09/09/2024 18:03

FusionFlow: an integrated system workflow for gene fusion detection in genomic samples

Federica Citarrella[1], Gianpaolo Bontempo[2,3][0000−0002−6908−1131], Marta Lovino[3][0000−0001−7124−8319], and Elisa Ficarra[3][0000−0002−8061−2124]

[1] Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, Italy
federica.citarrella@studenti.polito.it
[2] University of Pisa, via Lungarno Pacinotti 43, Pisa, Italy
gianpaolo.bontempo@phd.unipi.it
[3] Enzo Ferrari Engeneering Department, University of Modena and Reggio Emilia, Via Vivarelli 10/1, Modena, Italy
{marta.lovino, elisa.ficarra}@unimore.it

Abstract. Gene fusion is a genomic alteration where two genes after a break event are juxtaposed to form a new hybrid gene, leading to possible cancer development and progression. However, identifying gene fusions is not a trivial process as it requires the management and processing countless amounts of data. Genomic data (particularly DNA and RNA) can reach up to 300 GB per sample. Furthermore, specific software and hardware architectures are required to correctly process this type of data. Although many tools are available for detecting gene fusions, to date, systematic workflows that are free and easily usable even by non-specialists are hardly available.
This paper presents an integrated system for identifying gene fusions in RNA and DNA genomic samples, focusing on hardware and software architectural aspects. The proposed workflow is easy-to-use, scalable, and highly reproducible. It includes five gene fusion detection tools, three mainly intended for RNA samples (EricScript, Arriba, FusionCatcher) and two for DNA samples (INTEGRATE and GeneFuse). The workflow runs on servers exploiting Nextflow (a DSL for data-driven computational pipelines), Docker containers, and Conda virtual environments.

Keywords: Gene Fusions · Gene Fusion Detection · genomic samples.

1 Introduction

Gene fusion is a phenomenon that occurs when two or more genes become juxtaposed, forming a single hybrid gene or transcript. Gene fusions remarkably contribute to the evolutionary process by providing a continuous source of new genes. However, at the same time, they often lead to genomic disorders or cancer. Numerous gene fusions have been recognized as essential drivers for various cancer types. Thus, the discovery of novel gene fusions can better comprehend tumour development and progression [29]. For these reasons, gene fusion identification employing gene fusion detection tools has become crucial in bioinformatics

research [30]. Recent advances in deep learning and convolutional networks[3, 6, 4, 23, 22, 9, 8, 10, 21] have also progressively spread to tools for gene fusion detection [19, 17].

Although many gene fusions detection tools have been developed over the past years, it is still challenging to use them. In addition, the RNA-seq artefacts, introduced by library preparation and sequence alignment, make gene fusions predictions hardly reliable [15].
The typical practice is executing multiple tools and using the union or intersection of their results. Unfortunately, this approach is computationally demanding. There are several limitations in traditional tools usage:

- each tool has specific installation requirements and version dependencies that must be precisely adhered to;
- downloading files and databases and executing tools is time-consuming;
- distinct tools can require different input data formats;
- multiple complementary fusion detection tools are needed to improve sensitivity.

During the last years, bioinformatics workflows (which consist of a wide array of algorithms executed in a predefined sequence) were developed to deal with multiple bioinformatic issues (e.g., RNA data processing and CNA detection) [24]. However, only a limited number of gene fusion detection workflows is available, and no one of them can simultaneously handle both RNA and DNA sequencing data [28].
This paper presents FusionFlow, an easily reproducible and scalable bioinformatics workflow for detecting gene fusions from RNA and DNA data. It processes numerous sequence data and their associated metadata through multiple transformations using a series of software components, databases, and operation environments (hardware and operating system). It includes five gene fusion detection tools executed through multiple processes. The processes are built using Nextflow Groovy/JVM-based framework exploiting Docker and Conda technologies. Indeed, Nextflow allows running tools downloads, installation, and execution concurrently in the interest of time constraints. At the same time, Docker and Conda engines are used to create virtual environments precisely configured for each tool. Finally, the pipeline inputs standard data formats and eventually converts them directly inside specific converter processes.

