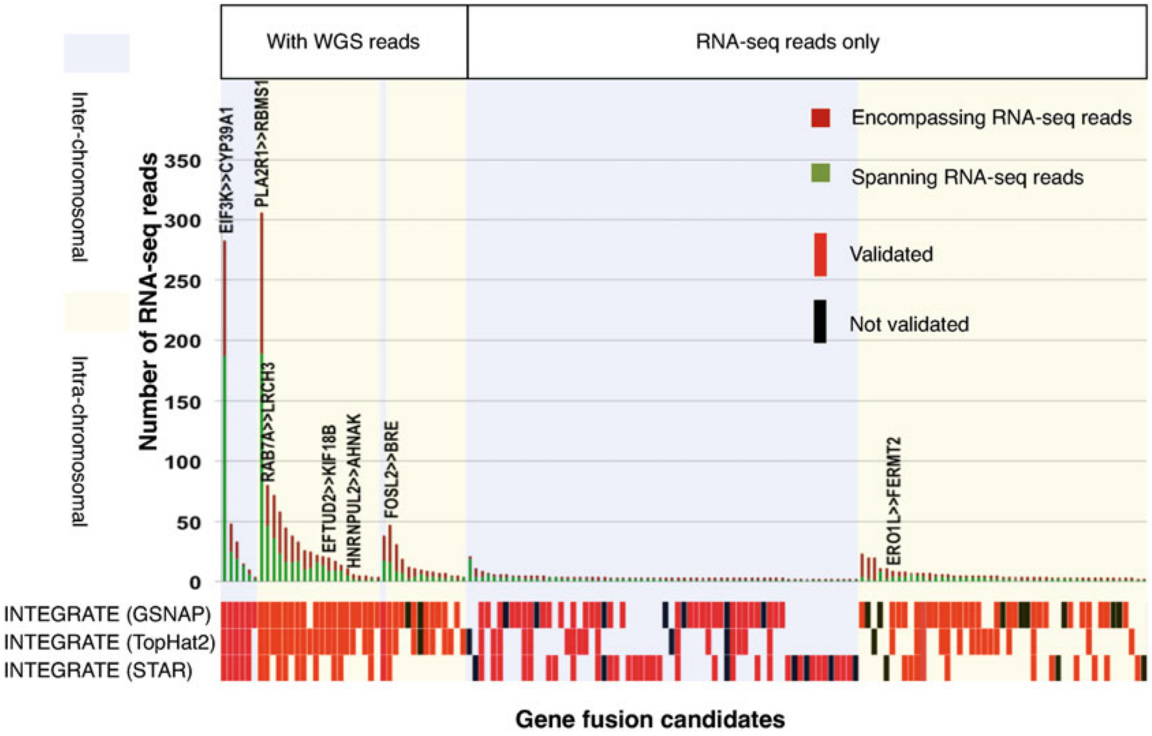
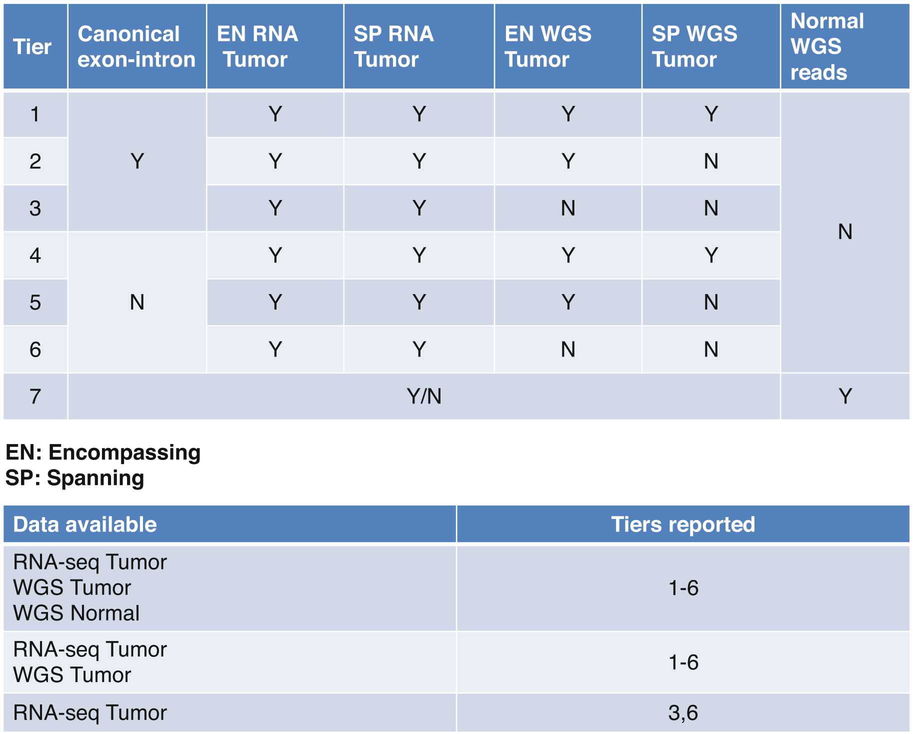


Fig. 1 Validated gene fusions in HCC1395 cells. INTEGRATE discovers gene fusions with both RNA-seq and WGS reads, as well as only RNA-seq reads. Not all gene fusion will have WGS read support. Previously reported gene fusions in the cells are written at the top of the bars for number of RNA-seq reads

variations will not yield any true positive gene fusions but would take time for an algorithm to analyze. Third, although less appreciated a couple of years ago when bulk sequencing was the dominant sequencing technology, recent years have seen more reports on the clonal evolution of tumors using single nucleotides somatic variations; therefore, conceptually a highly expressed gene fusion driving an emerging aggressive subclone may show only very low variant allele frequency (VAF) in the primary tumor but could be traceable using RNA-seq data. Last, RNA only chimeras, such as trans-spliced and read-through chimeric RNAs, 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 gene fusion candidates followed by examining their genomic breakpoints but not necessarily requires WGS reads to discover gene fusions (Fig. 1).

2.2 Annotating Gene Fusion Candidates Using Tiers

The highly sensitive, accurate, and efficient INTEGARTE gene fusion discovery algorithm has improved our ability to comprehensively detect gene fusions in human cancers. To prioritize gene fusion candidates, INTEGRATE reports gene fusions in Tiers corresponding to the level of sequencing support (Fig. 2), which provides potential insights into their mechanism of formation.



Tier	Canonical exon-intron	EN RNA Tumor	SP RNA Tumor	EN WGS Tumor	SP WGS Tumor	Normal WGS reads
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	
7	Y/N					Y

EN: Encompassing
SP: Spanning

Data available	Tiers reported
RNA-seq Tumor WGS Tumor WGS Normal	1-6
RNA-seq Tumor WGS Tumor	1-6
RNA-seq Tumor	3,6

Fig. 2 Annotating gene fusion candidates using Tiers. Gene fusions with canonical exon–intron boundaries are included in Tiers 1–3 whereas gene fusions without canonical exon–intron boundaries are included in Tier 4–6. When only RNA-seq data is available INTEGRATE reports gene fusions with and without canonical exon–intron boundaries as Tier 3 and 6, respectively

Tiers 1, 2, and 3 all involve gene fusions with canonical exon–intron boundaries. Tier 1 candidates have the highest confidence as they have both encompassing and spanning RNA-seq and WGS reads supporting a gene fusion (Fig. 3). Tier 2 gene fusion candidates also have both WGS and RNA-seq read support; however, they only have encompassing WGS read support and lack spanning WGS reads. Tier 3 lacks any WGS read support but has both encompassing and spanning RNA reads. However, Tier 3 includes both non-read-through gene fusions (Tier 3-nr) and read-throughs (Tier 3-r). Our analysis showed that Tier 3-nr gene fusions have a pattern of exon usage similar to Tier 1 and Tier 2 gene fusions, and Tier 3-r read-throughs have a different pattern of exon usage (Fig. 4).

Gene fusions with junctions not at canonical exon–intron boundaries (i.e., at introns or truncated exons) are included in Tiers 4–6. Tier 4 candidates have both encompassing and spanning RNA-seq and WGS reads. Tier 5 gene fusion candidates only have encompassing WGS reads. Tier 6 lacks any WGS read support but has both encompassing and spanning RNA reads. Tier 6 can also include both non-read-through gene fusions (Tier 6-nr) and read-throughs (Tier 6-r). When only RNA-seq data is available,

