MetaFusion: A high-confidence metacaller for filtering and prioritizing RNA-seq gene fusion candidates.

Michael Apostolides[1], Yue Jiang[1], Mia Husić[1], Robert Siddaway[2], Cynthia Hawkins[2,3,4], Andrei L Turinsky[1], Michael Brudno[1,5,6,7], Arun K Ramani[1]*.

[1]Centre for Computational Medicine, The Hospital For Sick Children, Toronto, ON, Canada; [2]The Arthur and Sonia Labatt Brain Tumour Research Centre, The Hospital for Sick Children, Toronto, ON, Canada; [3]Department of Laboratory Medicine and Pathobiology, University of Toronto, Toronto, ON, Canada. [4]Division of Pathology, The Hospital for Sick Children, Toronto, ON, Canada. [5]Genetics and Genome Biology Program, The Hospital for Sick Children, Toronto, ON, Canada. [6]Department of Computer Science, University of Toronto, Toronto, ON, Canada. [7]University Health Network, Toronto, ON, Canada.

*To whom correspondence should be addressed.

Abstract

Motivation: Gene fusions are often associated with cancer, yet current fusion detection tools vary in their calling approaches, making selecting the right tool challenging. Ensemble fusion calling techniques appear promising; however, current options have limited accessibility and function.

Results: MetaFusion is a flexible meta-calling tool that amalgamates the outputs from any number of fusion callers. Results from individual callers are converted into Common Fusion Format, a new file type that standardizes outputs from callers. Calls are then annotated, merged using graph clustering, filtered and ranked to provide a final output of high confidence candidates. MetaFusion consistently outperformed individual callers with respect to recall and precision on real and simulated datasets, achieving up to 100% precision. Thus, an ensemble calling approach is imperative for high confidence results. MetaFusion also labels fusions found in databases using the FusionAnnotator package, and is provided with a benchmarking toolkit to calibrate new callers.

Availability: MetaFusion is freely available at https://github.com/ccmbioinfo/MetaFusion

Contact: arun.ramani@sickkids.ca

1. Introduction

Gene fusions, or hybridizations between two independent wildtype genes, are recognized as an important class of genomic alteration, particularly in cancer. They arise most frequently from chromosomal rearrangements, though recent evidence indicates that they are also caused by mechanisms such as cis-splicing of adjacent genes (cis-SAGe) (Brien et al. 2019; Gao et al. 2018; Grosso et al. 2015; Hu et al. 2018; Li et al. 2008). Cancer-related fusions can lead to increased oncogene expression, decreased tumour suppressor expression, and the formation of oncogenic fusion proteins. Such changes are well-documented in the tumourigenesis of multiple cancers (Mertens et al. 2015; Mitelman et al. 2007; Xiao et al. 2018; Yoshihara et al. 2015), and it is estimated that fusions account for nearly 20% of human cancer morbidity (Gao et al. 2018; Mitelman et al. 2007). Fusions may serve as disease biomarkers, such as the breast cancer-specific *SCNN1A-TNFRSF1A* and *CTSD-IFITM10* (Varley et al. 2014), or as targets for treatments such as imatinib, a drug that inhibits tyrosine kinase activity of the causative *BCR-ABL* fusion in chronic myeloid leukemia (Druker 2008; Mitelman et al. 2007). Accurate identification of biologically relevant gene fusions in cancer is thus critical, as it can contribute to patient diagnosis and care in the rise of precision medicine.

Although a number of fusion calling tools are currently available, they can vary significantly from one another. Callers differ in the genomic regions they consider, the numbers of alignment steps they have, their read coverage requirements, filters, output formats and so on. Some may only prioritize specific types of fusions, such as those caused by chromosomal rearrangements, and consider all others to be transcriptional noise or part of normal cell biology. Such differences can give inconsistent results between methods for any single dataset, and lead to biologically relevant fusions being excluded from final outputs.

These problems are further compounded by individual fusion caller limitations, including low precision (high false positive rate) and sub-par recall. Poor precision can be caused by outdated filters, reliance on outdated databases of false positive fusions, or having overly lenient criteria for keeping reads. Benchmarking with only simulated data can also cause more false positives than expected when the caller is run on real cancer data. Regarding poor recall, some callers have stringent criteria for keeping reads, and low read depth on a true fusion may cause it to be missed. Often, a tool which performs well in one of these areas will fall short in another (Liu et al. 2016). Finally, fusion callers often provide outputs that are large and ambiguous (Haas et al. 2019), making it a challenge to prioritize biologically relevant fusions for experimental validation. This leaves users with the arduous task of determining which tools are best suited for their needs.

Ensemble approaches have shown promise in overcoming these challenges in various fields within genetics and gene expression (Aghaeepour et al. 2013; Lichtenberg et al. 2017; Yang and Deng 2020), and a study of 23 fusion callers has shown that using multiple tools leads to improved results (Haas et al. 2019). Yet any current means of ensemble fusion identification, or meta-calling, have been largely preliminary (Liu et al. 2016), and to our knowledge no robust approach has been developed. Software tools such as Pegasus (Abate et al. 2014) may standardize the interface of various callers, but do not merge their outputs effectively, making downstream analysis difficult. Chimera collates results from 10 callers, and can visualize junction coverage and predict the oncogenic potential of a given fusion (Beccuti et al. 2014); however, it is not a stand-alone meta-caller. Instead, it functions as a library that is compatible with the output files of only 10 pre-defined callers, making it limited and inflexible in its utility. Fusion search engine-based approaches such as FusionHub also exist but harness information in existing databases, as opposed to combining the results of various fusion callers (Panigrahi et al. 2018).

These challenges highlight a need for a single, flexible method that utilizes multiple approaches for fusion identification and evaluation to provide a concise high-confidence list of candidate fusions. We have thus developed MetaFusion, an ensemble fusion calling method that incorporates predictions from any number of callers, leverages the results of all tools included, filters out false positives for optimal precision, and ranks calls based on the number of contributing tools. MetaFusion takes fusion calls in a Common Fusion Format (CFF) - a novel file type that we have developed to standardize fusion caller outputs - and can be run on one's local machine using a Docker container with installed dependencies, and with around 7G of RAM, which is ideal for downstream analysis and visualization. MetaFusion dependencies can also be installed on one's machine, although using Docker is the preferred method. As it can incorporate any number of fusion callers, MetaFusion is highly adaptable and can be easily updated as new tools become available. Here, we use a series of simulated and real cancer datasets to show that MetaFusion consistently performs with high precision and recall, and provides high-confidence candidate fusions for further experimental validation.

2. Methods

2.1. MetaFusion Workflow

The MetaFusion workflow was developed to consolidate the outputs from various pre-existing fusion callers and to allow for further filtering, refining and benchmarking of those outputs (Figure 1).

Metafusion has been developed as a standalone tool, and the workflow to run the fusion callers has been implemented in GenPipes (Bourgey et al. 2018), an open-source, Python-based framework for -omics pipeline development and deployment. The current GenPipes implementation contains seven fusion callers, with MetaFusion downstream for consolidation and further analysis of their outputs. MetaFusion dependencies are available as a Docker image. The MetaFusion workflow is described in detail below.

