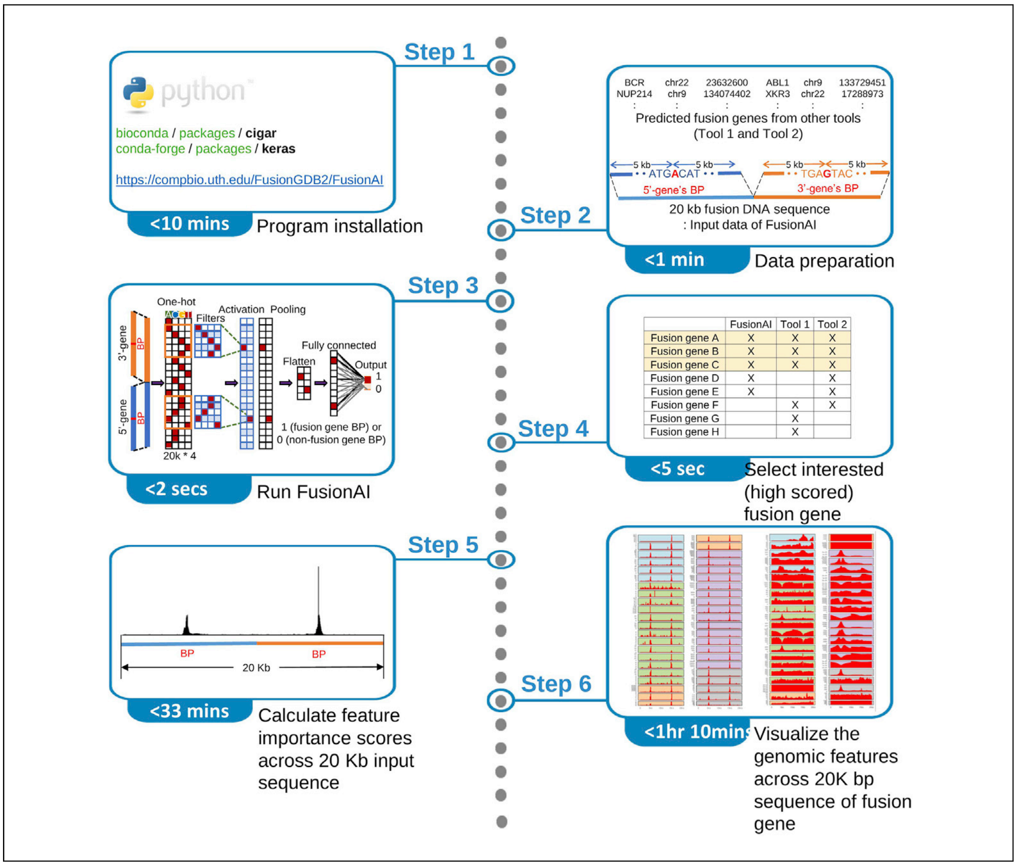
STAR Protocols

CellPress
OPEN ACCESS

Protocol

FusionAI, a DNA-sequence-based deep learning protocol reduces the false positives of human fusion gene prediction


Step 1
python
bioconda / packages / cigar
conda-forge / packages / keras
https://compbio.uth.edu/FusionGDB2/FusionAI
<10 mins
Program installation
Step 2

BCR	chr22	23632600	ABL1	chr9	133729451
NUP214	chr9	134074402	XKR3	chr22	17288973
:	:	:	:	:	:

	FusionAI	Tool 1	Tool 2
Fusion gene A	X	X	X
Fusion gene B	X	X	X
Fusion gene C	X	X	X
Fusion gene D	X		X
Fusion gene E	X		X
Fusion gene F		X	X
Fusion gene G		X	
Fusion gene H		X	

<5 sec
Select interested (high scored) fusion gene
Step 5
BP
BP
20 Kb
<33 mins
Calculate feature importance scores across 20 Kb input sequence
Step 6
<1hr 10mins
Visualize the genomic features across 20K bp sequence of fusion gene

Even though there were many tool developments of fusion gene prediction from NGS data, too many false positives are still an issue. Wise use of the genomic features around the fusion gene breakpoints will be helpful to identify reliable fusion genes efficiently. For this aim, we developed FusionAI, a deep learning pipeline predicting human fusion gene breakpoints from DNA sequence. FusionAI is freely available via https://compbio.uth.edu/FusionGDB2/FusionAI.