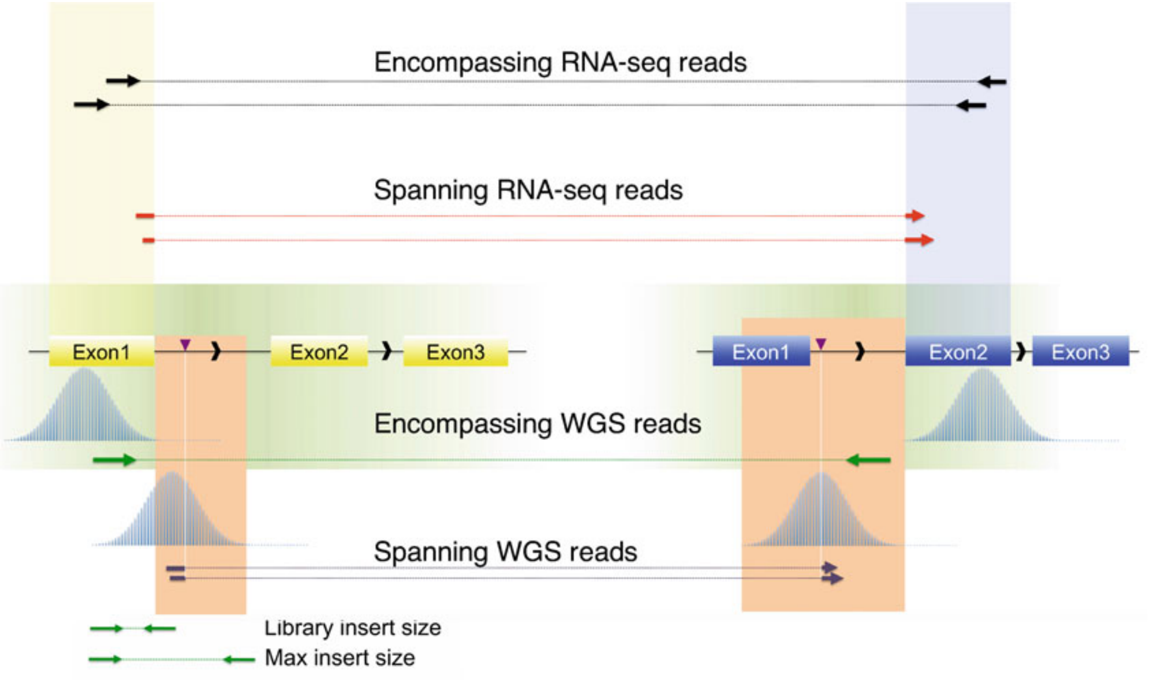


Fig. 3 Encompassing and spanning RNA-seq and WGS reads. When encompassing (black) and spanning (red) RNA-seq reads have been mapped to the genes involved in a gene fusion, the encompassing WGS reads (green) are expected from focal encompassing WGS regions (green area) bounded by maximum insert size upstream of or downstream from the fusion junctions of the transcripts. The spanning WGS reads (purple) are expected to align within focal WGS regions (orange area) bounded by fusion junction and maximum insert size downstream from the encompassing WGS reads

INTEGRATE only reports gene fusions with and without canonical exonic boundaries as Tier 3 and 6, respectively. When adjacent normal or blood samples are available, INTEGRATE can also report a Tier 7, which are the candidates discovered in the tumor samples, but at the same time likely to be germ line.

In INTEGRATE's ".summary.txt" output files, gene fusion candidates are annotated with the categories of "inter-chromosomal," "intra-chromosomal," or "read-through," depending on whether the two gene partners are located on the same chromosome and the locus of the gene partners. There are multiple distinct categories of patterns—we call each a pattern a Class of gene fusions—we observed to show frequently among gene fusion candidates. The first category (Class I) is caused by an interchromosomal translocation resulting in the juxtaposition of two genes thereby creating a gene fusion such as *BCR-ABL*. Class II are complex interchromosomal rearrangements that may result in the regulatory region of a nearby gene being altered following a rearrangement as exemplified by the gene fusion of *MIPOL1-DGKB* activating *ETV1* in LNCaP cells. Class III fusions are classified by an intrachromosomal deletion resulting in the fusion of the two genes flanking the deleted region as exemplified by *TMPRSS2-ERG*.

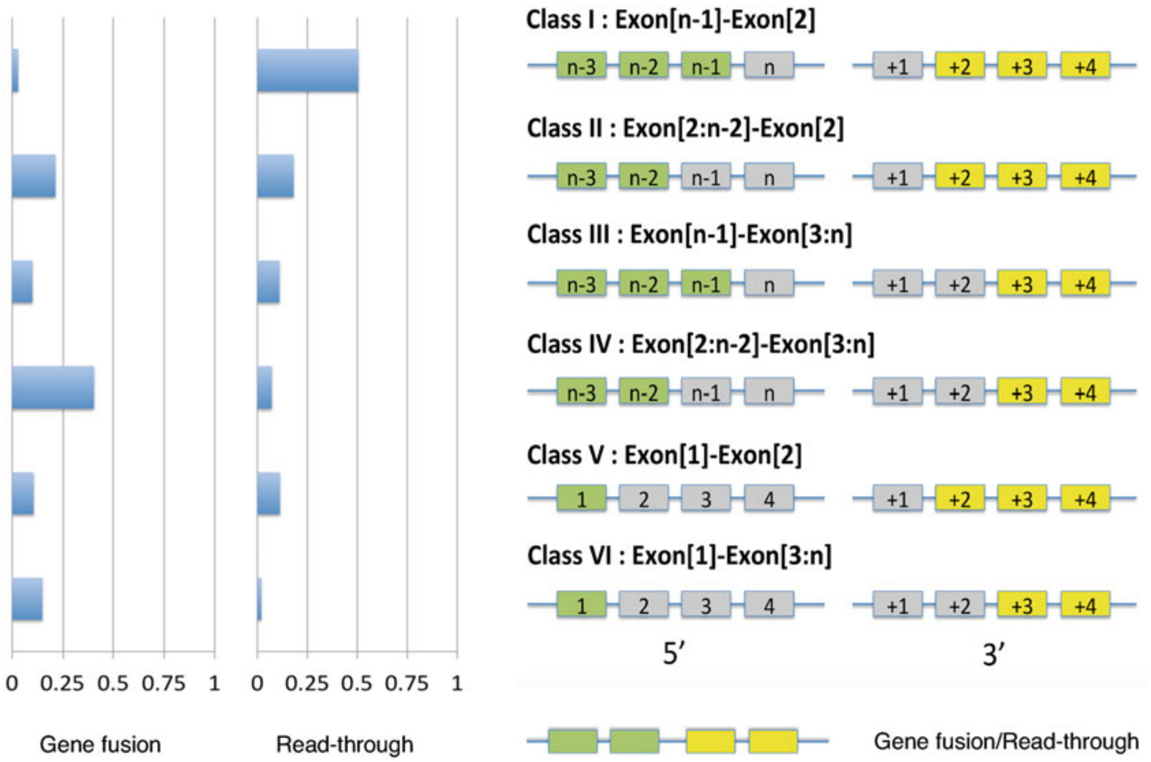


Fig. 4 Exon usage for gene fusions in different Tiers. A gene fusion (or read-through) transcript can be categorized into six classes involving the first, second to last, or any other exon of the 5′ gene with either the second or downstream exon of the 3′ gene. Exons involved in gene fusions and read-throughs follow different patterns

Class IV fusions are complex intrachromosomal rearrangements in which a single gene may harbor multiple rearrangements as exemplified by *HJURP* harboring multiple breakpoints and thereby generated the *HJURP-EIF4E2* and *INPP4-HJURP* fusion transcripts. There are also many different kinds of intrachromosomal focal gene fusions that could arise from local events including duplication and inversion. The last category is comprised of read-through events (Class V). Overall, such a classification system may help distinguish and prioritize gene fusions that may arise from different mechanisms. The users can also use a companion gene fusion visualization tools, called INTEGRATE-Vis, we developed recently [17], to visualize gene fusions and predict their functions. Our tool not only supports INTEGRATE but also supports all gene fusion discovery tools that adopt a standard BEDPE format for gene fusions [17]. The user can not only use the tool to visualize the structure of the Classes, but also visualize and predict the functions of the gene fusions including exon expression, gene expression, and protein domains.

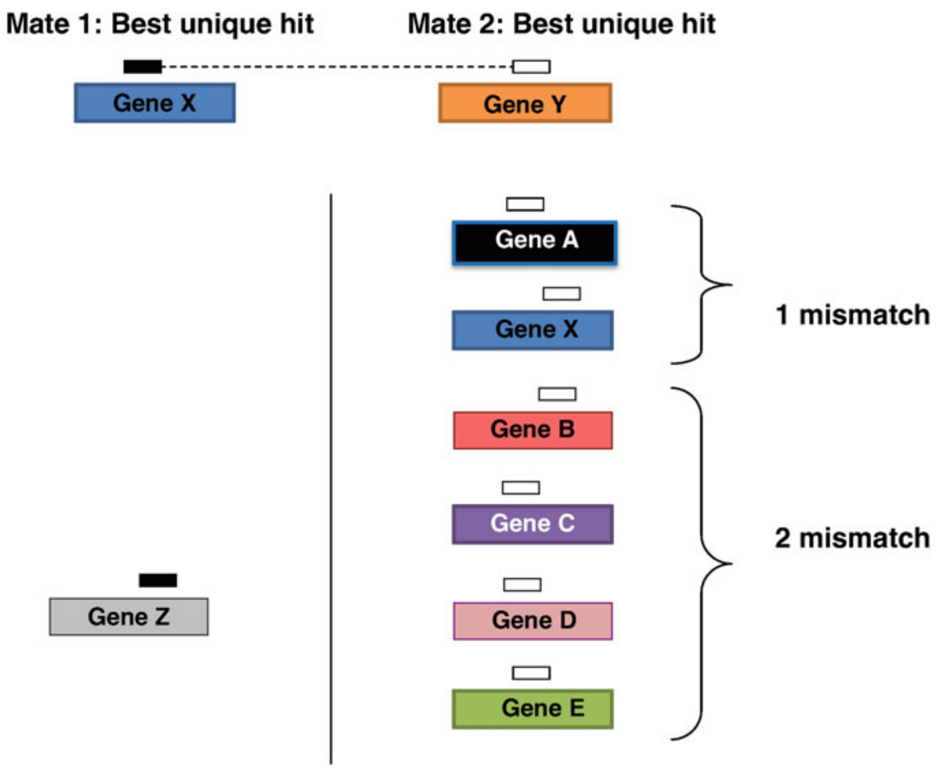


Fig. 5 Iteration across suboptimal mappings reveals logical pairing. Suppose a pair of reads can be aligned to two genes, X and Y, respectively. This does not necessarily mean they are from a gene fusion. This example shows the Mate 2 of the pair can also be aligned to Gene X, if considering suboptimal alignments. Therefore, it is very likely the pair of reads is from gene X, instead of a gene fusion between genes X and Y

2.3 Modeling Gene-Fusion Reads Using Wild-Type Genes

To define a Bioinformatics application using sequencing data, usually certain signatures from sequencing reads from a biological event were abstracted to facilitate algorithm design. Encompassing reads, spanning reads, copy number changes, and exon expression changes, are often used in structural variation and gene fusion discovery tools. Often times a structural variation or gene fusion discovery tool requires a certain number of each type of reads as the threshold of calling the biological event. From the frequentist point of view, the more sequencing reads supporting the same event the more likely the event is real. When we implemented INTEGRATE, we adopted a Bayesian point of view, seeking simpler explanations rather than gene fusions that could explain the signatures from the sequencing reads. Specifically, the INTEGRATE null model is that there is no gene fusion, unless the encompassing and spanning reads provide strong evidence and there are no other possible explanations. For all the candidate encompassing and spanning reads between two gene partners, INTEGRATE checks that whether a suboptimal alignment of the read can fit in one wild-type gene of the two candidate gene partners. If this is the case, these aligned reads will be regarded as normal reads from a wild-type gene instead of a read from a gene fusion event (Fig. 5).