2 The workflow

FusionFlow includes five fusion detection tools: EricScript [5], Arriba [26], FusionCatcher [20], GeneFuse [7] and Integrate [32]. Three of them, EricScript, Arriba, and FusionCatcher, accept as input just RNA-seq data. Concerning DNA tools, GeneFuse takes just DNA data, while Integrate has two input options: 1) just RNA data or 2) both RNA and DNA data. All gene fusion detection tools are made up of three steps: 1) preliminary alignment of the reads (a row in the

genomic input files) to the transcriptome to build specific gene fusion references; 2) alignment of previously unmapped reads to gene fusion references to support the gene fusion detection; 3) cleaning filters to discard false positives. The main differences between the tools consist of the alignment type (e.g., BLAST vs BWA) and the properties of the cleaning filters. Although a proper gold standard procedure for gene fusion detection has not been established, the most widely used approach involves applying multiple gene fusion detection tools, unifying the results obtained. Ericscript, FusionCatcher, and Arriba have been selected for this workflow due to their spread and unique characteristics. Ericscript and FusionCatcher have been selected due to the differences in the cleaning filters. The former exploits, among the others, heuristic filters to remove analysis artefacts, while the latter removes false positives using known and novel criteria, which make biological sense. In the end, Arriba has been chosen since it can find aberrations that the competitors hardly find (e.g. intragenic and intergene duplications/inversions/translocations). Since the DNA sequencing method has only recently spread on a large scale [16], the panorama in DNA gene fusion detection tools includes a few software available. GeneFuse and INTEGRATE deserve to be mentioned for their user experience. GeneFuse can detect gene fusions from DNA samples alone, while INTEGRATE requires both RNA and DNA data from the same sample to provide the gene fusion list. At the same time, it can reconstruct gene fusion junctions and genomic breakpoints by split-read mapping in a complete way.

In order to make the pipeline usage as simple as possible, the only mandatory inputs are the RNA or DNA files to be analyzed. In this case, the workflow looks for tools' required files in default paths. The gene fusion detection tools start processing data if the files are present. Otherwise, the pipeline downloads and installs all the necessary tools and files before the tools' execution. FusionFlow receives input RNA only, DNA only, or both RNA and DNA data.

The FusionFlow pipeline produces several files divided into two categories: tools' required files and gene fusions' output files. The first category includes all the files needed to execute the tools. These files can be directly provided to the workflow, skipping their downloads processes, or can be downloaded while running the workflow for the first time. Then, the files will be saved in a specific path to be available to the pipeline for the subsequent runs.
The second category of output includes the files produced as output from the gene fusion tools. Each tool gives as output one or more files in specific formats. The most diffused formats are Tab Separated Value (TSV), Variant Call Format (VCF), and standard text format [2].

In the following, the general workflow architecture is described.

2.1 Architecture

The general workflow structure is based on Nextflow, a dataflow programming model that simplifies writing complex distributed pipelines.

Nextflow Groovy/JVM-based framework is selected among a series of workflow management systems (e.g., Galaxy[11], Toil[27], Snakemake[14], Bpipe) due to its peculiar features. In particular it allows:

- the existence of several processes written in different languages. Nextflow recognizes the script's language automatically, and it generates a launch file per process dynamically;
- to process data as stream step by step. Indeed, each process can communicate through the input/output channel definition. These channels can also be used for synchronization mechanisms in order to make the pipeline sequential;
- integration with sharing platforms such as GitHub. Nextflow can notice if the repository is not installed and, in that case, it downloads all the requirements, environments included;
- integration with the most famous containers as Docker and Singularity. This feature is crucial for gene fusion tools since they often require conflicting packages. The current pipeline has considered each process in a separate environment;
- integration with several schedulers as SLURM. Due to the substantial memory boundaries requested by the gene fusion tools, the pipeline can be executed basically on large systems servers. Rarely are they used without a scheduler.