Pora Kim, Hua Tan, Jiajia Liu, Himansu Kumar, Xiaobo Zhou

pora.kim@uth.tmc.edu

Highlights

FusionAI can predict the fusion breakpoints from the given DNA sequence

FusionAI can reduce the false positives of the predicted fusion genes by other tools

FusionAI can identify the genomic features related to the genomic breakage

FusionAI creates a landscape image of 44 human genomic features around the breakpoints

Kim et al., STAR Protocols 3, 101185
March 18, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.xpro.2022.101185

STAR Protocols

CellPress
OPEN ACCESS

Protocol

FusionAI, a DNA-sequence-based deep learning protocol reduces the false positives of human fusion gene prediction

Pora Kim,[1,5,6,7,*] Hua Tan,[1,5] Jiajia Liu,[1,4] Himansu Kumar,[1] and Xiaobo Zhou[1,2,3]

[1]School of Biomedical Informatics, The University of Texas Health Science Center at Houston, Houston, TX 77030, USA

[2]McGovern Medical School, The University of Texas Health Science Center at Houston, Houston, TX 77030, USA

[3]School of Dentistry, The University of Texas Health Science Center at Houston, Houston, TX 77030, USA

[4]College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China

[5]These authors contributed equally

[6]Technical contact

[7]Lead contact

*Correspondence: pora.kim@uth.tmc.edu

https://doi.org/10.1016/j.xpro.2022.101185

SUMMARY

Even though there were many tool developments of fusion gene prediction from NGS data, too many false positives are still an issue. Wise use of the genomic features around the fusion gene breakpoints will be helpful to identify reliable fusion genes efficiently. For this aim, we developed FusionAI, a deep learning pipeline predicting human fusion gene breakpoints from DNA sequence. FusionAI is freely available via https://compbio.uth.edu/FusionGDB2/FusionAI.

For complete details on the use and execution of this protocol, please refer to Kim et al. (2021b).

BEFORE YOU BEGIN

Since the accelerated accumulation of the next-generation sequencing data, there were many tool developments for the prediction of fusion genes from the RNA-seq data such as STAR-Fusion (Haas et al., 2019), Arriba (Uhrig et al., 2021), SOAPfuse (Jia et al., 2013), deFuse (McPherson et al., 2011), and FusionScan (Kim et al., 2019). The main difference between those tools comes from the ways of dealing with the RNA sequencing reads that were aligned far apart and repeat region mappings. However, too many false positives were the main problems in the prediction of fusion genes and the researchers regarded the fusion genes that were predicted in more than two prediction tools as reliable fusions. This selection approach can be helpful in reducing some false positives, but also not be helpful in terms of that all these tools are relying on the split RNA sequencing reads. Using other types of information like genomic sequence features around the breakpoint area can be a helpful and efficient way for better removal of the false positives. To help identify reliable fusion genes efficiently, we developed FusionAI, a deep learning pipeline predicting human fusion gene breakpoints from DNA sequences. For the given breakpoint of fusion genes, FusionAI provides the possibility of being used as the fusion gene breakpoints and landscapes of human genomic features around the fusion gene breakpoints. FusionAI is freely available via https://compbio.uth.edu/FusionGDB2/FusionAI.

The protocol below describes the specific steps for running FusionAI for the fusion genes predicted in K562 cell using STAR-Fusion (Haas et al., 2019). By combining the output result of FusionAI to these predicted fusion genes, we can have more reliable fusion genes with reduced false positives from the fusion DNA sequence using the genomic features of the fusion gene breakpoints.

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Software prerequisites and data requirements

Our model is installed and run under the Linux system. Before launching our program, preinstalled Python (>= v.3.0), TensorFlow, and Keras modules are required. You should also prepare fusion gene information that was predicted using other existing tools for your cancer sample. The example of prerequisites and input data format can be found on our website: https://compbio.uth.edu/FusionGDB2/FusionAI. All the R packages required to visualize 44 human genome features in a 20 Kb DNA sequence are listed in the key resources table under the "R packages to draw feature landscape image" category. The R package "bedtoolsr" can only be installed using devtools::install_github ("PhanstielLab/bedtoolsr") and other R packages can be installed using install.packages() function.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
newdat_newmod_jj.h5	FusionAI model in this paper.	https://compbio.uth.edu/FusionGDB2/FusionAI/newdat_newmod_jj.h5
gencode_hg19v19_.txt	Gene structure information file with UCSC genome browser known gene format of GENCODE version 19.	https://compbio.uth.edu/FusionGDB2/FusionAI/gencode_hg19v19_.txt
nib_files_hg19.tar.gz	Nib files of all chromosomes of hg19, which were transformed from fasta files provided from the UCSC genome browser.	https://compbio.uth.edu/FusionGDB2/FusionAI/nib_files_hg19.tar.gz
chromosome_size.txt	This paper	https://compbio.uth.edu/FusionGDB2/FusionAI/chromosome_size.txt
features_info.txt	This paper	https://compbio.uth.edu/FusionGDB2/FusionAI/features_info.txt
feature.tar.gz	This paper	https://compbio.uth.edu/FusionGDB2/FusionAI/feature.tar.gz
Software and algorithms		
Python (>=3.0)	Python Software Foundation, 2021: high-level programming language	https://www.python.org/downloads/
nibFrag	Converts portions of a .nib file back to fasta format.	http://hgdownload.soe.ucsc.edu/admin/jksrc.zip
Tensor flow	TensorFlow is an end-to-end open source platform for machine learning.	https://anaconda.org/conda-forge/tensorflow
keras	A deep learning framework developed by François Chollet	https://github.com/keras-team/keras
pandas	A community project for fast and easy data analysis and manipulation	https://pandas.pydata.org/about/
numpy	Community project, 2021: array processing for numbers, strings, records, and objects	https://numpy.org/
argparse	A python module that makes it easy to write user-friendly command-line interfaces	https://docs.python.org/3/library/argparse.html
FusionAI_pred.py	This paper	https://compbio.uth.edu/FusionGDB2/FusionAI/FusionAI_pred.py
FusionAI_FIS.py	This paper	https://compbio.uth.edu/FusionGDB2/FusionAI/FusionAI_FIS.py
pre_processing_for_FusionAI_from_tab_delim.py	This paper	https://compbio.uth.edu/FusionGDB2/FusionAI/pre_processing_for_FusionAI_from_tab_delim.py
bedtools (>=2.26.0)	(Quinlan and Hall, 2010): a powerful toolset for genome arithmetic	https://bedtools.readthedocs.io/en/latest/content/installation.html
R (>=3.5)	(Team, 2019): software environment for statistical computing and graphics	https://www.r-project.org/
devtools (>=1.13.6)	(Wickham et al., 2018): developing R Packages tool	https://cran.r-project.org/web/packages/devtools/index.html
bedtoolsr (2.30.0.1)	(Patwardhan et al., 2019): genomic data analysis and manipulation	http://phanstiel-lab.med.unc.edu/bedtoolsr-install.html
optparse (>=1.6.0)	(Davis, 2018): Command Line Option Parser	https://cran.r-project.org/web/packages/optparse/index.html