2.4 Supporting Multiple Formats for Split-Reads

INTEGRATE is designed to discover gene fusions using RNA-seq and WGS paired-end sequencing reads properly aligned to the reference genome in BAM format. Unlike many gene fusion tools which are programmed to use a specific reads mapping tool, INTEGRATE was implemented with the flexibility of using RNA-seq reads aligned by different tools, including GSNAP [18], TopHat2 [19], and STAR/STAR2 [20]. For a spanning RNA-seq read that is from a gene fusion, the read-alignment tools may have not aligned it yet, thus leave it as an unaligned read. The RNA-seq read-alignment tools may also have aligned some of the spanning RNA-seq reads and stored them in the BAM files in tool specific formats. For the aligned spanning reads, GSNAP and STAR/ STAR2 use soft-clipping to represent the partially aligned portion of the read. STAR/STAR2 also use hard-clipping for supplemental chimeric alignments when output to the Aligned.∗.bam. TopHat2 leaves all spanning RNA-seq reads as unaligned reads. INTEGRATE takes both the unaligned reads and partially aligned (soft-clipped or hard-clipped) reads as candidate spanning RNA-seq reads for gene fusions, and uses its own split-read alignment algorithm to map the read to the gene partners to decide whether it is a real spanning read, a wild-type read or a spurious read.

While INTEGRATE covers all three types of situations (i.e., unaligned, soft-clipped, and hard-clipped reads) to be flexible and user-friendly, real applications can be more complicated. For example, a pair of reads with a soft-clipped spanning read in the pair, can be interpreted as either “properly aligned” in the sense that the aligned parts of the whole fragment are from the same wild-type gene, or “not properly aligned” in the sense that portions of the whole fragment can be aligned in two genes. Two read-alignment tools could choose either representation and therefore set opposite bitwise flags, yet both methods follow the Sequence Alignment Map (SAM) format. The same read alignment tool can represent different spanning reads using all three types reads, that is, unaligned, soft-clipped, and hard-clipped reads. For each type of reads, the tool can interpret and encode them differently from other tools in the SAM format. INTEGRATE handles these reads using different subroutines implemented in the tool, which can handle the different representations from the three tools we mentioned above. Apparently, we cannot implement INTEGRATE to support all read-alignment tools, and not all authors of the read alignment tools will support to convert the output of their tools to be consistent with the above mentioned three RNA-seq read alignment tools. Therefore, we kindly ask the users of the INTEGRATE tool to test using positive controls whether the BAM files they prepared are consistent with INTEGRATE. Over the years, from the feedbacks of the users of the INTEGRATE tool, we found different situations that INTEGRATE finished in processing the BAM files provided by the user and printed out result files, however the result may only contain read-throughs, or a subset of gene fusions discovered by one subroutine of INTEGRATE. Had the users be more aware of the issues of representing split-reads in the SAM format and prepared the BAM files accordingly, more gene fusions could be discovered from the same data (*see* **Note 1**).

3 Methods

INTEGRATE is a genomics tool for discovering gene fusions with exact fusion junctions and genomic breakpoints by combining RNA-Seq and WGS data. It is highly sensitive and accurate by applying a fast split-read alignment algorithm based on Burrows–Wheeler transform. In this section, we will walk you through the details of installing and executing INTEGRATE and how to interpret INTEGRATE output, as well as multiple aspects for advanced users to optimize their analyses.

3.1 Installing INTEGRATE

INTEGRATE is a standalone tool with minimal software dependencies and additional requirements provided (*see* **Note 2**). The users will find it easy to install and use. In this section, we provide the steps of installing INTEGRATE to a Linux system, such as Ubuntu. The procedures would be applicable to other Linux/Unix systems with minimum changes. Alternatively, the users can choose to run the Docker image we provided (*see* **Note 3**).

1. Download and install SAMtools. INTEGRATE calls multiple of the interfaces of SAMtools to retrieve read alignments from BAM files. SAMtools can be downloaded at http://www.htslib.org/download/. SAMtools can be installed using the make and make install commands.
2. Download and install libdivsufsort. INTEGRATE calls interfaces of libdivsufsort to build Burrows–Wheeler transforms for reference sequences. libdivsufsort can be downloaded at https://code.google.com/p/libdivsufsort/. libdivsufsort requires the CMake tool to install (*see* **Note 2**).
3. Download and install CMake. We have provided code to use CMake to automatically compile INTEGRATE. CMake can be downloaded at: http://www.cmake.org/. CMake can be installed using the make and make install commands.
4. Download the INTEGRATE package. The INTEGRATE package, named following the format of INTEGRATE.X.Y.Z.tar.gz can be downloaded at https://sourceforge.net/projects/integrate-fusion/files/ or https://github.com/ChrisMaherLab/INTEGRATE, where X, Y, and Z are digits for the version (e.g., INTEGRATE.0.2.6.tar.gz).

5. Extract the INTEGRATE package. Command using tar to extract the INTEGRATE package is:

```
$ tar xvzf INTEGRATE.X.Y.Z.tar.gz
```

   Now, the source code of INTEGRATE, prerequisite packages, CMake code for automatically compiling INTEGRATE and other prerequisite packages are in the folder called INTEGRATE_X_Y_Z (e.g., INTEGRATE_0_2_6).

6. Compile INTEGRATE using CMake. Create a folder called INTEGRATE-build and compile INTEGRATE in this folder, instead of in the folder of the INTEGRATE source code (*see* **Note 4**).

```
$ cd INTEGRATE_X_Y_Z
$ mkdir INTEGRATE-build
$ cd INTEGRATE-build
$ cmake ../Integrate/ -DCMAKE_BUILD_TYPE=release
$ make
```

   Now the executable software of Integrate is at INTEGRATE_X_Y_Z /INTEGRATE-build/bin/, you can copy it to any folder you prefer, or add the path to this folder to the system PATH variable.

3.2 Build Index Files for the Chromosomes in the Reference Genome

1. Download the human reference genome. Here we use GRCh38 v85 as an example, which is for the Genome Reference Consortium Human Build 38 version 85. We provide the command lines for downloading the reference genome from Ensembl's FTP site. The users can choose to use different builds and versions (*see* **Note 5**).

```
$ wget ftp://ftp.ensembl.org/pub/release-
85/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38.d
na.chromosome.{1..22,X,Y,MT}.fa.gz
        $ gunzip -c
Homo_sapiens.GRCh38.dna.chromosome.* >
GRCh38_r85.all.fa
```

2. Build index files for the Burrows–Wheeler transform (BWT) of the sequences in the reference genome (*see* **Note 6**).

   First create a directory, called *bwts*:

```
$ mkdir ./bwts
```

Then, run INTEGRATE's *mkbwt* tool on the reference genome:

```
$ Integrate mkbwt GRCh38_r85.all.fa
```

3.3 Prepare a Gene Annotation File

We recommend the users to create a gene annotation file from using a GTF file. Here we use Ensembl genes based on GRCh37 v85 as an example. The users can create their own gene annotation files from the databases they prefer. In addition, the users can also use the UCSC Genome Browser to create gene annotation files from multiple different databases (*see* **Note 7**).

1. Download and extract the GTF file.

```
$ wget ftp://ftp.ensembl.org/pub/release-
85/gtf/homo_sapiens/Homo_sapiens.GRCh38.85.gtf.
gz
$ gunzip Homo_sapiens.GRCh38.85.gtf.gz
```

2. Run the following commands to create the gene annotation file:

```
$ gtfToGenePred -genePredExt -geneNameAsName2
Homo_sapiens.GRCh38.85.gtf
Homo_sapiens.GRCh38.85.genePred
$ cut -f 1-10,12
Homo_sapiens.GRCh38.85.genePred > tmp.txt
$ echo -e
"#GRCh38.ensGene.name\tGRCh38.ensGene.chrom\tGR
Ch38.ensGene.strand\tGRCh38.ensGene.txStart\tGR
Ch38.ensGene.txEnd\tGRCh38.ensGene.cdsStart\tGR
Ch38.ensGene.cdsEnd\tGRCh38.ensGene.exonCount\t
GRCh38.ensGene.exonStarts\tGRCh38.ensGene.exonE
nds\tGRCh38.ensemblToGeneName.value" >
annot.ensembl.GRCh38.v85.txt
$ cat tmp.txt >> annot.ensembl.GRCh38.v85.txt
```

3.4 Prepare Input BAM Files

1. Get accepted_hits.bam and unmapped.bam for RNA-seq data using TopHat2 (http://ccb.jhu.edu/software/tophat/index.shtml). The users can also create the equivalent of the accepted_hits.bam and unmapped.bam files using other RNA-seq read alignment tools (*see* **Note 8**).
2. Get dna.tumor.bam by aligning WGS reads to the reference genome using BWA's aln and sampe algorithms (*see* **Note 9**). BWA [21] can be downloaded from http://bio-bwa.sourceforge.net.

3. Make indices for all the BAM files with SAMtools index (*see* **Note 10**).

3.5 Execute INTEGARTE to Discovery Gene Fusions

1. Run a typical INTEGRATE command line that is shown as follows using both RNA-seq and WGS data.

```
$ Integrate fusion path/to/GRCh38_r85.all.fa
path/to/Homo_sapiens.GRCh38.85.txt
path/to/bwts/
path/to/accepted_hits.bam path/to/unmapped.bam
path/to/BWA.tumor.WGS.sort.bam
```

GRCh38_r85.all.fa is the reference genome; Homo_sapiens.GRCh38.85.txt is the gene annotation file; accepted_-hits.bam and unmapped.bam are BAM files for aligned and unaligned RNA-seq reads; and BWA.tumor.WGS.sort.bam is the BAM file for WGS reads from the tumor sample.