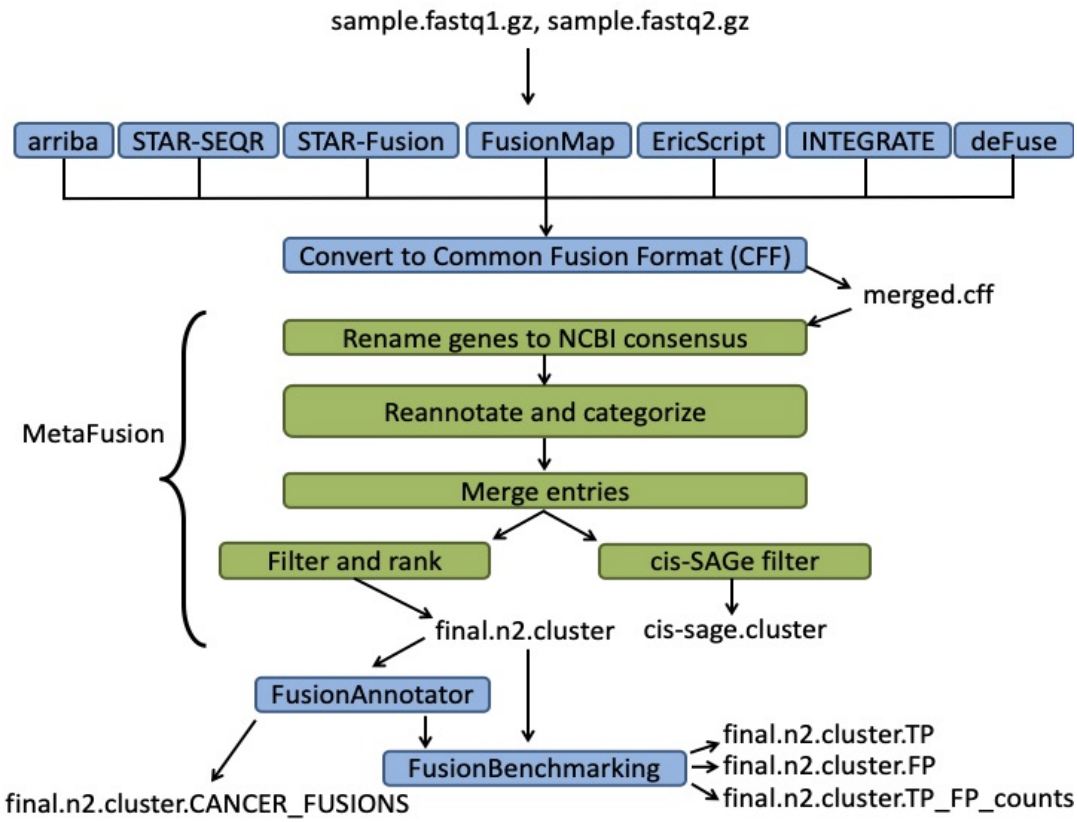

Figure 1: The MetaFusion workflow. MetaFusion consolidates outputs from various fusion callers. The callers are run independently on a given dataset and their outputs are converted into CFF files, which are used as input into the MetaFusion pipeline (green). This pipeline consists of renaming, reannotation, categorization, merging and filtering steps (see text for details). Afterwards, tools such as FusionAnnotator and the Benchmarking Toolkit may be used to further refine results.

2.1.1. Fusion calling and caller output conversion to CFF

For this study, we selected seven fusion callers to be used as input to MetaFusion: deFuse (McPherson et al. 2011), Arriba (Uhrig 2020), STAR-SEQR (STAR-SEQR 2020), STAR-Fusion (Haas et al. 2017), INTEGRATE (Zhang et al. 2016), EricScript (Benelli et al. 2012), and FusionMap (Ge et al. 2011). These callers were chosen because they are widely used in the literature and represent

a broad array of fusion calling approaches. Each of these callers is independently run on the input data to generate their own fusion calls.

All seven callers provide outputs in varying formats. To standardize this, we have developed the CFF file format (Supplementary Table 1A, 1C). Prior to the start of the MetaFusion workflow, a wrapper script converts fusion caller outputs into CFF. Separate sections of this script exist for each caller, where caller output file fields are mapped to CFF fields. Each line in a CFF file represents one fusion call by a given caller (Supplementary Table 1). Subsequent MetaFusion steps add further information to the CFF file, for example a unique fusion identifier (FID; e.g. F00000001) and fusion category (Figure 2).

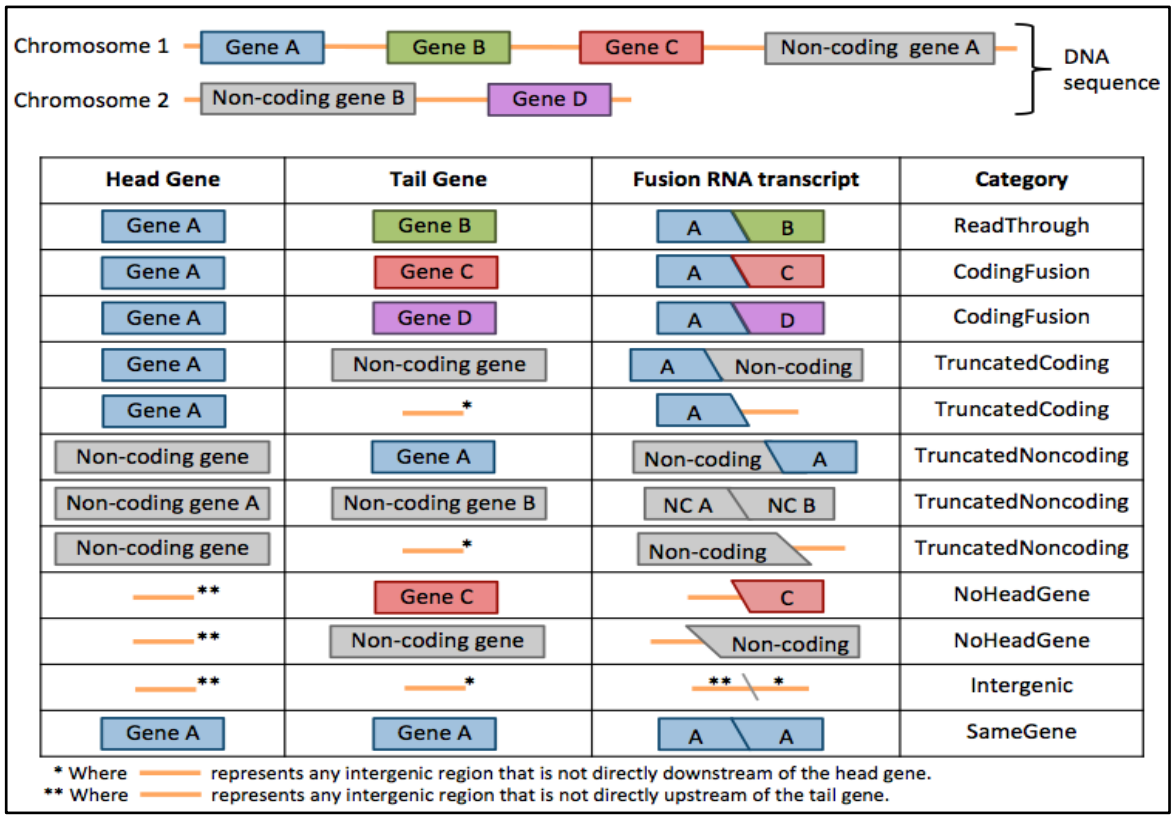

Figure 2: The seven categories used by MetaFusion to distinguish fusions. Two chromosomes containing coding (coloured) and non-coding (grey) genes are shown above, with intergenic sequences represented by orange lines between genes. The chart below shows head and tail genes, the resultant fusion RNA transcripts and the fusion categories that the transcripts would be assigned to. NC: non-coding.