(Continued on next page)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
doParallel (1.0.16)	(Corporation and Weston, 2020): parallel backend	https://cran.r-project.org/web/packages/doParallel/index.html
iterators (1.0.13)	(Analytics and Weston, 2020): a package to allow a programmer to traverse through all the elements of a vector, list, or other collection of data	https://cran.r-project.org/web/packages/iterators/index.html
magrittr (2.0.1)	(Bache and Wickham, 2020): A Forward-Pipe Operator for R	https://cran.r-project.org/web/packages/magrittr/index.html
foreach (1.5.1)	(Microsoft and Weston, 2020): an idiom that allows for iterating over elements in a collection, without the use of an explicit loop counter.	https://cran.r-project.org/web/packages/foreach/index.html
ggplot2 (3.3.5)	(Wickham, 2016): Elegant Graphics for Data Analysis	https://cran.r-project.org/web/packages/ggplot2/index.html
gridExtra (2.3)	(Auguie, 2017): a package to arrange multiple grid-based plots on a page	https://cran.r-project.org/web/packages/gridExtra/index.html
scales (1.1.1)	(Wickham and Seidel, 2020): Graphical scales map data to aesthetics, and provide methods for automatically determining breaks and labels for axes and legends.	https://cran.r-project.org/web/packages/scales/index.html
cowplot (1.1.1)	(Wilke, 2020): a set of themes, functions to align plots and arrange them into complex compound figures, and functions that make it easy to annotate plots and or mix plots with images.	https://cran.r-project.org/web/packages/cowplot/index.html
ggpubr (>=0.1.7)	(Kassambara, 2018): 'ggplot2' Based Publication Ready Plots	https://cran.r-project.org/web/packages/ggpubr/index.html

MATERIALS AND EQUIPMENT

The program in this protocol was written in the Ubuntu Linux system using Python language (>=v.3.0). All experiments were carried out and evaluated under the Ubuntu system with the computational resources listed in Table 1.

> ⚠ CRITICAL: The implementation of the model is lightweight. However, the required memory usage in practice depends on the size of your own data.

> *Alternatives:* 1. Our model can work with fewer CPU cores and less RAM memory, although it may take a longer time for a large dataset. During running the example input for FusionAI, it used 17.6% of a CPU and 0.3% of the memory of the server with the computation capacity described in Table 1. 2. If the user does not need to draw the feature images, then no need to install the software and algorithms to draw the feature landscape images listed in the key resources table.

STEP-BY-STEP METHOD DETAILS

Download our package and install the prerequisites

> ⏲ Timing: < 10 min

1. Download the latest version of FusionAI into your preferred directory. The running will be executed inside of this directory (a, b, c, and d are required for running FusionAI. e, f and g are required to draw feature landscape images for the chosen fusion genes):
   a. Download FusionAI_pred.py from https://compbio.uth.edu/FusionGDB2/FusionAI/FusionAI_pred.py
   b. Download FusionAI model (newdat_newmod_jj.h5) from https://compbio.uth.edu/FusionGDB2/FusionAI/newdat_newmod_jj.h5