The paths to these input files must be in the exact order as shown in the above sample command line. Options can be inserted between the application name of "fusion" and path to the reference genome. BWA.tumor.WGS.sort.bam can be omitted to run RNA-seq only mode, and BWA.normal.WGS.sort.bam can be appended if an adjacent or blood normal sample is available. INTEGRATE will automatically figure out the combinations of data sets that are available for the analysis (*see* **Note 11**).

2. Examine the output files of INTEGRATE. The most import output file from INTEGRATE is the BEDPE file. INTEGRATE is currently supporting the BEDPE format used by the SMC-RNA challenge. The format contains 11 fields in order, and only includes the data but not headers. The fields are:

    (a) chrom1: The name of the chromosome on which 5′ gene partner is located.

    (b) start1: position of 5′ fusion junction or 5′ end of transcript.

    (c) end1: position of 5′ fusion junction or 5′ end of transcript.

    (d) chrom2: The name of the chromosome on which 3′ gene partner is located.

    (e) start2: position of 3′ fusion junction or 3′ end of transcript.

    (f) end2: 3′ position of 3′ fusion junction or 3′ end of transcript.

    (g) name: name of the gene fusion, GENE1> > GENE2.

    (h) score: any number or string.

(i) strand1: ±.

(j) strand2: ±.

(k) quantity of expression: any number. This column will be used for further evaluation in quantification challenge. "." for unknown if not taking quantification challenge.

The challenge only allows: 1-22, X, and Y for chrom1 and chrom2 due to GRCh37 used in the challenge. However, the INTEGRATE user can use any names of the sequences they provided if not participating in the challenge.

Rules for start1/start2/end1/end2 are as follows:

(a) start[1,2] is 0-based.

(b) end[1,2] is 1-based.

(c) Use −1 for unknown.

(d) Three allowed situations (using start1 as an example, start2 follows the same set of rules):

- start1 + 1 = end1, start1 ≠ −1 and end1 ≠ −1: *Both start1 and end1 represent the same position of the fusion junction.*
- start1 + 1 < end1, start1 ≠ −1 and end1 ≠ −1: *Starting and ending positions of the part of transcript in the fusion transcript.*
- One is −1, the other is not −1: *The position that is not −1 represents the fusion junction.*

Examples (tab delimited) are given in Table 1.

The "summary.tsv" file provides multiple key characteristics about the gene fusion candidate, including whether the gene fusion is reciprocal, Tier, Type (inter-chromosomal, intra-chromosomal, read-through), number of supporting reads (*see* **Note 12**), and a list of fusion isoforms with numbers of spanning RNA-seq and types (whether it is at canonical exon–intron boundary).

The "exons.tsv" file provides the sequences of the exons where the gene fusion junctions are located. This only includes the gene fusions with canonical exon–intron boundaries. The users can align the exon sequences to the human genome using UCSC Genome Browser, which will provide background knowledge of the gene and whether the loci are repetitive.

The "breakpoints.tsv" file provides a summary of the coordinates of fusion junctions and genomic breakpoints. This is the most useful when both RNA-seq and WGS reads are used. It will help the users figure out the structure of a pair of structure variation and the resulting gene fusion.

Table 1
Examples of SMC-RNA's output in BEDPE format supported by INTEGRATE

21	42880007	−1	21	−1	39817544	TMPRSS2 ≫ ERG	3	–	–	10
21	42880007	42880008	21	39817543	39817544	TMPRSS2 ≫ ERG	3	–	–	10
21	42880007	42880085	21	39751949	39817544	TMPRSS2 ≫ ERG	3	–	–	10

The lines show three different representation of the same gene fusion transcript. Columns 8 and 11 are reserved for user quality score and SMC-RNA's quantification challenge. INTEGRATE outputs using the format in line one, where −1 represents that short sequencing reads at the fusion junctions cannot cover the starting or ending loci of the gene partners. INTEGRATE uses column 8 for Tiers, and column 11 for number of spanning reads for the fusion isoform

The "reads.txt" file provides all the encompassing and spanning reads from all the data used in the INTEGRATE run (e.g., tumor RNA-seq, tumor WGS, and normal WGS). Spanning RNA-seq reads for different isoforms for the same pair of gene fusion partners are grouped together. This file will facilitate the user for manually inspecting the result (e.g., BLAT using UCSC Genome Browser).

The "bk_sv.vcf" is reserved for the users who are more interested in the genomic breakpoints. However, as pointed out earlier in this chapter that there may be gene fusions discovered using RNA-seq data only or with no WGS read support; therefore, the resolution for the genomic breakpoints may be based on the gene fusion junctions.

3. Examine normal RNA-seq data. This step is optional. However, if matched RNA-seq data is available, germ line chimeric transcripts can be discovered. The users can use this data to filter the gene fusion candidates discovered from the tumor RNA-seq sample. When running INTEGRATE using normal RNA-seq data, we recommend the users to use the -normal option, which will insert the term "normal" in both the output files and their names. When normal WGS data is available, the user can also append it to the command line, for example as follows:

```
$ Integrate fusion -normal (other options)
reference.fasta annotation.txt directory_to_bwt
accepted_hits.normal.bam unmapped.normal.bam
dna.normal.bam
$ Integrate fusion -normal (other options)
reference.fasta annotation.txt directory_to_bwt
accepted_hits.normal.bam unmapped.normal.bam
```

4. Test with nondefault options. This step is optional.

   -cfn: default 3. INTEGRATE discovers gene fusions on both canonical exon–intron boundaries and other gene fusions with junctions in introns and truncated exons. For these "noncanonical" cases, a default cutoff of three spanning RNA-seq reads is used. The users can choose to increase the value to report smaller number of candidates that are not from the canonical exon–intron boundaries, or decrease the cutoff to be more sensitive.

   -minW: default: 2, is the minimum weight for the encompassing RNA-seq reads on an edge (The value is a weighted value according to how repetitive the reads are.) The users can reduce the number to a lower value (can be noninteger) to be more sensitive especially when the total number of reads from the sample is low.

   -minIntra: default: 400,000. If only using RNA-seq reads, a chimera with two adjacent genes in order is annotated as intra-chromosomal rather than read-through if the distance of the two genes is longer than this value. This helps prioritize a possible gene fusion caused by a deletion from all the other read-throughs. The user can also check the exon usage of these chimeras. A typical read-through skips the last exon of the first gene partner and fuses to the second exon of the second partner.

   -minDel default: 5000, is the minimum size of a deletion that are considered to be able to cause a fusion. When both RNA-seq and WGS data are used, there may be small deletions in the length range of a couple of hundreds of nucleotides or shorter discovered between the two gene partners of the read-throughs. Often times, a pair of reads from a longer fragment may also exist between the read-throughs by chance. This threshold keeps these read-throughs in Tier 3, rather than report them as gene fusions in Tiers 1 and 2. A user can reduce this value to be more sensitive.

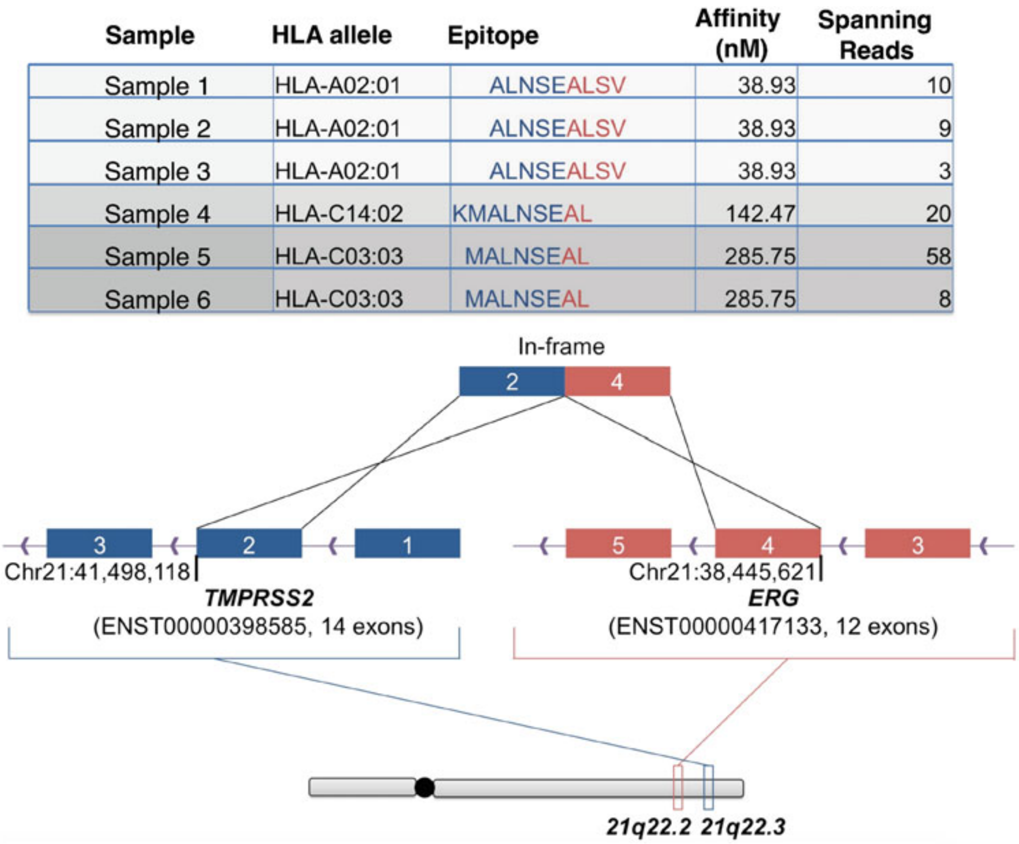
   -rt: default: 0, is the normal DNA–tumor DNA ratio. If the ratio is less than this value, then DNA reads from the normal DNA data set supporting a fusion candidate are ignored. The user may need to increase this value when many candidates are reported to have WGS reads supported from both the normal and tumor DNA samples. Otherwise, good gene fusion candidates could be ignored and marked as Tier 7.