Although these seven tools were used for optimization, any number and combination of fusion callers can be used as input into the MetaFusion pipeline, provided that caller results are converted into CFF. As a demonstration, we performed a subset of analyses with an additional caller, ChimeraScan (Iyer et

al. 2011), in a use-case to detect cis-SAGe ReadThrough fusions in prostate cancer. Analyses were done by either using all eight callers, or with a smaller subset of four callers found to have adequate ReadThrough recall (ChimeraScan, INTEGRATE, EricScript and deFuse) (Supplementary Figure 1).

2.1.2. Renaming of genes to current NCBI symbols

As gene names and symbols are regularly being updated and callers may vary in the database or GTF file they use to obtain gene name information, we have included a standardization step to rename the genes identified in fusions to the current NCBI symbols. This optimizes subsequent merging and benchmarking steps, as benchmarking of MetaFusion outputs relies on gene name matches to categorize fusions as either true or false positives (see "Benchmarking Toolkit" section below for more detail) .

A Python-based renaming script is used to assign current NCBI symbols to each CFF entry. This is done using information from the NCBI *Homo sapiens* gene alias file (*Homo_sapiens.gene_info.gz,* accessed May 7 2020), which is freely available on the NCBI FTP website (ftp://ftp.ncbi.nlm.nih.gov/gene). The script also removes delimiters and unusual characters from gene names. If the gene name is neither a known NCBI symbol, nor an alias of one, the original gene name is kept.

2.1.3. Reannotate and categorize

Once all gene names have been updated to current NCBI symbols, each fusion entry in the CFF is reannotated to include the following information:

(1) Each entry in the CFF is assigned a unique identifier (FID; e.g. F00000001)

(2) Breakpoints for each fusion are reannotated based on their intersection with genomic features (e.g. exon, intron, 5'UTR, 3'UTR, etc) in the gene annotation file. If multiple intersections occur for a given breakpoint, the genomic feature that matches the gene name is chosen.

(3) Each fusion entry in the CFF is assigned to one of seven categories, based on the coding status and adjacency of the fusion partners (Figure 2). For any given fusion, the upstream gene is referred to as the "head gene" and the downstream gene the "tail gene." A fusion's annotated category can be used to prioritize and filter fusion candidates.

2.1.4. Merging of fusion calls

After reannotation and categorization, fusion calls are merged using breakpoints and gene names (Figure 3). To consolidate fusion calls from multiple callers, we applied a graph-clustering algorithm,

in which nodes represent individual fusion calls from each caller, and edges represent intersections based on breakpoints, gene names or both. Breakpoints and gene names are both considered, as this allows for the most complete merging of calls (Figure 3A). Clustered fusion calls are then merged. All together, this process is done in four steps.

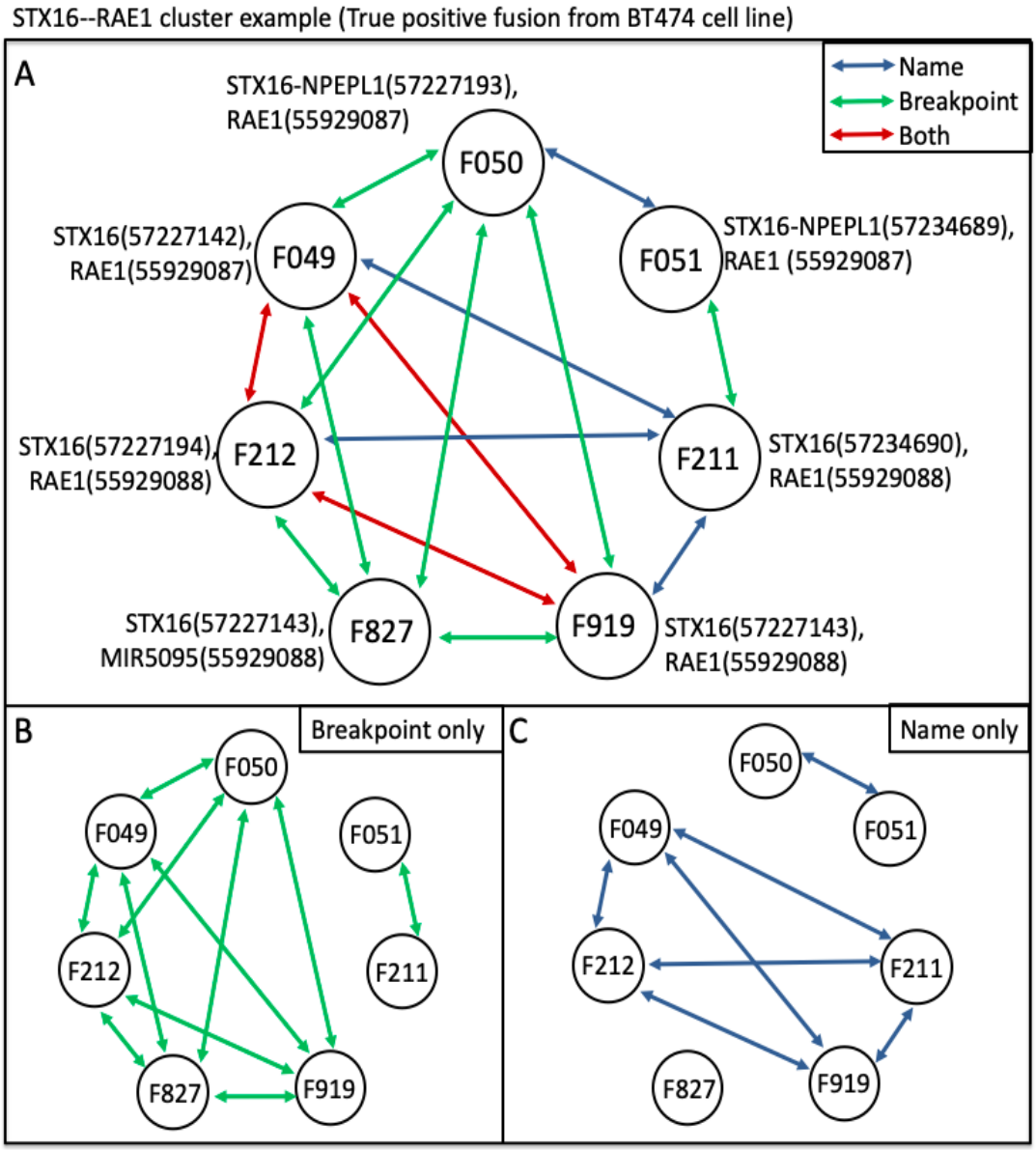

Figure 3: Diagram of the graph clustering approach used in MetaFusion. (A) MetaFusion relies on breakpoint (in parentheses) and gene name intersections to determine if multiple calls should be merged into one. Edges represent intersections based on breakpoint, gene name, or both. Edge colours indicate if calls are merged based on breakpoint (green), name (blue) or both (red). If only breakpoints **(B)** or only names **(C)** are considered for merging, calls may be merged incompletely or orphaned. The example depicted here is based on the true positive STX16-RAE1 fusion from the BT474 breast cancer cell line. FIDs correspond to those in Supplementary Table 1A, and corresponding CFF can be found in the test data .cff in test_data/cff/BRCA.cff.