Table 1. Computation resources used in this study

Operating system	Version
CentOS Linux	7.9.2009
CPU information	Parameter
RAM Memory	93 GB
Thread(s) per core	2
Core(s) per socket	2
Model	85
Model name	Intel(R) Xeon(R) Gold 6254 CPU @ 3.10 GHz
CPU MHz:	2899.816
CPU(s)	36

c. Download preprocessing script (pre_processing_for_FusionAI_from_tab_delim.py) from https://compbio.uth.edu/FusionGDB2/FusionAI/pre_processing_for_FusionAI_from_tab_delim.py
d. Download example fusion gene file (k562_starfusion.txt) https://compbio.uth.edu/FusionGDB2/FusionAI/k562_starfusion.txt
e. Download 44 human genomic feature information files (features.tar.gz, features_info.txt, and chromosome_size.txt) from https://compbio.uth.edu/FusionGDB2/FusionAI/features.tar.gz, https://compbio.uth.edu/FusionGDB2/FusionAI/features_info.txt, and https://compbio.uth.edu/FusionGDB2/FusionAI/chromosome_size.txt
f. Download human gene structure file and nib files (gencode_hg19v19_.txt and nib_files_hg19.tar.gz) from https://compbio.uth.edu/FusionGDB2/FusionAI/gencode_hg19v19_.txt and https://compbio.uth.edu/FusionGDB2/FusionAI/nib_files_hg19.tar.gz
g. Install R packages using the command install.packages(). Input the individual R package name into the parenthesis like install.packages('devtools'). These

Prepare input data of 20 Kb DNA sequence of fusion genes

⏱ Timing: < 1 min

FusionAI takes the input data of fusion gene breakpoint information, which is given by other fusion gene prediction tools or known fusion gene information (k562_starfusion.txt and Table 2). The preprocessing script will make 20 Kb DNA sequences for individual fusion genes, which is the combined sequence of +/-5 Kb flanking sequence from the two breakpoints' genomic position for individual fusion partner genes (Figure 1 and Table 3).

2. Run preprocessing script to make a 20 Kb DNA sequence from the given fusion gene information. The fusion gene information should include the following information in tab-delimited format: Hgene, Hchr, Hbp, Hstrand, Tgene, Tchr, Tbp, Tstrand. The command is shown below. Here the $ INPUT_FILE is the output file after checking the junction position of the fusion breakpoints in step 2.

```
> python pre_processing_for_FusionAI_from_tab_delim.py [INPUT_FILE]

> python pre_processing_for_FusionAI_from_tab_delim.py k562_starfusion.txt
```

⚠ CRITICAL: The timing is based on the number of fusion genes of the input file.

Run FusionAI

⏱ Timing: < 2 s (depending on your data)

Table 2. Fusion gene information example, which were predicted for K562 cell-line from STAR-fusion

Hgene	Hchr	Hbp	Hstrand	Tgene	Tchr	Tbp	Tstrand
BCR	chr22	23632600	+	ABL1	chr9	133729450	+
BAG6	chr6	31619433	-	SLC44A4	chr6	31833561	-
NUP214	chr9	134074402	+	XKR3	chr22	17288973	-
⋮	⋮	⋮		⋮	⋮	⋮	

FusionAI takes the 20 Kb DNA sequence of fusion genes from the previous step and outputs two probabilities as not being used and being used as the fusion gene breakpoints (Figure 2).

3. Run FusionAI prediction script to predict the fusion breakpoint tendency from the FusionAI model. Here the $ INPUT_FILE is the output file after making the 20 Kb DNA sequence in the previous step. $COLA and $COLB are the DNA sequences of 5′ and 3′ fusion partner genes that were created from the previous step. If the user wants to run for one specific fusion gene, then set $INDEX_OF_FUSION as row index of interested line in the input file.

```
> python FusionAI_pred.py [-h] -f [INPUT_FILE] -m [MODEL, default: newdat_newmod_jj.h5] -o
[OUTPUT_FILE] -A [COLA] -B [COLB] -I [INDEX_OF_FUSION]

> python FusionAI_pred.py -f k562_starfusion.FusionAI.input -o k562_starfusion.FusionAI
.output -m newdat_newmod_jj.h5
```

Select high scored fusion genes (or interested fusion genes) from FusionAI output

Timing: < 5 s

From the output scores of FusionAI for the fusion candidates that were predicted from other tools, the users can select high scored or interested fusion genes. This can be done by the user in a text editor or another appropriate tool of choice. The users can stop the pipeline at this step if they do not need to do further analyses including feature importance analysis or drawing a landscape image of human genomic features in fusion genes, which take relatively long. With the output scores of FusionAI, still uses can reduce the false positives. For better understanding, Table 4 shows the comparison results among different cutoff of FusionAI scores, other prediction tools, and experimentally validated fusion genes. Table 5 shows the accuracy comparisons. When we used a higher threshold of FusionAI output scores, we could reduce the false positives efficiently.

4. Sort the FusionAI prediction output based on the FusionAI scores of individual fusion genes and select high-scored fusion genes. The users can choose the cutoff score, which should be larger than 0.5. Table 4 below shows the examples that were chosen with different cutoffs like 0.5 or 0.95. Then, the selected fusion genes will be used for further analyses such as screening of the

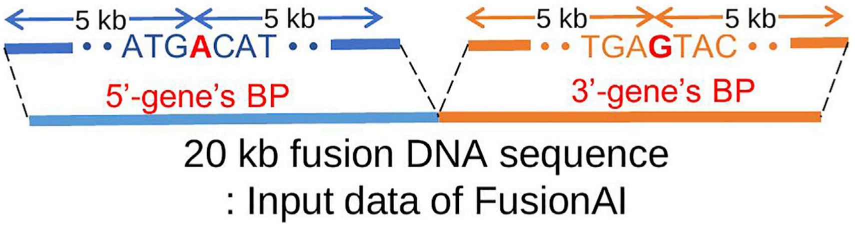


Figure 1. Make input data of FusionAI

Table 3. FusionAI input data example, which were made by running preprocessing script