5. Examine gene fusion candidates across the whole cohort of data. This step is optional, but could be very useful. When applying INTEGRATE to individual samples properly, usually a high sensitivity and specificity is expected. For a cohort of data, the real gene fusion candidates can only come from real gene fusion reads in the samples that harbor the gene fusions,

but number of false positive gene fusion candidates from different reasons can increase with the increase of the number of samples in the cohort. Therefore, if a gene fusion candidate has an unrealistically high frequency in the cohort than expected (e.g., way higher than the known representing gene fusions in the cancer type), the users would need to take a closer look at the candidates, or even perform manual inspection. Another advantage is that the bottom-line candidates (e.g., a read-through reported in one sample) may have better read support from other samples; therefore, a voting method could be considered to tease out these cases.

6. Run other INTEGRATE series tools (i.e., INTEGRATE-Neo and INTEGRATE-Vis). This step is optional. The mutational landscape of cancer genomes results in the production of tumor specific peptides recognizable by immune molecules. These so-called neoantigens can be exploited for personalized cancer immunotherapy [22]. NGS data has been used to discover tumor specific neoantigens [23], which relied on somatic missense mutation-based neoantigen discovery workflows (e.g., pVAC-Seq) [24]. We developed the first open source pipeline, called INTEGRATE-Neo, for gene fusion neoantigen discovery using NGS data. We apply INTEGRATE-Neo to the TCGA prostate cohort data (PRAD) to demonstrate its utility for identifying gene fusion neoantigens that may serve as personalized cancer immunotherapy targets. We identified 15% of gene fusions in the cohort have epitopes with binding affinity scores ≤500 nM, thus are predicted as gene fusion neoantigens (Fig. 6). The users can use the INTEGRATE-Neo pipeline for discovering gene fusion neoantigens, that may serve as personalized cancer immunotherapy targets. INTEGRATE-Neo can be downloaded at https://github.com/ChrisMaherLab/INTEGRATE-Vis.

   We also developed a gene fusion visualization tool, called INTEGRATE-Vis, that generates comprehensive, highly customizable, publication-quality graphics focused on annotating each gene fusion at the transcript- and protein-level and assessing expression within an individual sample or across a patient cohort. INTEGRATE-Vis provide functions to generate visualizations of the gene fusion transcript isoforms, the predicted protein structure of the gene fusion, RNA-Seq read coverage across each gene fusion partner to reveal changes in exon expression, and expression of each gene fusion partner across a cohort of cancer and normal samples (Fig. 7). These functions will help the users to tease out false positive gene fusion candidates, predict the function of the gene fusions, and prioritize gene fusions for further mechanistic studies. All the INTEGRATE series tools support the SMC-RNA BEDPE



Sample	HLA allele	Epitope	Affinity (nM)	Spanning Reads
Sample 1	HLA-A02:01	ALNSEALSV	38.93	10
Sample 2	HLA-A02:01	ALNSEALSV	38.93	9
Sample 3	HLA-A02:01	ALNSEALSV	38.93	3
Sample 4	HLA-C14:02	KMALNSEAL	142.47	20
Sample 5	HLA-C03:03	MALNSEAL	285.75	58
Sample 6	HLA-C03:03	MALNSEAL	285.75	8

Fig. 6 Epitopes predicted for a gene fusion transcript. The fusion transcript is between exon 2 of the *TMPRSS2* transcript (blue) with Ensembl Id ENST00000398585 and exon 4 of the *ERG* transcript (red) with Ensembl Id ENST00000417133. It is in-frame at the fusion junction. Six samples with different HLA alleles are shown in the figure, with epitopes predicted with varying binding affinities. More samples are available from the INTEGRATE-Neo paper

format; therefore, the users can directly apply these methods after running INTEGRATE on their data.

4 Notes

1. *Garbage in, garbage out–finished execution of INTEGRATE does not mean a successful or optimized analysis.* In addition to the format of the split-reads in the BAM file, there were more situations that a user may have a completed execution of INTEGARE that seemed to be successful but in fact was not the best or even completely failed in discovering gene fusions. INTEGRATE was implemented to be robust; therefore, we have not seen INTEGRATE crashed on any BAM files after >10,000 jobs performed by ourselves (Other than occasionally we did not reserve enough memory and had to run the tool again with more memory. Depending on the size of the input

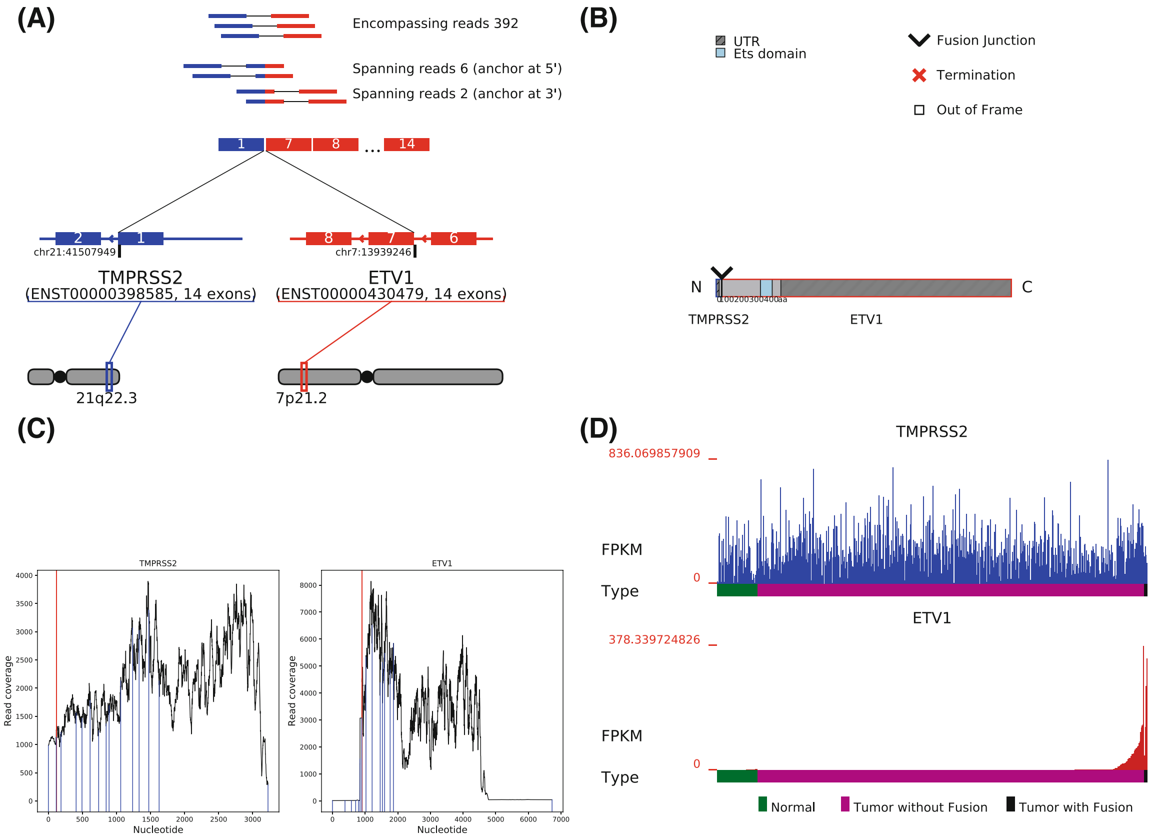


Fig. 7 A comprehensive gene fusion visualization. INTEGRATE-Vis output illustrated using the *TMPRSS2-ETV1* gene fusion in prostate cancer. INTEGRATE-Vis outputs four visualizations including: (**a**) gene fusion transcript isoforms, (**b**) the predicted protein structure of the gene fusion (with the complete ETS domain), (**c**) RNA-seq read coverage across each gene fusion partner to reveal changes in exon expression, and (**d**) expression of each gene partner across the TCGA PRAD cohort (The tumor without fusion and tumor with fusion categories are with regard to the *TMPRSS2-ETV1* gene fusion. Other *ETV1*gene fusion also upregulate *ETV1*gene)

BAMs, usually ~32 GB of memory is sufficient. If a std::bad_alloc error is encountered, you can increase the reserved memory using a job submission system). Some users provided BAM files from a read alignment tool that does not support read pairs that are not properly aligned (i.e., with long insert size or to two different chromosomes, e.g., TopHat1); therefore, virtually no gene fusion candidates were reported. In other cases, some data sets in the public domain were shared reflecting the applications of the original manuscripts (e.g., reads properly aligned to each gene for gene expression analysis, or gene fusion discovery using encompassing reads only); therefore, most of the gene fusions will be missed. (There may be a small portion of gene fusion reads that are still included in these data sets.) Taken together, it is very likely that any BAM files fed to INTEGRATE would result in finished executions, and certain amount of reported gene fusion candidates. This is not necessarily the sign of a successful and optimized gene fusion analysis. We recommend the users to follow the procedures provided in this book chapter, and always test with sample data as positive controls before applying their pipeline to large cohort of data.

2. *Prerequisite tools for INTEGRATE.* In the INTEGRATE package, there is a subdirectory, called *vendor*, which already included the *SAMtools* and *libdivsufsort* packages and companion code to automatically compile and install these tools. Therefore, this leaves the only required prerequisite to be the CMake tool. Advanced users can choose to tailor on the versions of the *SAMtools* and *libdivsufsort* packages they preferred. All other users can directly go to Step 3 in the Method section for installing the CMake tool.