The workflow is composed of fifteen processes. These processes can be divided into three main categories:

- **downloaders**: they are responsible for the tools installation and download input files. The downloaders processes are: *referenceGenome_downloader*, *arriba_downloader*, *ericscript_downloader*, *fusioncatcher_downloader*, *integrate_downloader* and *genefuse_downloader*;
- **converters**: they are responsible for the file preparation and format conversion if needed. The converters processes are: *integrate_converter* and *genefuse_converter*;
- **runners**: they allow the code and tools execution. The runner processes are: *arriba*, *ericscript*, *fusioncatcher*, *integrate*, *genefuse*, *referenceGenom_index*, *integrate_builder*.

The fifteen processes are structured into six main parallel lines shown in green in Figure 1.

Executing the script with Nextflow, the algorithm will look for the required files in the paths specified in *nextflow.config* configuration file or the paths specified in the command line. The associated downloader is skipped if the files exist, and the following processes can start processing.
Nextflow processes usually are executed concurrently. Nextflow queue channels are used to execute downloaders, converters, and runners sequentially and provide inter-communication between processes. A queue channel creates an asynchronous unidirectional FIFO queue and allows to connect processes or operators. Using a combination of queue channels permits the creation of predefined

sequences of processes. The processes expect to receive input data from the channels specified in the input block. When the inputs are emitted, the processes run.

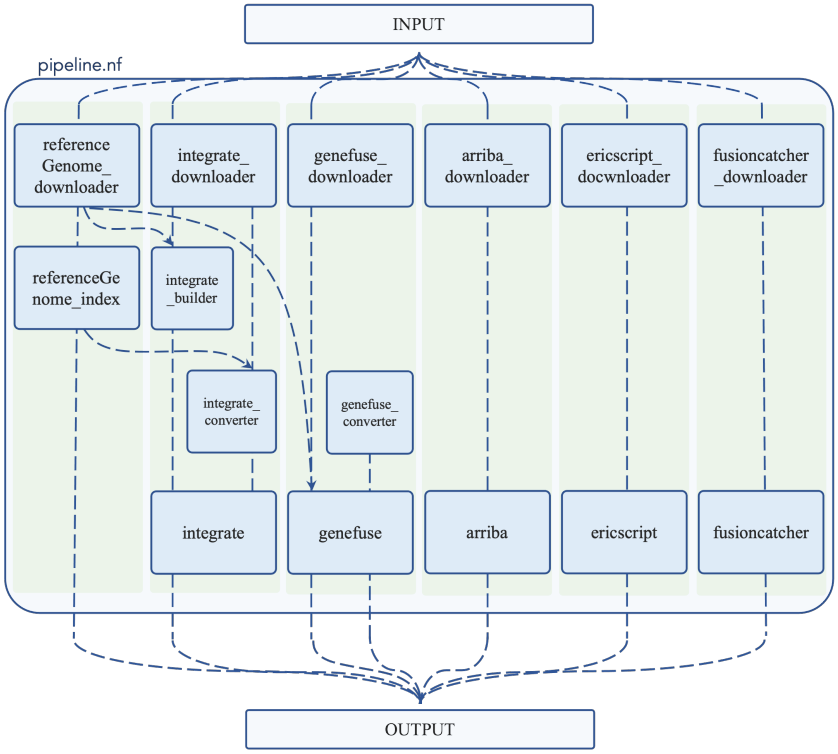

Fig. 1. Pipeline architecture parallelization: each tool is composed of multiple subunits (shown in blue) executed concurrently through six main parallel lines (shown in green) to optimize the workflow performances.

The five fusion detection tools included in FusionFlow are managed through Nextflow queue channels that provide inter-communication between the workflow processes. All processes have the same structure since they are triggered by input and, after the script block execution, provide output to trigger the subsequent processes. As illustrated in fig 2, for each channel, the first step consists of describing the channel configuration (e.g., DNA files, RNA files, the tool installation path, and further databases necessary for gene fusion tools). If the user does not define tool databases, a separate channel is used to download it. A data channel passes the database to the next process triggered at the moment. Then, the tool/database is installed if not present yet, and the data is converted in the correct format if the user requests it. Finally, the data is passed to the

gene fusion tool for the tool execution.