Hgene	Hchr	Hbp	Hstrand	Tgene	Tchr	Tbp	Tstrand	20 Kb fusion DNA sequence
BCR	chr22	23632600	+	ABL1	chr9	133729450	+	TACCAGAGCGGCTGCCAAC…
BAG6	chr6	31619433	-	SLC44A4	chr6	31833561	-	CAGTGATGCTTCTGCCTCC…
NUP214	chr9	134074402	+	XKR3	chr22	17288973	-	GATAAAATTTTTTCACTAA…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

feature importance scores and landscaping the human genomic features across 20 Kb fusion DNA sequence in the following steps.

Calculate the feature importance scores across 20 Kb DNA sequence

Timing: < 33 min

After selecting the reliable fusion gene candidates, the users can check the distribution of the feature importance scores of individual fusion genes across the 20 Kb fusion DNA sequence. To calculate the feature importance score (FIS), we masked 20 bp each time by setting all the 20 values to zero and measured the change of prediction outcome upon this masking. We slide this 20 bp window 20 nucleotides each time along the whole 20K input sequence and repeated the procedure to obtain the FIS for all the 20 bp segments. In this way, we got 20,000/20 = 1,000 FIS for each input sequence.

5. Run FusionAI feature importance score script to get the feature importance scores across the 20 Kb fusion DNA sequence. Here the $ INPUT_FILE is the output file after making 20 Kb DNA sequence in step 3. $COLA and $COLB are the DNA sequences of 5′ and 3′ fusion partner genes that were created from step 3. If the user wants to run for one specific fusion gene, then set $INDEX_OF_FUSION, the row indexes of interested lines in the input file. If the user can use multiple GPUs, then the user can control the number of GPUs using the parameter of NGPUS. However, the GPU is not necessary (Figure 3).

```
> python FusionAI_FIS.py [-h] -f FILENAME [-m MODEL, default: newdat_newmod_jj.h5] [-o
OUTPUT] [-A COLA] [-B COLB] [-I ROWI] [-N NGPUS]

> python FusionAI_FIS.py -f k562_starfusion.FusionAI.output -o
k562_starfusion.FusionAI.output.FIS
```

Visualize 44 human genomic features across 20 Kb DNA sequence

Timing: < 1 h 10 min and < 20 min for step 6 and 7, respectively

After getting reliable fusion gene candidates and feature importance scores, it is important to interpret the aspect of human genomic features. From our original work, we integrated 44 human genomic features across five important cellular mechanism categories such as integration site category of 6 viruses, 13 types of repeat category, 5 types of structural variant category, 15 different types of chromatin state category, and 5 gene expression regulatory category (Kim et al., 2021a, 2021b). From this step, the users can create two figures on the landscape of the fusion gene breakpoint-related genomic features across the 20 Kb fusion DNA sequence. Each script will create separate figures of individual fusion genes that have the FIS values from the previous step. All figures will be created under the user defined directory. The first figure is the overlap between the 20 Kb fusion DNA sequence and 44 genomic features and the second figure is the overlap between the top 1% FIS regions and 44 genomic features (Figure 4).

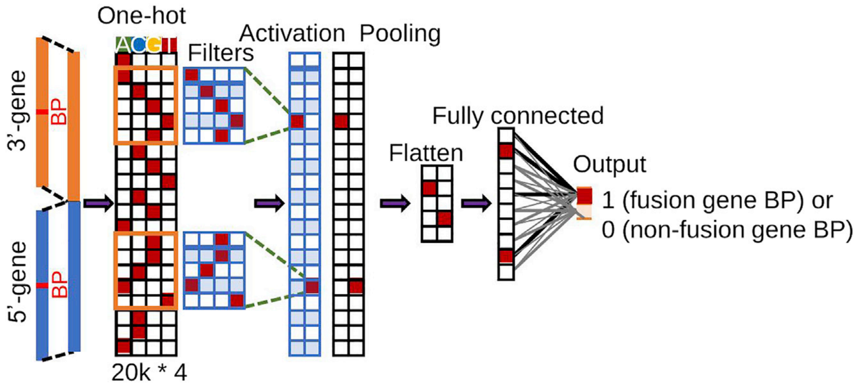


Figure 2. Diagram of fusion gene breakpoints classification by FusionAI

6. Visualize 44 human genomic features across a 20 Kb DNA sequence. Run FusionAI genomic feature analysis script to make a landscape image of overlap between fusion breakpoints area (+/- 5 Kb) and 44 human genomic features.

```
> Rscript FusionAI_genomic_features.R -g [FUSION_GENE_FILE] -f [FEATURE_PATH] -s [CHROMOSOME_SIZE_FILE] -i [FEATURE_INFO_FILE] -o [OUTPUT_FILE_PATH]

> Rscript FusionAI_genomic_features.r -g K562_STARfusion.FusionAI.output.FIS -f ./features/ -s chromosome_size.txt -i features_info.txt -o ./K562/whole_features/
```