3. *Docker image of INTEGRATE.* Since version 0.2.6, we provide docker images for the updated versions of INTEGRATE. The users can access the Docker image at https://bub.docker.com/r/jinwashu/integrate. The Docker pull command is:

```
$ docker pull jinwashu/integrate[:tag]
```

   Run the following commands using the INTEGRATE Docker image to get help information:

```
$ docker run jinwashu/integrate[:tag] Integrate
$ docker run jinwashu/integrate[:tag] Integrate mkbwt
$ docker run jinwashu/integrate[:tag] Integrate fusion
```

   where :tag is optional. An example of the tag is 0.2.6.

4. *Compile outside of source code.* You can choose any name for the folder to build INTEGRATE instead of using the name of INTEGRATE-build as in Subheading 3. For advanced users, the option of -DCMAKE_BUILD_TYPE=release can be changed to -DCMAKE_BUILD_TYPE=debug for tuning INTEGRATE on your platform.

5. *Make sure the reference genome and BAM files are consistent.* INTEGRATE does not require the input file for the human reference genome to be exactly the one that was used to generate the input BAM files. However, the build numbers need to be consistent. For example, if the reference genome is GRCh38 v90, the BAM files could have been generated using GRCh38 v85 or hg19 (INTEGRATE has internal code to convert, e.g., chr1 to 1, vice versa), but not GRCh37 or hg18. If you are downloading other versions of the human reference genome from Ensembl, you may need to log into the FPT site and figure out the exact paths. Just updating the version number

in the command lines may not always work. The symbols of {1..22,X,Y,MT} represent 25 independent commands, each for one chromosome or mitochondria (e.g., for chromosome 1, change {1..22,X,Y,MT} to 1 in the command).

6. *More about building indices.* You can choose any folder you like other than *bwts*, as long as the path exists. This whole step takes about 20–30 minutes, but only needs to run once. When you are running the INTEGRATE fusion tool to discover gene fusions, the path to this folder will be needed as an input. There is a parameter -mb (default: 10,000,000), which will exclude all short sequences in the human reference genome and keep only chromosomes 1 through 22, X, Y, and MT. User can lower the threshold if additional reference sequences are needed. This parameter will also be used when the INTEGRATE *fusion* tool is loading BWT indices.

7. *Creating gene annotation files from different databases using UCSC Genome Browser.* Here we show the methods of getting gene annotation files bases on several popular databases using the UCSC Genome Browser, including UCSC genes, Ensembl genes, Gencode genes, and RefSeq genes. Since INTEGRATE version 0.2.5, the annotation file uses 11 columns. The following examples reflect this update. Essentially, two additional columns cdsStart and cdsEnd are added to the previous version of the annotation file to support newer versions of the INTEGRATE tool. Please refer to the annot.enseml.txt annotation file at https://sourceforge.net/projects/integrate-fusion/files/ as an example.

   Examples of generating a gene annotation for INTEGRATE from the UCSC Genome Browser are as follows:

   (a) Go to http://genome.ucsc.edu/, click "Tools," and select "Table browser."

   (b) Click "click here" at "To reset all user cart settings (including custom tracks)."

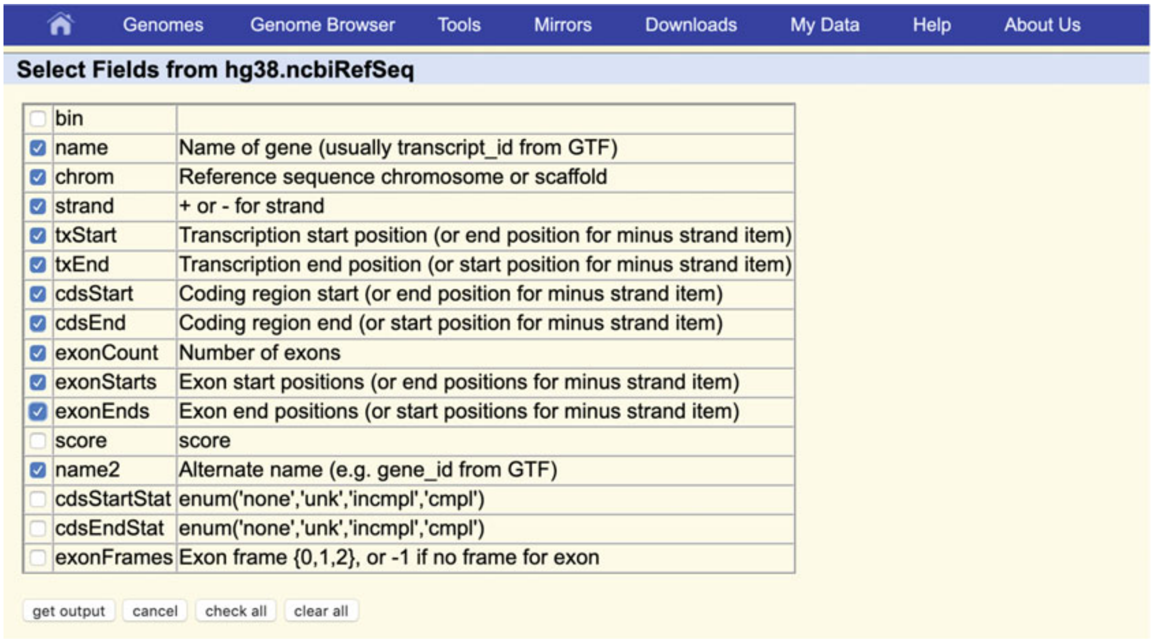
   (c) Select the version of human reference genome at the option of "assembly."

   (d) Choose the track you like the most (NCBI RefSeq, UCSC Genes, GENCODE V29, Ensembl 76, etc.)

   (e) Choose "selected fields from primary and related tables" as the output format. Enter an output file name (e.g., "annot.refseq.txt"). For some tracks, additional tables are needed for gene names.

   Example 1: Refseq Genes, choose the following and then click "get output" (Fig. 8).

   Example 2: Old UCSC Genes: choose the following and then click "get output" (Fig. 9).



Select Fields from hg38.ncbiRefSeq

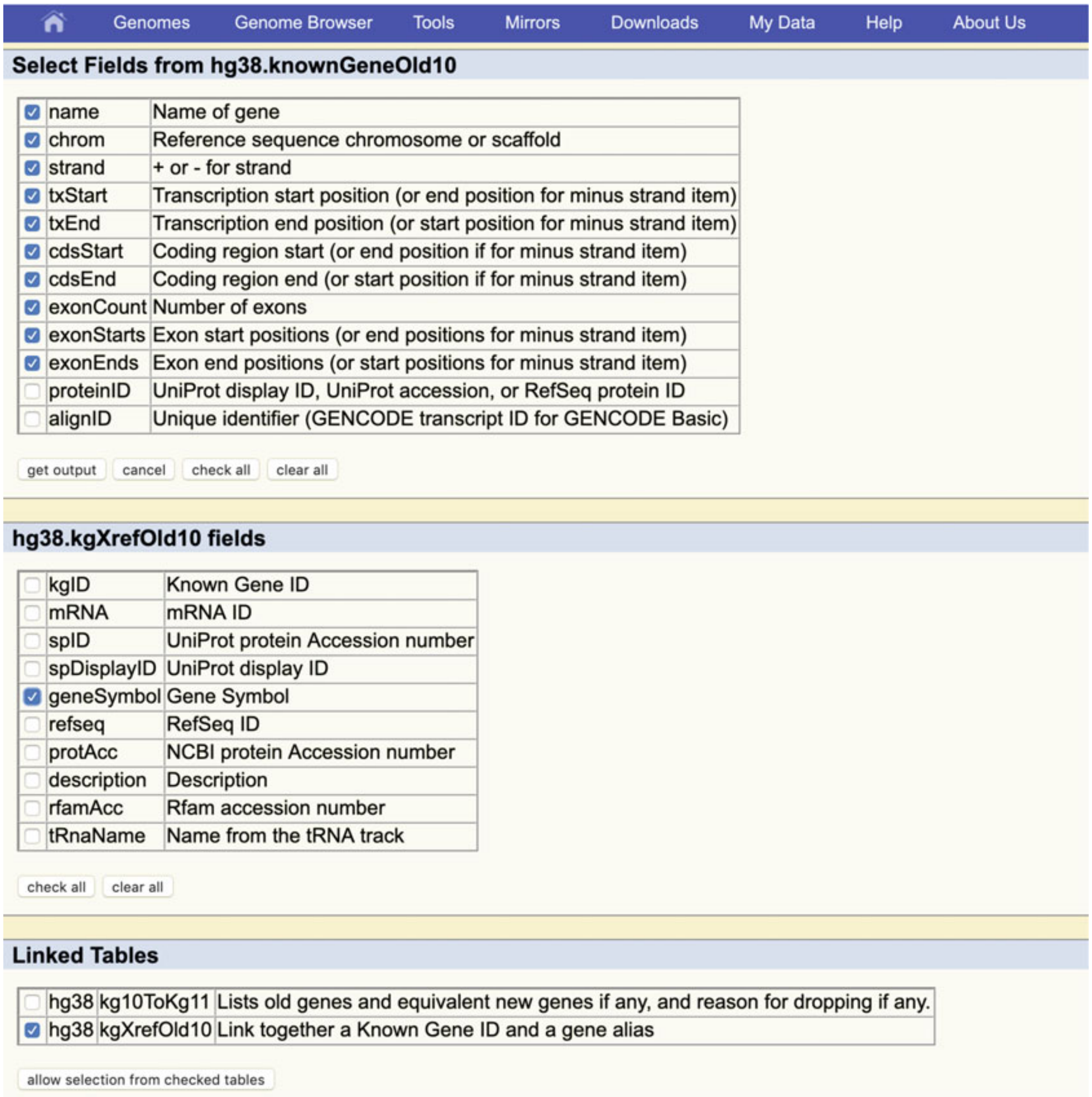
□ bin	
☑ name	Name of gene (usually transcript_id from GTF)
☑ chrom	Reference sequence chromosome or scaffold
☑ strand	+ or - for strand
☑ txStart	Transcription start position (or end position for minus strand item)
☑ txEnd	Transcription end position (or start position for minus strand item)
☑ cdsStart	Coding region start (or end position for minus strand item)
☑ cdsEnd	Coding region end (or start position for minus strand item)
☑ exonCount	Number of exons
☑ exonStarts	Exon start positions (or end positions for minus strand item)
☑ exonEnds	Exon end positions (or start positions for minus strand item)
□ score	score
☑ name2	Alternate name (e.g. gene_id from GTF)
□ cdsStartStat	enum('none','unk','incmpl','cmpl')
□ cdsEndStat	enum('none','unk','incmpl','cmpl')
□ exonFrames	Exon frame {0,1,2}, or -1 if no frame for exon

Fig. 8 Generating a gene annotation file using Refseq Genes

Example 3: GENCODE V29: choose the following and then click “get output” (Fig. 10).