In the end, the workflow provides as output the list of candidate gene fusions for each tool to be investigated by the user.

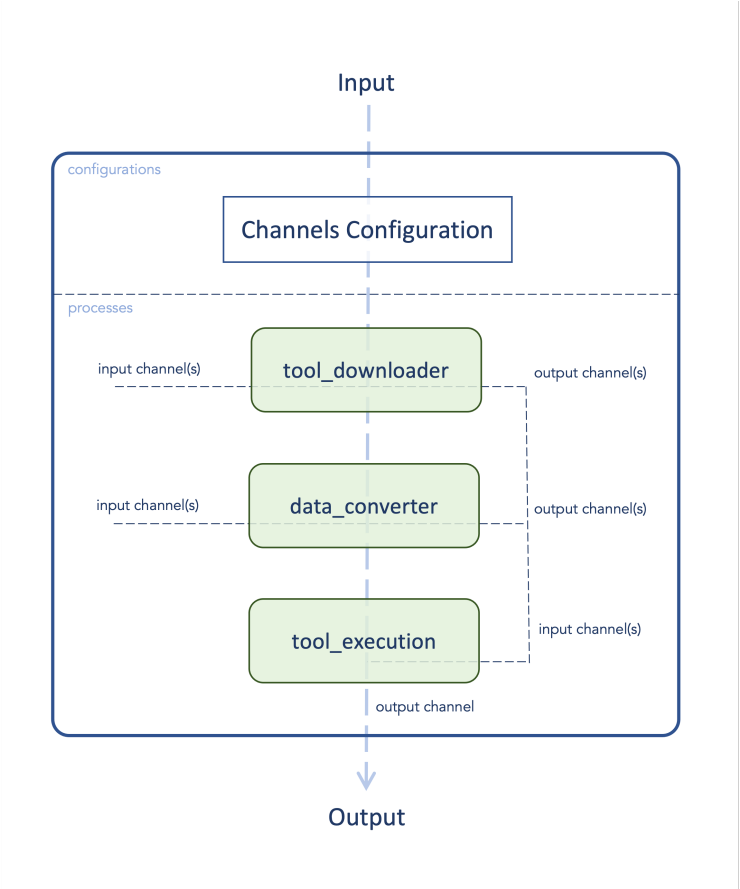

Fig. 2. General flow for each tool. Given the data as input, 1) the required configuration is set; 2) the tool/database is installed if not present; 3) the data is converted if with the wrong format; 4) the data is passed to the gene fusion tool.

3 Workflow test and discussion

The files used to test the pipeline are the same proposed in the FusionCatcher tool and publicly available both at https://github.com/ndaniel/fusioncatcher/tree/master/test and in the FusionFlow GitHub repository.

They are fastq compressed paired-end files (a standard file data format used to store genetic information) where the reads were manually selected to cover 17 already known fusion genes: FGFR3 - TACC3, FIP1L1 - PDGFRA, GOPC - ROS1, IGH - CRLF2, HOOK3 - RET, AKAP9 - BRAF, EWSR1 - ATF1, TMPRSS2 - ETV1, EWSR1 - FLI1, ETV6 - NTRK3, ETV6 - NTRK3, ETV6 - NTRK3, BRD4 - NUTM1, CD74 - ROS1, CIC - DUX4, DUX4 - IGH, DUX4 - IGH, EML4 - ALK, MALT1 - IGH, NPM1 - ALK.

Initially, each tool was tested separately on a linux operating system and inside a docker environment checking the setup (e.g. paths, files, profiles, libraries). Then, after making sure that all the tools worked, the entire Nextflow workflow was tested inside a single docker environment. In the scenario without docker, Conda virtual environments were manually created. Otherwise, with docker the setup is prepared automatically through the use of the dockerfile. The test files and the local profile were specified in the command line to execute these tests.