First, intersection edges are generated using bedtools pairToPair function (Quinlan and Hall 2010) from within the "pybedtools" Python package. A BED file is generated in which every line is a separate fusion call. The file is then intersected with itself, and an edge is generated from each intersection. Breakpoints must be within 100 base-pairs of one another to be considered a match. An example of edges generated based on breakpoints is represented in Figure 3B.

Second, edges are generated by matching calls based on gene names. Names must be matching exactly. A hash table is created which uses head and tail gene names as keys, and collects FIDs which match the "Head--Tail" key (i.e. {key: [fusion_id list]}). Then, for each key, an edge is created for each FID pair in its corresponding FID list. An example of edges generated based on gene names are represented in Figure 3C.

Third, a graph is generated using the R package RBGL (Carey et al. 2020). Edges from both breakpoint intersections and gene name matches are combined, and a graph is constructed from the edge set. Since edges contain only FIDs, the graph is built unaware of which edges are due to gene name match or breakpoint intersections (Figure 3A). For each connected subgraph, a list of FIDs is generated which correspond to CFF file entries.

Fourth, the CFF entries are converted to cluster format, MetaFusion's final output format (Supplementary Table 1B). Delimiter-separated lists are generated for fields such as sample, tool and breakpoint, which may differ among merged calls.

2.1.5. Custom filters

After merging of fusion calls is complete, the resulting list is refined using a series of MetaFusion filters.

ReadThrough filter. ReadThrough fusions, or those consisting of two adjacent coding genes in which the head gene is immediately upstream of the tail gene, occur in healthy tissues (Babiceanu et al. 2016) and are more likely to be the result of cis-splicing than chromosomal rearrangements (Qin et al. 2015; Tang et al. 2017), making them seldom of interest in cancer research. As such, MetaFusion includes a *ReadThrough* filter, which removes entries categorized as ReadThrough fusions during reannotation. If they are of interest, ReadThrough fusions can still be viewed in a separate *cis-SAGe.cluster* file, decribed below.

CallerfilterN. This filter is used to remove fusions identified by fewer than N fusion callers. Fusions called by N or more callers are kept as part of the final output. The user can set the value of N to their desired threshold. Unless otherwise indicated, for this study we use *callerfilter2* (i.e. calls have to be

made by at least 2 callers). The value of N also determines the naming for the final output file (e.g. if callerfilter2 is used, the final output file will be named *final.n2.cluster*).

Blocklist filter. MetaFusion's *blocklist* filter is based on part of Arriba's blacklist file. This filter is used to block out known ReadThroughs, T-cell receptors, MHC complexes and immunoglobulins, as fusions involving these genes are less frequently of interest in cancer-specific research. This filter intersects the blacklist file's "recurrent breakpoints", "T-cell receptors" and "MHC complexes" paired regions with fusion breakpoints using bedtools pairToPair. Those which intersect are removed from the final output. Users can add a personal list of false positive fusions to this file (*blocklist_breakpoints.bedpe*) if they wish to remove them from their results.

Adjacent noncoding filter. The *adjacent noncoding* filter removes all fusions in the TruncatedCoding, TruncatedNonCoding and NoHeadGene categories whose constituent gene partners are within 100kb of one another, as we noticed that fusions with these characteristics are heavily represented among our negative control dataset (Supplementary Figure 2). Using this filter also removed false positives from our benchmarking datasets and did not affect true positive counts. These fusions are included in the *cis-SAGe.cluster file* described below. Since our negative control is designed to contain no fusions caused by chromosomal rearrangements (further detail in Datasets and Results sections), we developed this filter.

cis-SAGe.cluster file. While cis-SAGe fusions such as ReadThroughs are often considered a part of normal biology or transcriptional noise (Babiceanu et al. 2016; Tang et al. 2017), some occur uniquely in certain types of cancer, and can serve as disease biomarkers (Varley et al. 2014; Qin et al. 2014, 2016; Rickman et al. 2009). MetaFusion stores cis-SAGe fusions, such as ReadThroughs, in a separate *cis-SAGe.cluster* file, instead of simply discarding them. Not all cis-SAGe fusions are necessarily ReadThroughs, therefore this file also contains SameGene fusions as well as those flagged by the *adjacent noncoding* filter, as their breakpoint proximity and orientation may be indicative of cis-splicing. Fusions in the *cis-SAGe.cluster* file are removed from the *final.n2.cluster* output using the above-described filters.

Ranking calls by number of callers. Fusions called by multiple tools are more likely to be true fusions, and MetaFusion leverages this to provide a ranking of final calls. After the above filters are applied, the final output file is sorted by the number of tools that call each fusion. Fusions in the file are sorted by the number of contributing tools in descending order (highest first). This information can then be used to prioritize fusions for further analysis.

2.1.6. Benchmarking Toolkit

The Benchmarking Toolkit, developed by Haas and colleagues (2019), allows for benchmarking of MetaFusion outputs. When a caller combination different from that used in this study is chosen, MetaFusion should be benchmarked with the provided test data to ensure adequate performance (see Software Availability section, Github wiki for links to fastqs). This series of perl scripts has been modified slightly to include the unique FIDs provided by MetaFusion's reannotation step which allows MetaFusion output to be partitioned into separate true positive and false positive files. We also modified the Benchmarking Toolkit's *genes.coords.gz* file to include an additional 41,496 entries corresponding to loci with updated NCBI symbols in NCBI's most recent *Homo_sapiens.gene_info* file (accessed May 7th, 2020). This ensures that gene naming is consistent with the most up-to-date NCBI symbols and allows CFF outputs of individual callers to be benchmarked without the need to run them through MetaFusion's gene renaming step.

More information about the Benchmarking Toolkit can be found at https://github.com/fusiontranscripts/FusionBenchmarking/wiki.

2.1.7. FusionAnnotator

MetaFusion also integrates the FusionAnnotator tool (Haas et al. 2019), which annotates fusion calls with metadata from a number of cancer and normal fusion databases. It relies on gene name matches for this. An enrichment in cancer-related fusions may indicate that a workflow is prioritizing fusions of interest. FusionAnnotator output is used to add a comma-separated list of database hits to the column *cancer_db_hits* of the *final.n2.cluster* file.

2.2. Datasets

To evaluate the MetaFusion pipeline, we used a series of simulated and real cancer data (Table 1). All datasets were run through the MetaFusion pipeline, and are described below.

2.2.1. Curation and renaming of truth sets

All of the datasets used to evaluate MetaFusion performance contain truth sets, with the exception of the negative control. These are subsets of fusions that have been either intentionally created in simulated data or have been experimentally confirmed in real cancer data.

Since the benchmarking toolkit relies on gene name matches to identify true and false positives, it is important that the names of the genes involved in these truth set fusions follow NCBI consensus

naming. Although some benchmarking approaches might use breakpoints for benchmarking, this can be a challenge as it may involve unannotated regions of the genome where it is much more difficult to distinguish noise from biologically relevant events.

Upon renaming the truth set for sim50/101, 257/2500 fusions contained an outdated NCBI symbol in at least one of the fusion partners. For this reason, we have used the NCBI *Homo_sapiens.gene_info* file to update names to the most recent NCBI symbols in the truth sets. Truth sets were run through the renaming script in a separate step, independent of the MetaFusion workflow.