7. Visualize the overlaps between the top 1% FIS regions and 44 human genomic features across 20 Kb DNA sequence. Run FusionAI genomic feature analysis script to have the landscape of overlap between high-FIS regions of fusion genes and 44 human genomic features.

```
> Rscript FusionAI_genomic_features2.R -g [FUSION_GENE_FILE] -f [FEATURE_PATH] -s [CHROMOSOME_SIZE_FILE] -i [FEATURE_INFO_FILE] -o [OUTPUT_FILE_PATH]

> Rscript FusionAI_genomic_features2.r -g K562_STARfusion.FusionAI.output.FIS -f ./features/ -s chromosome_size.txt -i features_info.txt -o ./K562/top1pct_features/
```

EXPECTED OUTCOMES

The above command will generate the following results from your fusion gene candidates' information: FusionAI output scores: FusionAI result will be saved in the current working directory with your preferred output file name. Feature importance scores: 1,000 feature importance scores of individual fusion genes that were resulted as the potential fusion breakpoints will be saved in the current working directory with your preferred output file name. Genomic feature landscape images: the distribution of 44 human genomic features across 20 Kb fusion DNA sequence will be saved in the current working directory with your preferred output file name.

LIMITATIONS

For prediction tasks, since our program provides an option of taking one fusion at a time, there is no problem running it on a CPU.

TROUBLESHOOTING

Problem 1

Installation of FusionAI fails due to uninstalled prerequisites (step 1).

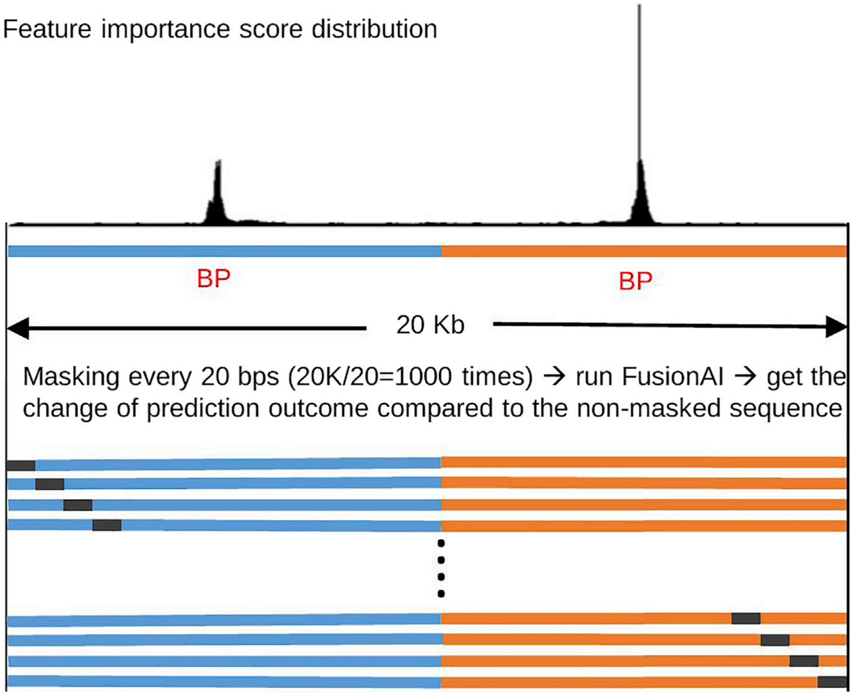

Feature importance score distribution
BP
BP
20 Kb
Masking every 20 bps (20K/20=1000 times) → run FusionAI → get the change of prediction outcome compared to the non-masked sequence

Figure 3. Calculate the feature importance scores across 20 Kb fusion DNA sequence

Potential solution

Please install the required dependencies manually through the links we provided in the key resources table, and then try installing FusionAI again.

Problem 2

Installation of FusionAI fails due to using old version python (step 1).

Potential solution

Please install the recent version of python at least v 3.0, and then try installing FusionAI again.

Problem 3

The preprocessing script fails to read the fusion gene information (step 2).

Potential solution

Please make the fusion gene information following the format described in step 2.

Problem 4

FusionAI fails to read the input file or parse it correctly.

Potential solution

Currently, FusionAI can only parse the tab- and space-separated file. Please check the format of the input file and make sure each column was properly separated and each row has the same number of columns.

Problem 5

FusionAI fails at the one-hot encoding step.