Each gene fusion detection tool gives output one or more files in specific formats. Generally, a summary file is produced in output to allow a quick predictions overview.
The outputs obtained from the tools are concordant with the gene fusions previously specified. All the tools in the pipeline recognize at least ten predictions out of seventeen fusions, except for GeneFuse, which recognizes just three of them. Although GeneFuse performances should be investigated on additional data, the poor result could be explained by the specific DNA filters implemented in GeneFuse.
In order to select the final gene fusions prediction drivers, different approaches can be used. The typical practice is to use the union or intersection of tools predictions. The union of the results gives numerous sets of predictions. This approach increases the probability of including the real drivers of cancer processes. However, it enhances the possibility of incorporating false positives or passenger mutations. Using the intersection approach, conversely, decreases the number of predictions radically. This approach allows discarding false positives and passenger mutations. However, this selection could also cause the discarding of the cancer drivers.
In this test case, the union of the results contains nineteen gene fusions predictions, while the intersection includes just two of them (ETV6-NTRK3 and GOPC-ROS1).

4 Conclusions

FusionFlow is an easy-to-use, flexible, highly reproducible, and integrated workflow. The workflow includes five gene fusion discovery tools that input both RNA and DNA data. Docker and Conda technologies allow performing tools installations, avoiding version conflicts. In addition, the Nextflow framework allows the execution of the five tools in parallel, optimizing time and resources usage

and managing the tool's installations and the file allocation. The workflow was tested using publicly available test files. The tests were performed using a local profile in two conditions: on a private server and the private server inside a docker container. In both cases, the outputs were satisfactory. Thus, the FusionFlow pipeline is available for further validation over additional DNA and RNA genomic data.

This work represents a foundation on which improvements and future works can be built. Indeed, one of the main problems related to gene fusion detection is determining which gene fusions are drivers of cancer processes and not just passenger mutations. The fusion detection tools already provide a first step for solving this problem. Indeed, fusion detection tools filter the candidate gene fusions based on the sample's reads, trying to decrease as much as possible the number of false positives. However, generally, this step is insufficient to determine the cancer drivers, and an additional step can be required. It consists of post-processing tools (called prioritization tools) that can predict a gene fusion's oncogenic potential. There is a high number of prioritization tools such as Oncofuse, Pegasus, DEEPrior, and ChimerDriver [25, 1, 19, 17, 18]. These tools are based on machine learning (ML) algorithms trained with the protein domains of the fusion proteins and allow the selection of the most probable cancer drivers. The post-processing step could also be completed by adding a different algorithm. This algorithm performs comparisons between the outputs of the tool and selects the more probable driver of cancer processes by analyzing the union and the intersection and taking into account the different characteristics of the gene fusion detection tools.
Another crucial question is related to visualization tools. Humans can efficiently distinguish true positives from false positives if the evidence is provided in an easily interpretable form. These tools also better interpret the potential consequence of gene fusion events. Several visualization tools were released in the last years, such as INTEGRATE-vis [31], FGviewer [13], and FuSpot [12].

5 Funding

This study was funded by the European Union's Horizon 2020 research and innovation programme DECIDER under Grant Agreement 965193.

References

1. Abate, F., Zairis, S., Ficarra, E., Acquaviva, A., Wiggins, C.H., Frattini, V., Lasorella, A., Iavarone, A., Inghirami, G., Rabadan, R.: Pegasus: A comprehensive annotation and prediction tool for detection of driver gene fusions in cancer. BMC Systems Biology **8** (9 2014). https://doi.org/10.1186/s12918-014-0097-z
2. Ahmed, S., Ali, M.U., Ferzund, J., Sarwar, M.A., Rehman, A., Mehmood, A.: Modern data formats for big bioinformatics data analytics (2017), www.ijacsa.thesai.org
3. Allegretti, S., Bolelli, F., Cancilla, M., Pollastri, F., Canalini, L., Grana, C.: How does connected components labeling with decision trees perform on GPUs? In:

International Conference on Computer Analysis of Images and Patterns. pp. 39–51. Springer (2019)
4. Allegretti, S., Bolelli, F., Pollastri, F., Longhitano, S., Pellacani, G., Grana, C.: Supporting Skin Lesion Diagnosis With Content-Based Image Retrieval. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 8053–8060. IEEE (2021)
5. Benelli, M., Pescucci, C., Marseglia, G., Severgnini, M., Torricelli, F., Magi, A.: Discovering chimeric transcripts in paired-end rna-seq data by using ericscript. Bioinformatics **28**, 3232–3239 (12 2012). https://doi.org/10.1093/bioinformatics/bts617
6. Bolelli, F., Baraldi, L., Pollastri, F., Grana, C.: A Hierarchical Quasi-Recurrent Approach to Video Captioning. In: 2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS). pp. 162–167. IEEE (2018)
7. Chen, S., Liu, M., Huang, T., Liao, W., Xu, M., Gu, J.: Genefuse: Detection and visualization of target gene fusions from dna sequencing data. International Journal of Biological Sciences **14**, 843–848 (5 2018). https://doi.org/10.7150/ijbs.24626
8. Cirrincione, G., Randazzo, V., Kumar, R.R., Cirrincione, M., Pasero, E.: Growing curvilinear component analysis (gcca) for stator fault detection in induction machines. In: Neural Approaches to Dynamics of Signal Exchanges, pp. 235–244. Springer (2020)
9. Cirrincione, G., Randazzo, V., Pasero, E.: Growing curvilinear component analysis (gcca) for dimensionality reduction of nonstationary data. In: Multidisciplinary Approaches to Neural Computing, pp. 151–160. Springer (2018)
10. Cirrincione, G., Randazzo, V., Pasero, E.: A neural based comparative analysis for feature extraction from ecg signals. In: Neural approaches to dynamics of signal exchanges, pp. 247–256. Springer (2020)
11. Goecks, J., Nekrutenko, A., Taylor, J.: Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome biology **11**(8), 1–13 (2010)
12. Killian, J.A., Topiwala, T.M., Pelletier, A.R., Frankhouser, D.E., Yan, P.S., Bundschuh, R.: Fuspot: A web-based tool for visual evaluation of fusion candidates. BMC Genomics **19** (2 2018). https://doi.org/10.1186/s12864-018-4486-3
13. Kim, P., Yiya, K., Zhou, X.: Fgviewer: An online visualization tool for functional features of human fusion genes. Nucleic Acids Research **48**, W313–W320 (2021). https://doi.org/10.1093/NAR/GKAA364
14. Köster, J., Rahmann, S.: Snakemake—a scalable bioinformatics workflow engine. Bioinformatics **28**(19), 2520–2522 (08 2012). https://doi.org/10.1093/bioinformatics/bts480, https://doi.org/10.1093/bioinformatics/bts480
15. Latysheva, N.S., Babu, M.M.: Discovering and understanding oncogenic gene fusions through data intensive computational approaches. Nucleic Acids Research **44**, 4487–4503 (6 2016). https://doi.org/10.1093/nar/gkw282
16. Lovino, M., Bontempo, G., Cirrincione, G., Ficarra, E.: Multi-omics classification on kidney samples exploiting uncertainty-aware models. In: International Conference on Intelligent Computing. pp. 32–42. Springer (2020)
17. Lovino, M., Ciaburri, M.S., Urgese, G., Di Cataldo, S., Ficarra, E.: Deeprior: a deep learning tool for the prioritization of gene fusions. Bioinformatics **36**(10), 3248–3250 (2020)
18. Lovino, M., Montemurro, M., Barrese, V.S., Ficarra, E.: Identifying the oncogenic potential of gene fusions exploiting mirnas. Journal of Biomedical Informatics **129**, 104057 (2022)