2.2.2. Simulated datasets

DREAM dataset. This dataset comprises the sim45 and sim52 datasets from round 3 of the SMC DREAM RNA challenge. The sim45 dataset, containing 30 true fusions, has 60 million reads, 101bp long with a 150-160 fragment size. The sim52 dataset, containing 31 true fusions, has 135 million reads, 101bp long with a fragment size of 150-160. Together, the DREAM dataset has a truth set of 61 fusions.

Negative control BEERS dataset. This dataset was created using the Benchmarker for Evaluating the Effectiveness of RNA-seq software (BEERS) simulator (Grant et al. 2011) by the authors of JAFFA (Davidson et al. 2015). It is designed to contain no fusions, which for the purposes of this dataset are defined as those occurring due to chromosomal rearrangements. It is used to identify fusion callers with high false positive rates, allowing users to exclude them from further analyses of real data. It is also used to determine characteristics of false positives, allowing for further refinement of results.

sim50 and sim101 datasets. Both the sim50 and sim101 simulated datasets were generated by Haas and colleagues using the Fusion Transcript Simulation Toolkit (Haas et al. 2019). The sim50 dataset has a truth set of 2500 fusions, and 50 base pair reads. The sim101 dataset has the same truth set as sim50, but has 101 base pair reads.

A summary of all datasets is provided in Supplementary Table 2

2.2.3. Real cancer datasets

Breast cancer (BRCA). This dataset consists of 53 experimentally validated fusions found in four breast cancer cell lines -- BT474, KPL4, MCF7, and SKBR3 (Edgren et al. 2011). It was previously used as a benchmarking dataset to evaluate the performances of 23 fusion callers (Haas et al. 2019). These samples were downloaded from the Broad Institute's Trinity index.

Melanoma. This dataset consists of six melanoma patient-derived samples (SRR018259 SRR018260 SRR018261 SRR018265 SRR018266 SRR018267) and two chronic myelogenous leukemia cell lines (SRR018268, SRR018269) (Supplementary Table 2) (Berger et al. 2010; Jia et al. 2013). It has a 17 fusion truth set, and was previously used for benchmarking of the SOAPfuse fusion caller (Jia et al. 2013).

Prostate cancer. This dataset contains 44 cis-SAGe fusions, most of which fall into the ReadThrough category (Kumar et al. 2016; Qin et al. 2015). This data is divided into 100nt read length (SRR1657556, SRR1657557) and 50nt read length (SRR1657558 SRR1657559, SRR1657560 and SRR1657561). Samples SRR1657557, SRR1657559 and SRR1657561 are siCTCF-treated, whereas SRR1657556, SRR1657558 and SRR1657560 are negative controls.

A summary of all datasets is provided in Supplementary Table 2

3. Results

3.1 Precision and recall of MetaFusion and individual callers

To evaluate MetaFusion's performance, we began by comparing its precision and recall to that of the seven individual callers we used. We selected three simulated datasets (DREAM, sim50, and sim101; Figure 4 A-C; Supplementary Table 4) and two real cancer datasets (BRCA and melanoma; Figure 4 D, E; Supplementary Table 5) for this comparison. The BRCA and melanoma datasets were selected as they have well-defined truth sets of validated fusions, and were both used as validation datasets in previous studies (Haas et al. 2019; Jia et al. 2013).

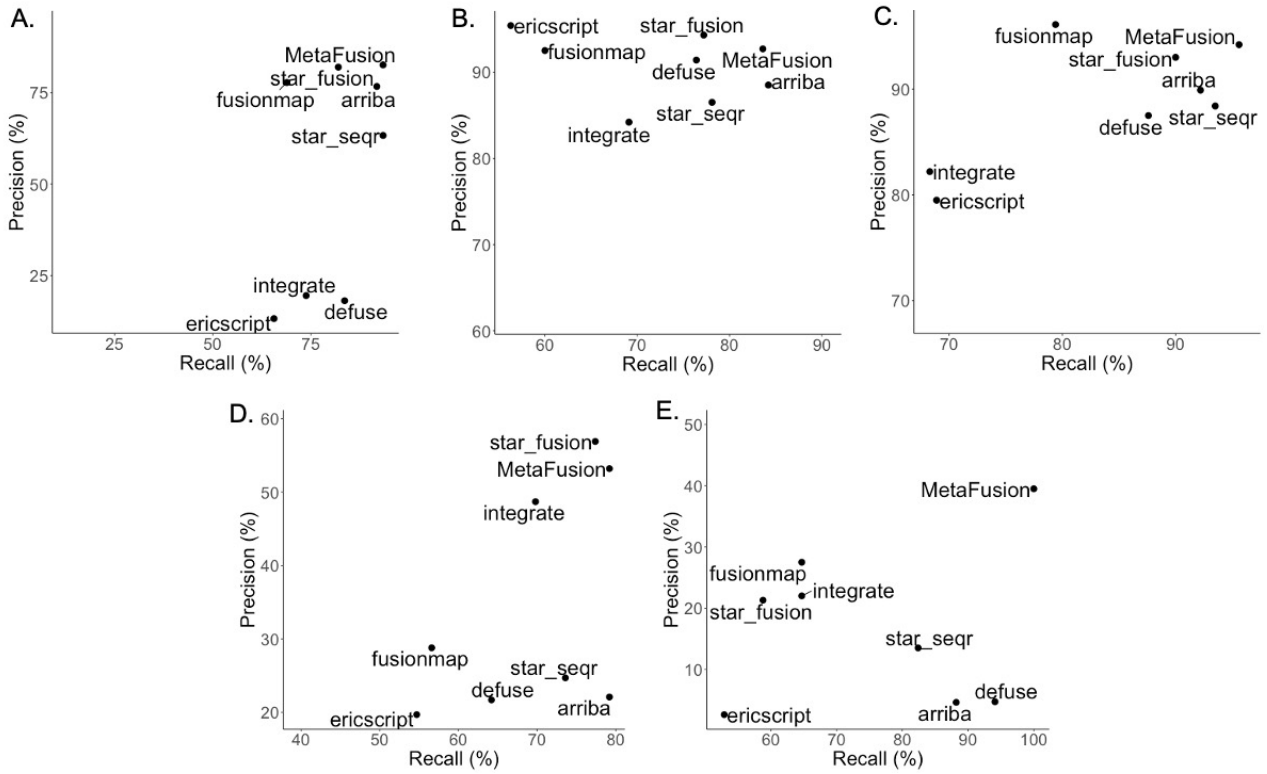

Figure 4: MetaFusion consistently outperforms individual fusion callers on simulated datasets. Precision and recall plots showing performance of seven fusion callers and MetaFusion for the DREAM **(A)**, sim50 **(B)** and sim101 simulated datasets **(C),** as well as the BRCA **(D)** and melanoma **(E)** real cancer datasets. MetaFusion generally outperforms individual callers across all three simulated datasets. It performs comparably to STAR_fusion on the BRCA data and outperforms all callers on the melanoma data. It was the only tool to identify all 14 true positive melanoma fusions and had the highest F1 score

We analyzed the five datasets using each of the seven callers with default parameters, then calculated precision, recall and F1 scores of each caller. The caller outputs were run jointly through MetaFusion (see Methods for filter settings), and precision, recall and F1 scores were calculated.