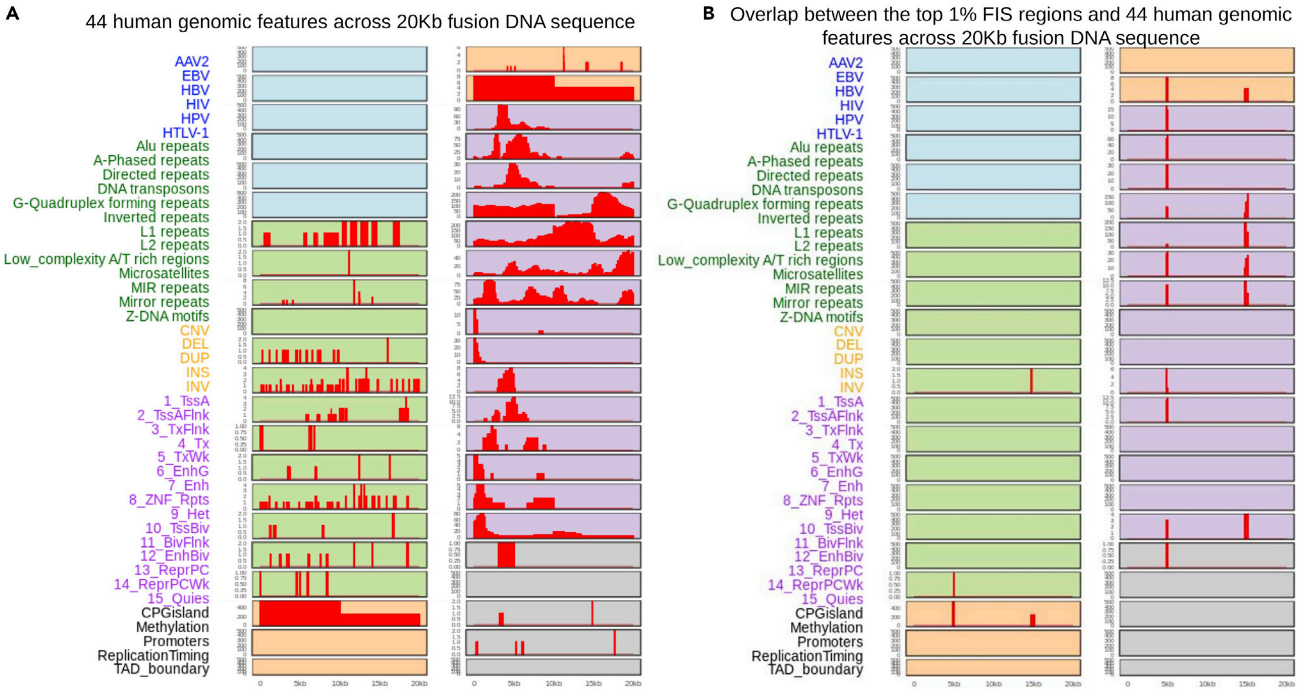


Figure 4. Left - distribution of 44 human genomic features across 20 Kb fusion DNA sequence

Right - overlap between the top 1% FIS regions and 44 different types of human genomic features across 20 Kb fusion DNA sequence.

Potential solution

Make Sure the input DNA sequences contain only five letters: A, C, G, T, and N.

Problem 6

Running FusionAI fails due to missing parameters (step 3).

Potential solution

Please provide the essential parameters to run FusionAI such as input and output file names, and then run FusionAI again.

Problem 7

Creating genomic feature landscape image fails due to not downloading human genomic feature information files (step 3).

Potential solution

Please download the human genomic feature information files from the link we provided in the key resources table, and then run the script again.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Pora Kim (pora.kim@uth.tmc.edu).

Materials availability

This study did not generate new unique reagents.

Table 4. Selection of common fusion genes between FusionAI and other tools based on the FusionAI score including validated fusion genes for the user's information

Hgene	Hchr	Hbp	Hstrand	Tgene	Tchr	Tbp	Tstrand	STAR-fusion	STAR-fusion & FusionAI >0.5	STAR-fusion & FusionAI >0.95	STAR-fusion & arriba	Validated	FusionAI score
BCR	chr22	23632600	+	ABL1	chr9	133729450	+	X	X	X	X	X	0.9999999
IMMP2L	chr7	111127293	-	DOCK4	chr7	111409733	-	X	X	X	X	X	0.9999999
BAG6	chr6	31619432	-	SLC44A4	chr6	31833561	-	X	X	X		X	0.99999857
RP11-344E13.3	chr17	20771998	+	UBBP4	chr17	21730694	+	X	X	X	X		0.9999932
BAG6	chr6	31619432	-	SLC44A4	chr6	31833378	-	X	X	X		X	0.9999831
C10orf76	chr10	103799769	-	KCNIP2	chr10	103588956	-	X	X	X	X		0.99743265
RP11-321F6.1	chr15	66874586	+	SMAD6	chr15	67004005	+	X	X	X			0.9900406
NUP214	chr9	134074402	+	XKR3	chr22	17288973	-	X	X	X	X	X	0.95663476
RP11-96H19.1	chr12	46781755	+	RP11-446N19.1	chr12	47046172	+	X	X				0.93317753
RP11-96H19.1	chr12	46781755	+	RP11-446N19.1	chr12	46965038	+	X	X				0.9303843
RP5-964N17.1	chrX	113181480	-	LRCH2	chrX	114398346	-	X	X				0.8816845
UPF3A	chr13	115070392	+	CDC16	chr13	115037658	+	X	X		X	X	0.8794392
CTC-786C10.1	chr16	85205413	+	RP11-680G10.1	chr16	85391068	+	X	X				0.8380846
C16orf87	chr16	46858297	-	ORC6	chr16	46729473	+	X	X		X		0.6423692
RP11-680G10.1	chr16	85391249	+	GSE1	chr16	85667519	+	X					0.30633911
C16orf87	chr16	46858297	-	ORC6	chr16	46727004	+	X			X		0.13516404
RP11-680G10.1	chr16	85391249	+	GSE1	chr16	85682157	+	X					0.040422514