19. Lovino, M., Urgese, G., Macii, E., Di Cataldo, S., Ficarra, E.: A deep learning approach to the screening of oncogenic gene fusions in humans. International journal of molecular sciences **20**(7), 1645 (2019)
20. Nicorici, D., Satalan, M., Edgren, H., Kangaspeska, S., Murumagi, A., Kallioniemi, O., Virtanen, S., Kilkku, O.: Fusioncatcher - a tool for finding somatic fusion genes in paired-end rna-sequencing data. bioRxiv p. 011650 (2014). https://doi.org/10.1101/011650
21. Paviglianiti, A., Randazzo, V., Pasero, E., Vallan, A.: Noninvasive arterial blood pressure estimation using abpnet and vital-ecg. In: 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). pp. 1–5. IEEE (2020)
22. Ponzio, F., Deodato, G., Macii, E., Di Cataldo, S., Ficarra, E.: Exploiting "uncertain" deep networks for data cleaning in digital pathology. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). pp. 1139–1143. IEEE (2020)
23. Ponzio, F., Villalobos, A.E.L., Mesin, L., de'Sperati, C., Roatta, S.: A human-computer interface based on the "voluntary" pupil accommodative response. International Journal of Human-Computer Studies **126**, 53–63 (2019)
24. Roy, S., Coldren, C., Karunamurthy, A., Kip, N.S., Klee, E.W., Lincoln, S.E., Leon, A., Pullambhatla, M., Temple-Smolkin, R.L., Voelkerding, K.V., Wang, C., Carter, A.B.: Standards and guidelines for validating next-generation sequencing bioinformatics pipelines: A joint recommendation of the association for molecular pathology and the college of american pathologists (1 2018). https://doi.org/10.1016/j.jmoldx.2017.11.003
25. Shugay, M., Mendíbil, I.O.D., Vizmanos, J.L., Novo, F.J.: Oncofuse: A computational framework for the prediction of the oncogenic potential of gene fusions. Bioinformatics **29**, 2539–2546 (10 2013). https://doi.org/10.1093/bioinformatics/btt445
26. Uhrig, S., Ellermann, J., Walther, T., Burkhardt, P., Hutter, B., Toprak, U.H., Neumann, O., Stenzinger, A., Scholl, C., Fröhling, S., Brors, B.: Accurate and efficient detection of gene fusions from rna sequencing data
27. Vivian, J., Rao, A.A., Nothaft, F.A., Ketchum, C., Armstrong, J., Novak, A., Pfeil, J., Narkizian, J., Deran, A.D., Musselman-Brown, A., et al.: Toil enables reproducible, open source, big biomedical data analyses. Nature biotechnology **35**(4), 314–316 (2017)
28. Wang, Q., Xia, J., Jia, P., Pao, W., Zhao, Z.: Application of next generation sequencing to human gene fusion detection: Computational tools, features and perspectives. Briefings in Bioinformatics **14**, 506–519 (7 2013). https://doi.org/10.1093/bib/bbs044
29. Wang, Y., Shi, T., Song, X., Liu, B., Wei, J.: Gene fusion neoantigens: Emerging targets for cancer immunotherapy (5 2021). https://doi.org/10.1016/j.canlet.2021.02.023
30. Williford, A., Betrán, E.: Gene fusion (5 2013). https://doi.org/10.1002/9780470015902.a0005099.pub3, https://onlinelibrary.wiley.com/doi/10.1002/9780470015902.a0005099.pub3
31. Zhang, J., Gao, T., Maher, C.A.: Integrate-vis: A tool for comprehensive gene fusion visualization. Scientific Reports **7** (12 2017). https://doi.org/10.1038/s41598-017-18257-2
32. Zhang, J., White, N.M., Schmidt, H.K., Fulton, R.S., Tomlinson, C., Warren, W.C., Wilson, R.K., Maher, C.A.: Integrate: Gene fusion discovery using whole genome and transcriptome data. Genome Research **26**, 108–118 (1 2016). https://doi.org/10.1101/gr.186114.114