Example 4: Ensembl 76 Genes: choose the following and then click “get output” (Fig. 11).

8. *Creating the equivalent of the accepted_hits.bam and unmapped.bam files using other multiple RNA-seq read-alignment tools.* Different read alignment tools may need different index files for the reference genome, and some of them may require gene annotation files. These usually come with the code or precompiled tools from the websites of the read-alignment tools. Alternatively, steps of generating the files are usually come with the manuals of the tools. We assume the users are already familiar with their favorable read-alignment tools and have installed and tested the tools. Here we provide sample command lines of three popular read alignment tools for generating the BAM files needed for INTEGRATE. This will serve as a useful guideline even the later versions of the software have updated options. The hg19 reference genome is used as an example in these command lines. The users can also change parameters according to the datasets and servers they use, including considering the read length, read depth, library type, etc.



Select Fields from hg38.knownGeneOld10

☑	name	Name of gene
☑	chrom	Reference sequence chromosome or scaffold
☑	strand	+ or - for strand
☑	txStart	Transcription start position (or end position for minus strand item)
☑	txEnd	Transcription end position (or start position for minus strand item)
☑	cdsStart	Coding region start (or end position if for minus strand item)
☑	cdsEnd	Coding region end (or start position if for minus strand item)
☑	exonCount	Number of exons
☑	exonStarts	Exon start positions (or end positions for minus strand item)
☑	exonEnds	Exon end positions (or start positions for minus strand item)
☐	proteinID	UniProt display ID, UniProt accession, or RefSeq protein ID
☐	alignID	Unique identifier (GENCODE transcript ID for GENCODE Basic)

hg38.kgXrefOld10 fields

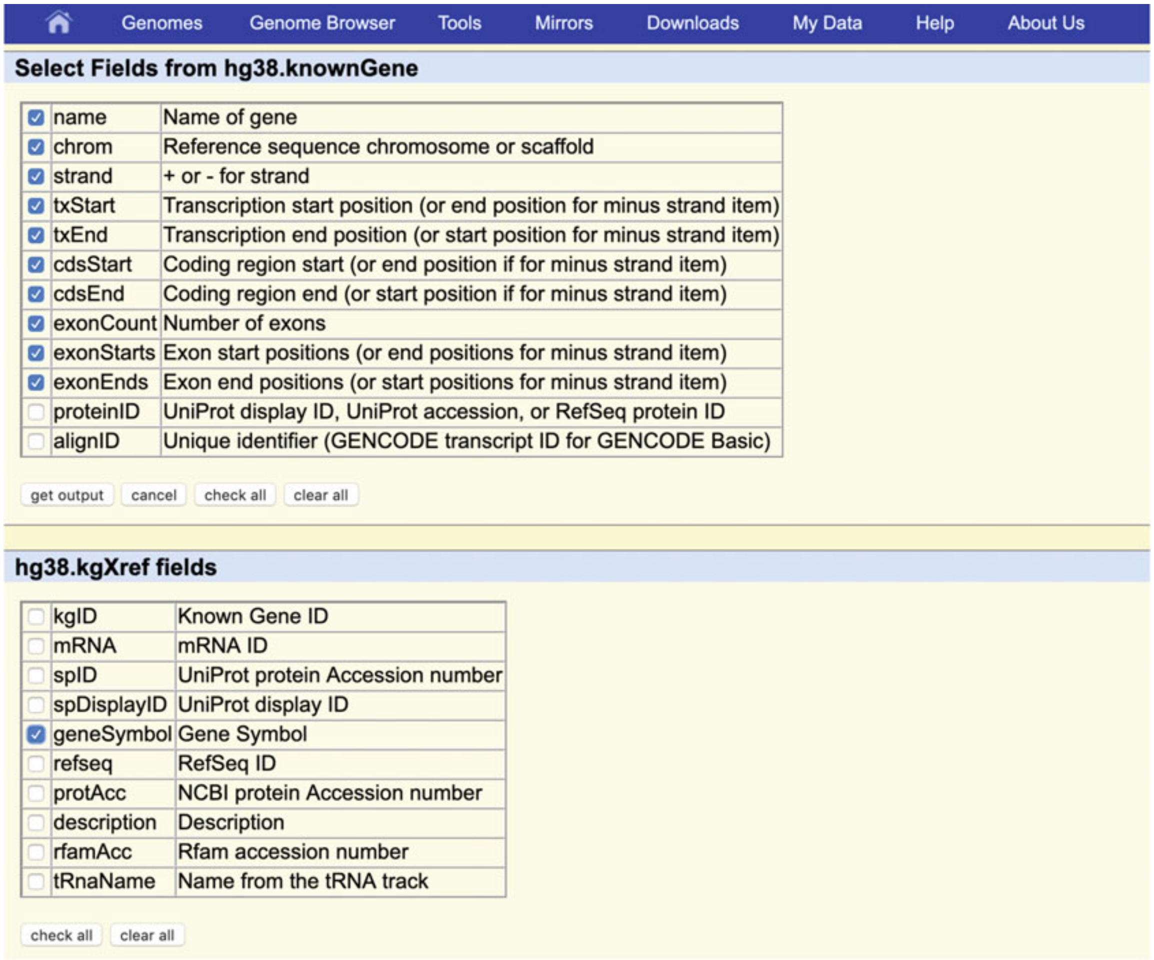
☐	kgID	Known Gene ID
☐	mRNA	mRNA ID
☐	spID	UniProt protein Accession number
☐	spDisplayID	UniProt display ID
☑	geneSymbol	Gene Symbol
☐	refseq	RefSeq ID
☐	protAcc	NCBI protein Accession number
☐	description	Description
☐	rfamAcc	Rfam accession number
☐	tRnaName	Name from the tRNA track

Linked Tables

☐	hg38	kg10ToKg11	Lists old genes and equivalent new genes if any, and reason for dropping if any.
☑	hg38	kgXrefOld10	Link together a Known Gene ID and a gene alias

Fig. 9 Generating a gene annotation file using UCSC Genes

```
$ gsnap -d hg19 -D path/to/hg19.gmap_index/ --format=sam
--nthreads=12 -s path/to/hg19.gmap_index/hg19/hg19.splice-
sites.iit
1.fastq 2.fastq
$ STAR --runThreadN 12 --genomeDir
/path/to/star_indices_overhang100/
--readFilesIn 1.fastq 2.fastq --outFileNamePrefix prefix --
chimOutType SeparateSAMold --
chimSegmentMin 18
$ tophat2 --library-type fr-unstranded --mate-inner-dist 254 --
mate-std-dev 50 --transcriptome-index=path/to/reference_se-
quence
-p 4 -o output path/to/reference_sequence.fa 1.fastq 2.fastq
```



Select Fields from hg38.knownGene

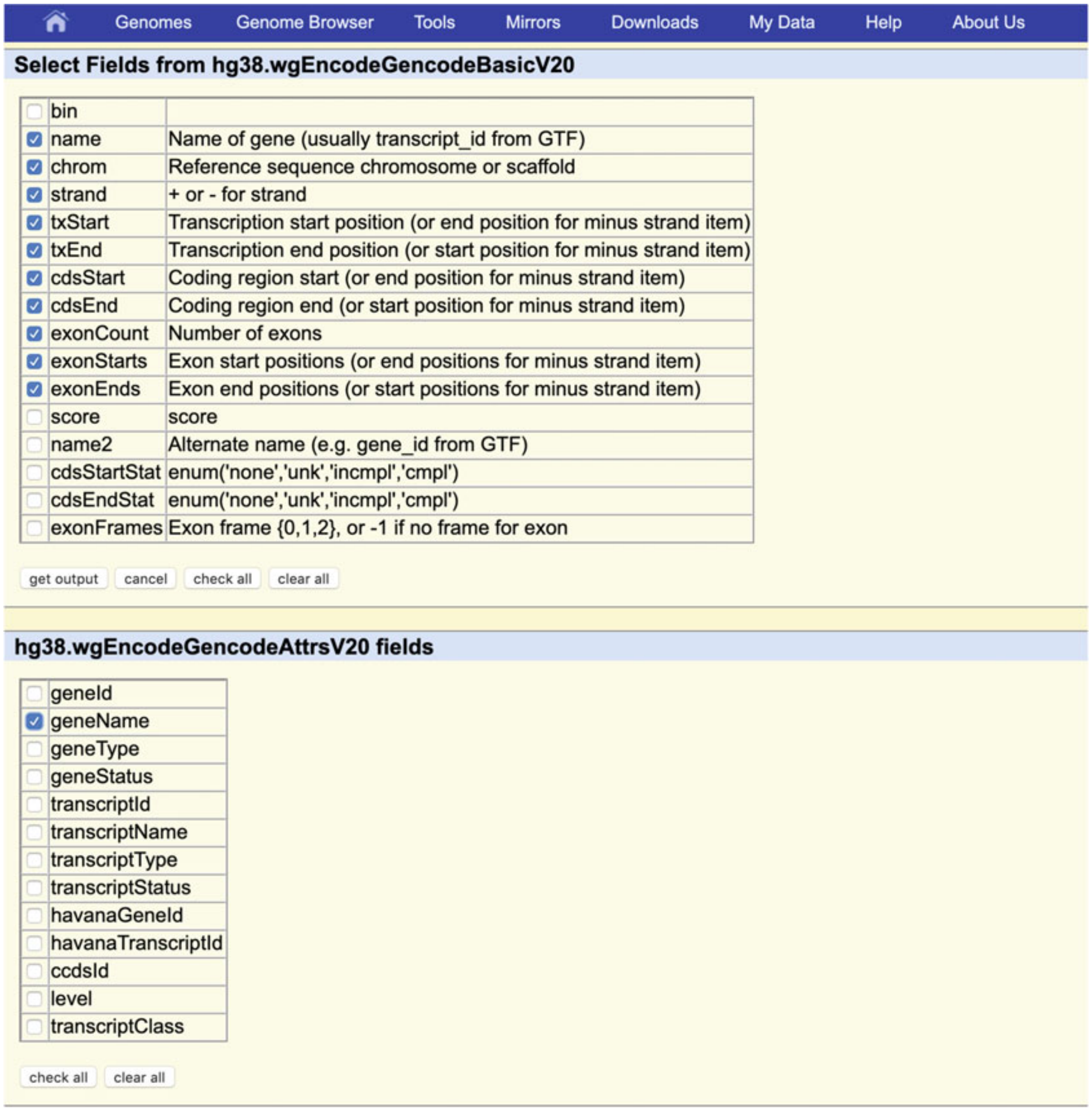
☑	name	Name of gene
☑	chrom	Reference sequence chromosome or scaffold
☑	strand	+ or - for strand
☑	txStart	Transcription start position (or end position for minus strand item)
☑	txEnd	Transcription end position (or start position for minus strand item)
☑	cdsStart	Coding region start (or end position if for minus strand item)
☑	cdsEnd	Coding region end (or start position if for minus strand item)
☑	exonCount	Number of exons
☑	exonStarts	Exon start positions (or end positions for minus strand item)
☑	exonEnds	Exon end positions (or start positions for minus strand item)
□	proteinID	UniProt display ID, UniProt accession, or RefSeq protein ID
□	alignID	Unique identifier (GENCODE transcript ID for GENCODE Basic)