Counting false positive calls was done on a per-sample basis. For example, if fusion GeneA-GeneB is a false positive present in three samples, this will be counted as three separate false positives. It should, however, be noted that MetaFusion represents such GeneA-GeneB fusion as one row in the *final.n2.cluster* output file (with the names of the affected samples shown in the corresponding column), hence the number of entries in the final output file may be fewer than the sum of true and false positive calls.

MetaFusion generally outperforms individual tools for all five datasets, as indicated by precision, recall and F1 measurements (Figure 4; Supplementary Table 4, 5). In instances where either the precision or the recall of an individual caller is greater than that of MetaFusion, MetaFusion's F1 score is often higher, indicating better overall performance. For example, both EricScript and STAR_fusion

have greater precision than MetaFusion for the sim50 dataset, yet MetaFusion has a superior F1 score, due to its improved recall compared to either of these tools. MetaFusion's categorization, merge function and custom filters thus result in a refined list of higher-confidence calls which can be used to prioritize fusion candidates for Sanger or PCR validation.

Moreover, MetaFusion performs favourably even though our combination of callers included those with generally poor performance with both real and simulated datasets, such as EricScript. This is because false positive calls tend to be uncorrelated among methods, and are removed by MetaFusion's filters and joint calling approach. Thus MetaFusion can be used to improve upon callers with lower performance to provide high confidence candidate fusions.

Importantly, MetaFusion outperforms individual fusion callers on both BRCA and melanoma real cancer datasets (Figure 4 D, E; Supplementary Table 5). It should be noted that fusion calling tools generally have poorer performance on real datasets compared to simulated data, and this trend remains with MetaFusion, as it relies on the final results of these tools for its input. Furthermore, complete truth sets for real cancer data cannot be known with certainty, and are often amalgamated from various sources in the literature (Asmann et al. 2011; Edgren et al. 2011; Kangaspeska et al. 2012; Maher et al. 2009). It is thus possible, and even likely, that fusions labelled as false positives in these samples are not yet part of the known truth set (Haas et al. 2019). This may explain why the precision of MetaFusion is lower for the BRCA and melanoma datasets compared to simulated data, and why in the melanoma dataset a higher ratio of false to true positives is detected.

It is also worth noting that unfiltered MetaFusion output from all seven callers detects 47/53 BRCA true positives. The precision, however, is only ~9% (47/528). When MetaFusion filters are applied, four of these true positives are filtered out because they are supported by only one tool (not the same tool in each case), while the fifth is a ReadThrough that is instead stored in the *cis-SAGe.cluster* file. Interestingly, the four fusions called by one tool have junction and spanning read counts between 0-5 reads, substantially lower than the read counts for the other 42 true positives (mean split=39, span=40). This suggests that lower coverage fusions are more likely to be missed by callers.

3.2 Negative control benchmarking with BEERS dataset

Continuing our benchmarking, we used the BEERS negative control dataset to compare MetaFusion's performance with that of individual callers with respect to false positives. MetaFusion identified five false positive fusions, and was only outperformed by FusionMap, which identified four

(Figure 5). In comparison, deFuse and EricScript called over 200 false positives each (275 and 258, respectively). All false positives identified with MetaFusion were in the CodingFusion category.

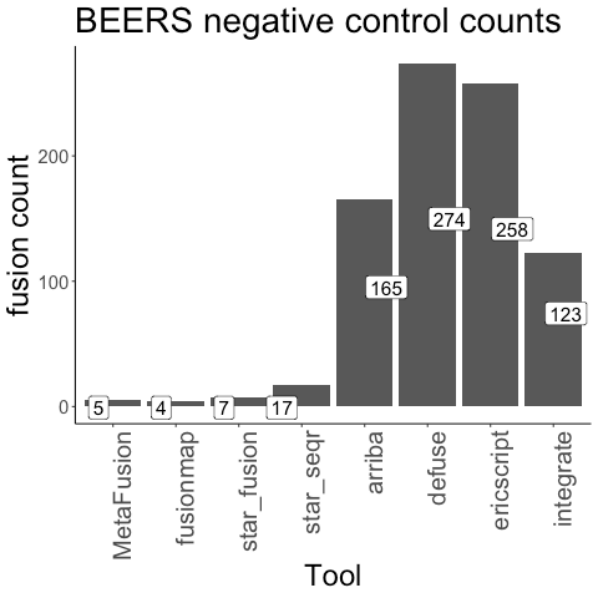

Figure 5: MetaFusion has a low false positive detection rate compared to most individual callers. The BEERS negative control dataset was used to evaluate the detection of false positive calls by MetaFusion and seven individual callers. MetaFusion detected five false positives, outperforming all callers except FusionMap. Most callers identified a significantly larger number of false positives, particularly deFuse and EricScript.

3.3 cis-SAGe file

MetaFusion was designed to be a versatile and flexible tool that facilitates ensemble fusion calling of various fusion types for a range of applications. This is in contrast to many currently available fusion calling tools, which may favour specific fusion types. In particular, many tools filter out cis-SAGe fusions, in which neighbouring genes are transcribed into a single pre-mRNA (Qin et al. 2015), and prioritize CodingFusions, which occur between non-adjacent coding genes and are typically caused by chromosomal rearrangements. Yet some cis-SAGe fusions, such as *SCNN1A-TNFRSF1A* and *CTSD-IFITM10,* are translated into fusion proteins, and can contribute to cancer progression or serve as disease biomarkers (Varley et al. 2014). Instead of discarding these types of fusion calls, MetaFusion stores them in a separate *cis-SAGe.cluster* file that can be used for downstream analysis if cis-SAGe fusions are of interest (see Methods, Figure 1).

Indeed, both *SCNN1A-TNFRSF1A* and *CTSD-IFITM10* are identified by MetaFusion in our BRCA benchmarking data. Additionally, *SLC45A3-ELK4*, a ReadThrough present in the urine of men at risk for prostate cancer (Rickman et al. 2009), is detected in our prostate data. These cis-SAGe fusions are

filtered out of the *final.n2.cluster* results by MetaFusion's filters and stored in the *cis-SAGe.cluster* file, as they cannot be distinguished from cis-SAGe RNAs found in normal biology. We further show that *cis-SAGe.cluster* files are enriched for cis-SAGe fusions by analyzing a prostate cancer dataset containing 44 cis-SAGe fusions verified by Sanger sequencing (Qin et al. 2015; Kumar et al. 2016) (Supplementary Figure 1). These results also exemplify that MetaFusion can be easily adapted to specific research questions, such as those pertaining to fusion types that may be of interest in certain tissues or diseases.

3.4 MetaFusion uses FusionAnnotator to identify cancer related fusions in databases

Once a final output of candidate fusions has been obtained, it can be difficult to determine which fusions should be prioritized for further investigation. Cross-referencing databases of known oncogenic fusions can identify calls that have been previously validated in other forms of cancer, and may thus be of more interest for downstream analysis. FusionAnnotator, which is an optional feature of MetaFusion, leverages such databases and can be used on *.cluster* output files to annotate cancer-related fusions. This is done based on gene name.

For example, for the BRCA dataset MetaFusion provides 76 total calls, 58 (76%) of which are in cancer fusion databases, as indicated by FusionAnnotator. 41/42 (98%) of BRCA true positive calls identified with MetaFusion are among these 58. The remaining 17 fusions found in databases may be thus also true positives that were not validated when this truth set was established, making them strong candidates for further experimental analysis.