Table 5. Accuracies across different comparisons of results for the users' information

	STAR-fusion	FusionAI > 0.5	FusionAI > 0.95	Arriba	Validated
TP	6	6	5	4	6
FP	11	8	3	4	0
TN	0	3	8	9	11
FN	0	0	1	2	0
Precision	0.35	0.43	0.63	0.50	1.00
Recall	1.00	1.00	0.83	0.67	1.00
Accuracy	0.35	0.53	0.76	0.68	1.00
F-measure	0.52	0.60	0.71	0.57	1.00
MCC	NA	0.34	0.54	0.34	1.00

Data and code availability

Code is available at https://compbio.uth.edu/FusionGDB2/FusionAI/.

ACKNOWLEDGMENTS

This work was partially supported by the National Institutes of Health grants (R35GM138184) to P.K. The funders had no role in study design, data collection, analysis, decision to publish, or preparation of the manuscript. Funding for open access charge: Startup Fund to P.K. from the University of Texas Health Science Center at Houston.

AUTHOR CONTRIBUTIONS

Model data preparation, P.K.; model development, H.T. and P.K.; genomic feature data preparation, P.K. and J.L.; visualization of genomic features, P.K. and J.L.; test, H.K.; manuscript writing, P.K.; figures, P.K.; supervision, P.K. and X.Z.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Analytics, R., and Weston, S. (2020). Iterators: Provides Iterator Construct, R Package Version 1.0.13.

Auguie, B. (2017). gridExtra: Miscellaneous Functions for "Grid" Graphics, R package version 2.3.

Bache, S.M., and Wickham, H. (2020). Magrittr: A Forward-Pipe Operator for R, R Package Version 2.0.1.

Corporation, M., and Weston, S. (2020). doParallel: foreach parallel adaptor for the 'parallel' package, R package version 1.0.16.

Davis, T. (2018). Optparse: Command Line Option Parser, R Package version 1.6. 0.

Haas, B.J., Dobin, A., Li, B., Stransky, N., Pochet, N., and Regev, A. (2019). Accuracy assessment of fusion transcript detection via read-mapping and de novo fusion transcript assembly-based methods. Genome Biol. *20*, 213.

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

Kassambara, A. (2018). ggpubr: 'ggplot2' Based Publication Ready Plots, R package version 0.1.7.

Kim, P., Jang, Y.E., and Lee, S. (2019). FusionScan: accurate prediction of fusion genes from RNA-Seq data. Genomics Inform. *17*, e26.

Kim, P., Tan, H., Liu, J., Lee, H., Jung, H., Kumar, H., and Zhou, X. (2021a). FusionGDB 2.0: fusion gene annotation updates aided by deep learning. Nucleic Acids Res. *50*, D1221–D1230.

Kim, P., Tan, H., Liu, J., Yang, M., and Zhou, X. (2021b). FusionAI: predicting fusion breakpoint from DNA sequence with deep learning. iScience *24*, 103164.

McPherson, A., Hormozdiari, F., Zayed, A., Giuliany, R., Ha, G., Sun, M.G., Griffith, M., Heravi Moussavi, A., Senz, J., Melnyk, N., et al. (2011). deFuse: an algorithm for gene fusion discovery in tumor RNA-Seq data. PLoS Comput. Biol. *7*, e1001138.

Microsoft, and Weston, S. (2020). Foreach: Provides Foreach Looping Construct, R package version 1.5.1.

Patwardhan, M.N., Wenger, C.D., Davis, E.S., and Phanstiel, D.H. (2019). Bedtoolsr: an R package for genomic data analysis and manipulation. J. Open Source Softw. *4*, 1742.

Quinlan, A.R., and Hall, I.M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics *26*, 841–842.

Team, R.C. (2019). R: A Language and Environment for Statistical Computing (R Foundation for Statistical Computing).

Uhrig, S., Ellermann, J., Walther, T., Burkhardt, P., Frohlich, M., Hutter, B., Toprak, U.H., Neumann, O., Stenzinger, A., Scholl, C., et al. (2021). Accurate and efficient detection of gene fusions from RNA sequencing data. Genome Res. *31*, 448–460.

Wickham, H., Hester, J., and Chang, W. (2018). Devtools: Tools to Make Developing R Packages Easier, R package version 1.1.3.6.

Wickham, H., and Seidel, D. (2020). Scales: Scale Functions for Visualization, R Package version 1.1.1.

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis (Springer-Verlag).

Wilke, C.O. (2020). Cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2', R Package version 1.1.1.