hg38.kgXref fields

□	kgID	Known Gene ID
□	mRNA	mRNA ID
□	spID	UniProt protein Accession number
□	spDisplayID	UniProt display ID
☑	geneSymbol	Gene Symbol
□	refseq	RefSeq ID
□	protAcc	NCBI protein Accession number
□	description	Description
□	rfamAcc	Rfam accession number
□	tRnaName	Name from the tRNA track

Fig. 10 Generating a gene annotation file using GENCODE Genes

By default, TopHat2 will generate a BAM file called accepted_hits.bam and a BAM file called unmapped.bam. The accepted_hits.bam file contains aligned reads, and unmapped.bam contains all other reads. INTEGRATE takes the two BAM files as input files in the order of accepted_hits.bam and unmapped. bam. When using GSNAP, only one BAM/SAM that includes both aligned and unaligned reads will be reported. The users can provide INTEGRATE with the path to the BAM file twice in consecutive order. This will instruct INTEGRATE to search for aligned reads from the BAM file in the first occurrence of the path to the file, and search for unaligned reads from the BAM file in the second occurrence of the path to the file. For STAR, the user can provide INTEGRATE with the path to *prefix*.Chimeric.out.bam (sorted and indexed) twice in consecutive order.



Select Fields from hg38.wgEncodeGencodeBasicV20

☐	bin	
☑	name	Name of gene (usually transcript_id from GTF)
☑	chrom	Reference sequence chromosome or scaffold
☑	strand	+ or - for strand
☑	txStart	Transcription start position (or end position for minus strand item)
☑	txEnd	Transcription end position (or start position for minus strand item)
☑	cdsStart	Coding region start (or end position for minus strand item)
☑	cdsEnd	Coding region end (or start position for minus strand item)
☑	exonCount	Number of exons
☑	exonStarts	Exon start positions (or end positions for minus strand item)
☑	exonEnds	Exon end positions (or start positions for minus strand item)
☐	score	score
☐	name2	Alternate name (e.g. gene_id from GTF)
☐	cdsStartStat	enum('none','unk','incmpl','cmpl')
☐	cdsEndStat	enum('none','unk','incmpl','cmpl')
☐	exonFrames	Exon frame {0,1,2}, or -1 if no frame for exon

hg38.wgEncodeGencodeAttrsV20 fields

☐	geneId
☑	geneName
☐	geneType
☐	geneStatus
☐	transcriptId
☐	transcriptName
☐	transcriptType
☐	transcriptStatus
☐	havanaGeneId
☐	havanaTranscriptId
☐	ccdsId
☐	level
☐	transcriptClass

Fig. 11 Generating a gene annotation file using Ensembl Genes

9. *INTEGRATE requires soft-clipping for the BAMs for WGS reads*. By default, BWA turns on soft-clipping for split-reads. Make sure soft-clipping is turned on if using other tools. Also make sure the format of representing soft-clipped reads is the same as BWA *aln* and *sampe*.

10. *INTEGRATE requires all BAM files be sorted and indexed*. The best practice will be running SAMtools' *sort* function before running the *index* function and feed the sorted and indexed BAM files to INTEGRATE. By default, many but not all the read alignment tools report sorted BAM files. If the read alignment tool reports SAM files instead of BAM files, please

run SAMtools's *view* function and create the BAM files. The users can use Sambamba [25] as an alternative and more efficient software to substitute SAMtools. Sambamba can be downloaded at http://lomereiter.github.io/sambamba/.

11. *Tumor WGS data is expected when providing matched normal WGS data*. Please do not omit BWA.tumor.WGS.sort.bam and append BWA.normal.WGS.sort.bam at the same time, which will trick INTEGRATE to take BWA.normal.WGS.sort.bam as the BAM file for WGS reads from the tumor sample.

12. *The fields of the "summary.txt" file reflect the input data types*. If running with only RNA-seq data, number of encompassing and spanning RNA-seq reads will be appended as two columns. When using RNA-seq and WGS data from the tumor sample, numbers of encompassing RNA-seq reads, spanning RNA-seq reads, encompassing WGS reads, and spanning WGS reads will be appended. If normal WGS reads are also provided, numbers of encompassing WGS reads from the normal DNA sample, and spanning WGS reads from the normal DNA sample will the appended.

References

1. Hanahan D, Weinberg RA (2011) Hallmarks of cancer: the next generation. Cell 144 (5):646–674
2. Mertens F et al (2015) The emerging complexity of gene fusions in cancer. Nat Rev Cancer 15(6):371–381
3. White NM, Feng FY, Maher CA (2013) Recurrent rearrangements in prostate cancer: causes and therapeutic potential. Curr Drug Targets 14(4):450–459
4. Melo JV et al (1993) The ABL-BCR fusion gene is expressed in chronic myeloid leukemia. Blood 81(1):158–165
5. Maher CA et al (2009) Transcriptome sequencing to detect gene fusions in cancer. Nature 458(7234):97–101
6. Maher CA et al (2009) Chimeric transcript discovery by paired-end transcriptome sequencing. Proc Natl Acad Sci U S A 106 (30):12353–12358
7. Iyer MK, Chinnaiyan AM, Maher CA (2011) ChimeraScan: a tool for identifying chimeric transcription in sequencing data. Bioinformatics 27(20):2903–2904
8. Carrara M et al (2013) State-of-the-art fusion-finder algorithms sensitivity and specificity. Biomed Res Int 2013:340620
9. Zhang J et al (2016) INTEGRATE: gene fusion discovery using whole genome and transcriptome data. Genome Res 26(1):108–118
10. Li S et al (2013) Endocrine-therapy-resistant ESR1 variants revealed by genomic characterization of breast-cancer-derived xenografts. Cell Rep 4(6):1116–1130
11. Lei JT et al (2018) Functional annotation of ESR1 gene fusions in estrogen receptor-positive breast cancer. Cell Rep 24 (6):1434–1444 e7
12. Griffith OL et al (2016) A genomic case study of mixed fibrolamellar hepatocellular carcinoma. Ann Oncol 27(6):1148–1154
13. Griffith M et al (2016) Comprehensive genomic analysis reveals FLT3 activation and a therapeutic strategy for a patient with relapsed adult B-lymphoblastic leukemia. Exp Hematol 44(7):603–613
14. Kim J et al (2018) MYBL1 rearrangements and MYB amplification in breast adenoid cystic carcinomas lacking the MYB-NFIB fusion gene. J Pathol 244(2):143–150
15. Pareja F et al (2019) Assessment of HMGA2 and PLAG1 rearrangements in breast adenomyoepitheliomas. NPJ Breast Cancer 5:6
16. Zhang J, Mardis ER, Maher CA (2017) INTEGRATE-neo: a pipeline for personalized gene fusion neoantigen discovery. Bioinformatics 33(4):555–557

17. Zhang J, Gao T, Maher CA (2017) INTEGRATE-Vis: a tool for comprehensive gene fusion visualization. Sci Rep 7(1):17808

18. Wu TD, Nacu S (2010) Fast and SNP-tolerant detection of complex variants and splicing in short reads. Bioinformatics 26(7):873–881

19. Kim D et al (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol 14(4):R36

20. Dobin A et al (2013) STAR: ultrafast universal RNA-seq aligner. Bioinformatics 29(1):15–21

21. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25(14):1754–1760

22. Heemskerk B, Kvistborg P, Schumacher TN (2013) The cancer antigenome. EMBO J 32 (2):194–203

23. Carreno BM et al (2015) Cancer immunotherapy. A dendritic cell vaccine increases the breadth and diversity of melanoma neoantigen-specific T cells. Science 348 (6236):803–808

24. Hundal J et al (2016) pVAC-Seq: a genome-guided in silico approach to identifying tumor neoantigens. Genome Med 8(1):11

25. Tarasov A et al (2015) Sambamba: fast processing of NGS alignment formats. Bioinformatics 31(12):2032–2034