It should be noted that some cancer fusion databases do contain certain fusions found in normal tissues (Singh et al. 2020). Therefore, although enrichment for hits using FusionAnnotator is useful in identifying and prioritizing fusions that may be expressed in cancer samples, some of these fusions may not be cancer-specific.

3.5 Ranking MetaFusion calls by number of callers

Using multiple tools is the best practice for the field of fusion calling. This was demonstrated by a study in which 23 callers were used to examine the same BRCA dataset that we used, with results showing that implementing three or more callers improved fusion detection (Haas et al. 2019). Specifically, increasing fusion caller number led to enrichment of true fusions that have been experimentally validated.

Likewise, assigning a rank to MetaFusion calls based solely on the number of contributing callers highly correlates with true fusion calls in benchmarking datasets (Figure 6; Supplementary Table 6). Indeed, using *callerfilter7* on MetaFusion output results in 100% precision for all benchmarking datasets (Figure 6). This ranking system is particularly meaningful for the real BRCA dataset, where 14/19 calls made by five to six callers, and all 21 calls made by seven callers are experimentally validated true positives.

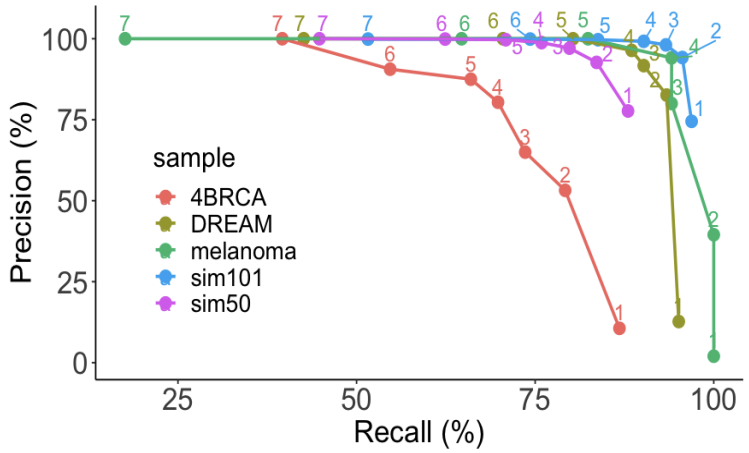

Figure 6: Precision-recall curves for fusions identified by one to seven fusion callers. Precision and recall was evaluated for the five benchmarking datasets (DREAM, sim50, sim101, BRCA and melanoma), for fusions identified by an increasingly stringent number of callers. As the caller requirement increased, recall decreased but precision rose. In all five datasets, all fusions identified by 7 callers were true positives.

Although recall decreases as the number of required callers increases, precision can improve substantially. Our results thus suggest that fusions detected by multiple callers are indeed more likely to be true fusions expressed in a given sample. Fusion calls in the MetaFusion *final.n2.cluster* output file are therefore sorted by the number of callers that identify them, in descending order. Fusions called by more tools are placed at the top of the list, and can be prioritized for further validation using methods such as PCR and Sanger sequencing.

4. Discussion

Here we introduce MetaFusion, a tool for consolidating and prioritizing fusion calls from multiple callers. The MetaFusion pipeline leverages the recall of chosen callers, standardizes format and gene naming among calls, merges fusion calls, and implements a series of stringent filters to provide a final output of fusion candidates. It consistently outperforms individual callers, overcoming limitations of

current fusion calling approaches including high false positive rates, poor recall, lack of a common output format such as VCF for genetic variants, and inconsistent fusion definitions and gene naming conventions. MetaFusion is also equipped with components for further benchmarking and database cross-referencing, making it a flexible tool for fusion calling in cancer and genetic research.

A hallmark of MetaFusion is the seamless integration of multiple fusion callers via the standardization of caller outputs into CFF. While various fusion calling tools are readily available, they can vary significantly in the formatting of their outputs, making consolidation and direct comparison of their results difficult. We thus created the CFF as a means to unify the file formats of each tool used for fusion analysis. This allows for any combination of fusion callers to be used with MetaFusion, as integration of new tools is as simple as converting their output to CFF. Users may choose any combination of callers available to them for input into the pipeline, instead of being limited to a specific set of callers. For these reasons, we hope that future fusion calling tools will include CFF as an output format for their pipelines.

A necessary feature of joint calling is identification of a given fusion by an N number of callers (callerfilterN), and true positives may be missed if merging is not robust enough. MetaFusion relies on a two-component merging process, in which both gene name and breakpoint matching are considered. This is in contrast to other joint calling approaches that rely on one component, such as gene name matching (Beccuti et al. 2014). To rectify naming inconsistencies among callers, MetaFusion renames calls to current NCBI symbols early in the pipeline, improving the downstream merge. Our graph clustering method allows for multiple points of contact between groups of similar calls, reducing the chance that a matching call will be orphaned. Calls can be merged together even if they do not intersect directly (Figure 3). This eliminates the need for manual result curation to check for orphaned calls that may have escaped merge, increasing merging reliability.

Another benefit of MetaFusion is its flexibility and versatility. While individual fusion callers are limited to their own definitions of fusions, MetaFusion combines the ways fusions are identified by all callers included by the user. This makes MetaFusion incredibly customizable, as users can select callers based on their specific needs pertaining to the data they have, the tissues and diseases they are working with, tool preference and more. Furthermore, MetaFusion will only improve as callers with more precise fusion detection capabilities are developed and included in the pipeline. The included Benchmarking Toolkit allows for easy evaluation of new caller combinations using the truth sets provided along with the MetaFusion software. This is a unique feature, as most fusion calling tools do not provide built-in benchmarking functionality. Once caller outputs have been merged, a series of

filters is used to refine the results. Enrichment for cis-SAGe RNAs – which many callers discard – or for fusions found in cancer databases further allows for tailoring of the MetaFusion pipeline to specific research questions.

While many studies rely on a single fusion calling tool for analysis, we show that a joint fusion calling approach can yield higher confidence results. This is key particularly in the context of cancer, where insight into genetic drivers of disease is critical for diagnosis and can shape treatment and prognosis. Oncogenic fusion research is an ever-growing field, and MetaFusion provides several functionalities to aid in cancer-related investigations, including cancer database enrichment and reporting of cis-SAGe. Ultimately, MetaFusion is a novel tool that provides a robust yet versatile approach to fusion calling.

Software availability and implementation

MetaFusion is a free software tool implemented in Python, with bash scripts used as wrappers. The MetaFusion source code is available on GitHub at [https://github.com/ccmbioinfo/MetaFusion]. For convenience and ease of installation, a platform-independent Docker image containing installed dependencies is available at [https://hub.docker.com/r/mapostolides/metafusion]. Instructions for downloading the Docker container, running MetaFusion software, discerning output files and fastq file data access can be found at https://github.com/ccmbioinfo/MetaFusion/wiki. The individual fusion callers used by MetaFusion are available at their respective software repositories (see References).

Supplementary Material

Supplementary material provided online.

Acknowledgements

The authors would like to thank Brian J Haas for his assistance and valuable feedback on this study, and Man Yu for help with data preparation.

Funding

This work was supported by the Canadian Center for Computational Genomics (C3G), part of the Genome Technology Platform (GTP), funded by Genome Canada through Genome Quebec and Ontario Genomics.

Conflict of Interest: none declared.

References

Abate F, Zairis S, Ficarra E, Acquaviva A, Wiggins CH, Frattini V, Lasorella A, Iavarone A, Inghirami G, Rabadan R. 2014. Pegasus: a comprehensive annotation and prediction tool for detection of driver gene fusions in cancer. BMC Syst Biol 8: 97.

Aghaeepour N, Finak G, Hoos H, Mosmann TR, Brinkman R, Gottardo R, Scheuermann RH. 2013. Critical assessment of automated flow cytometry data analysis techniques. Nature Methods 10: 228–238.

Babiceanu M, Qin F, Xie Z, Jia Y, Lopez K, Janus N, Facemire L, Kumar S, Pang Y, Qi Y, et al. 2016. Recurrent chimeric fusion RNAs in non-cancer tissues and cells. Nucleic Acids Research 44: 2859–2872.

Beccuti M, Carrara M, Cordero F, Lazzarato F, Donatelli S, Nadalin F, Policriti A, Calogero RA. 2014. Chimera: a Bioconductor package for secondary analysis of fusion products. Bioinformatics 30: 3556–3557.

Benelli M, Pescucci C, Marseglia G, Severgnini M, Torricelli F, Magi A. 2012. Discovering chimeric transcripts in paired-end RNA-seq data by using EricScript. Bioinformatics 28: 3232–3239.

Berger MF, Levin JZ, Vijayendran K, Sivachenko A, Adiconis X, Maguire J, Johnson LA, Robinson J, Verhaak RG, Sougnez C, et al. 2010. Integrative analysis of the melanoma transcriptome. Genome Res 20: 413–427.

Carey V, Long L, Gentleman R. 2020. RBGL: An interface to the BOOST graph library. https://bioconductor.org/packages/RBGL/ (Accessed July 28, 2020).

Davidson NM, Majewski IJ, Oshlack A. 2015. JAFFA: High sensitivity transcriptome-focused fusion gene detection. Genome Medicine 7. http://genomemedicine.com/content/7/1/43 (Accessed July 28, 2020).

Druker BJ. 2008. Translation of the Philadelphia chromosome into therapy for CML. Blood 112: 4808–4817.

Edgren H, Murumagi A, Kangaspeska S, Nicorici D, Hongisto V, Kleivi K, Rye IH, Nyberg S, Wolf M, Borresen-Dale A-L, et al. 2011. Identification of fusion genes in breast cancer by paired-end RNA-sequencing. Genome Biol 12: R6.

Gao Q, Liang W-W, Foltz SM, Mutharasu G, Jayasinghe RG, Cao S, Liao W-W, Reynolds SM, Wyczalkowski MA, Yao L, et al. 2018. Driver Fusions and Their Implications in the Development and Treatment of Human Cancers. Cell Reports 23: 227-238.e3.

Ge H, Liu K, Juan T, Fang F, Newman M, Hoeck W. 2011. FusionMap: detecting fusion genes from next-generation sequencing data at base-pair resolution. Bioinformatics 27: 1922–1928.

Grant GR, Farkas MH, Pizarro AD, Lahens NF, Schug J, Brunk BP, Stoeckert CJ, Hogenesch JB, Pierce EA. 2011. Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). Bioinformatics 27: 2518–2528.

Haas BJ, Dobin A, Li B, Stransky N, Pochet N, Regev A. 2019. Accuracy assessment of fusion transcript detection via read-mapping and de novo fusion transcript assembly-based methods. Genome Biology 20: 213.

Haas BJ, Dobin A, Stransky N, Li B, Yang X, Tickle T, Bankapur A, Ganote C, Doak TG, Pochet N, et al. 2017. STAR-Fusion: Fast and Accurate Fusion Transcript Detection from RNA-Seq. bioRxiv 120295.

Jia W, Qiu K, He M, Song P, Zhou Q, Zhou F, Yu Y, Zhu D, Nickerson ML, Wan S, et al. 2013. SOAPfuse: an algorithm for identifying fusion transcripts from paired-end RNA-Seq data. Genome Biology 14: R12.

Kumar S, Vo AD, Qin F, Li H. 2016. Comparative assessment of methods for the fusion transcripts detection from RNA-Seq data. Sci Rep 6. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4748267/ (Accessed July 6, 2020).

Lichtenberg J, Elnitski L, Bodine DM. 2017. SigSeeker: a peak-calling ensemble approach for constructing epigenetic signatures. Bioinformatics 33: 2615–2621.

Liu S, Tsai W-H, Ding Y, Chen R, Fang Z, Huo Z, Kim S, Ma T, Chang T-Y, Priedigkeit NM, et al. 2016. Comprehensive evaluation of fusion transcript detection algorithms and a meta-caller to combine top performing methods in paired-end RNA-seq data. Nucleic Acids Research 44: e47–e47.

McPherson A, Hormozdiari F, Zayed A, Giuliany R, Ha G, Sun MGF, Griffith M, Heravi Moussavi A, Senz J, Melnyk N, et al. 2011. deFuse: An Algorithm for Gene Fusion Discovery in Tumor RNA-Seq Data ed. S. Markel. PLoS Computational Biology 7: e1001138.

Mertens F, Johansson B, Fioretos T, Mitelman F. 2015. The emerging complexity of gene fusions in cancer. Nature Reviews Cancer 15: 371–381.

Mitelman F, Johansson B, Mertens F. 2007. The impact of translocations and gene fusions on cancer causation. Nature Reviews Cancer 7: 233–245.

Panigrahi P, Jere A, Anamika K. 2018. FusionHub: A unified web platform for annotation and visualization of gene fusion events in human cancer. PLOS ONE 13: e0196588.

Qin F, Song Z, Babiceanu M, Song Y, Facemire L, Singh R, Adli M, Li H. 2015. Discovery of CTCF-Sensitive Cis-Spliced Fusion RNAs between Adjacent Genes in Human Prostate Cells ed. J. Sage. PLOS Genetics 11: e1005001.

Quinlan AR, Hall IM. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics 26: 841–842.

STAR-SEQR. 2020. STAR-SEQR code and documentation. GitHub. https://github.com/ExpressionAnalysis/STAR-SEQR (Accessed August 13, 2020).

Tang Y, Qin F, Liu A, Li H. 2017. Recurrent fusion RNA DUS4L-BCAP29 in non-cancer human tissues and cells. Oncotarget 8. http://www.oncotarget.com/fulltext/16329 (Accessed May 2, 2019).

Uhrig S. 2020. Arriba code and documentation. GitHub. https://github.com/suhrig/arriba (Accessed August 13, 2020).

Xiao X, Garbutt CC, Hornicek F, Guo Z, Duan Z. 2018. Advances in chromosomal translocations and fusion genes in sarcomas and potential therapeutic applications. Cancer Treatment Reviews 63: 61–70.

Yang W, Deng L. 2020. PreDBA: A heterogeneous ensemble approach for predicting protein-DNA binding affinity. Scientific Reports 10: 1278.

Yoshihara K, Wang Q, Torres-Garcia W, Zheng S, Vegesna R, Kim H, Verhaak RGW. 2015. The landscape and therapeutic relevance of cancer-associated transcript fusions. Oncogene 34: 4845–4854.

Zhang J, White NM, Schmidt HK, Fulton RS, Tomlinson C, Warren WC, Wilson RK, Maher CA. 2016. INTEGRATE: gene fusion discovery using whole genome and transcriptome data. Genome Research 26: 108–118.