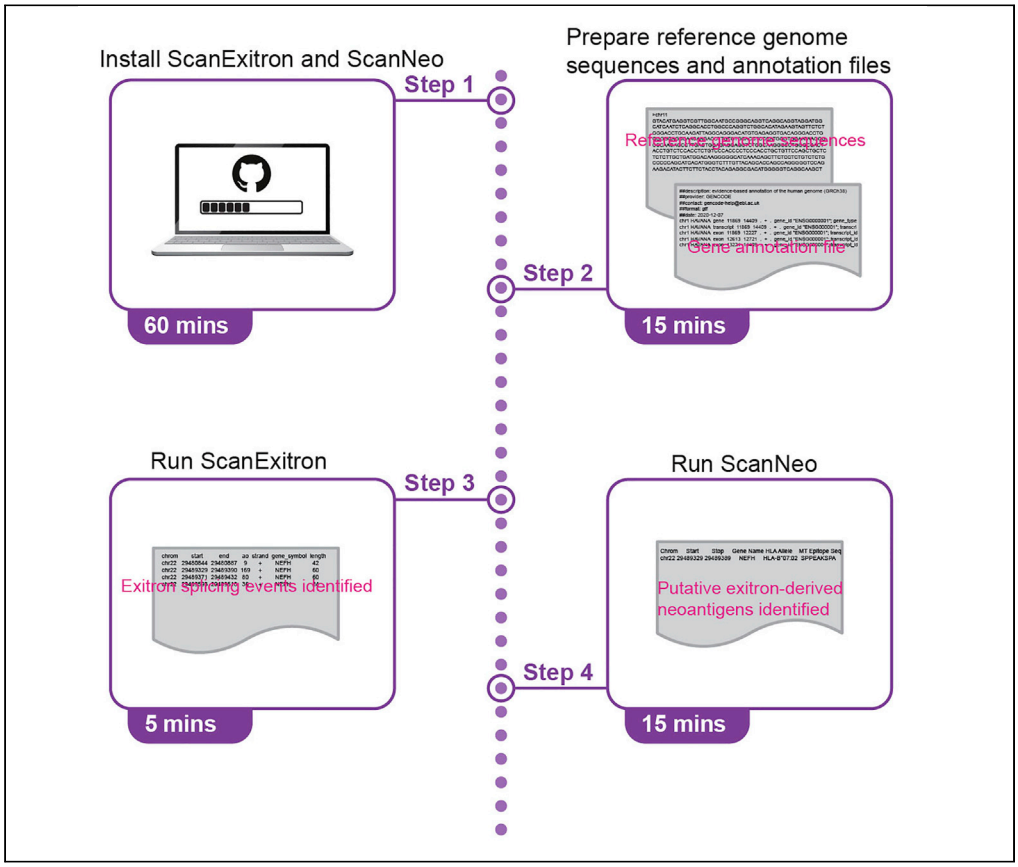
STAR Protocols

CellPress
OPEN ACCESS

Protocol

Integrated protocol for exitron and exitron-derived neoantigen identification using human RNA-seq data with ScanExitron and ScanNeo



Exitron splicing (EIS) events in cancers can disrupt functional protein domains to cause cancer driver effects. EIS has been recognized as a new source of tumor neoantigens. Here, we describe an integrated protocol for EIS and EIS-derived neoantigen identification using RNA-seq data. The protocol constitutes a step-by-step guide from data collection to neoantigen prediction.

Ting-You Wang, Rendong Yang

tywang@umn.edu (T.-Y.W.)
yang4414@umn.edu (R.Y.)

Highlights

A protocol for identifying exitron and exitron-derived neoantigens

Special focus on data preparation, and troubleshooting

Optional steps for applying this protocol to analyze TCGA PRAD cancer cohort

Wang & Yang, STAR Protocols 2, 100788
September 17, 2021
https://doi.org/10.1016/j.xpro.2021.100788

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Integrated protocol for exitron and exitron-derived neoantigen identification using human RNA-seq data with ScanExitron and ScanNeo

Ting-You Wang[1,3,*] and Rendong Yang[1,2,4,*]

[1]The Hormel Institute, University of Minnesota, Austin, MN 55912, USA

[2]Masonic Cancer Center, University of Minnesota, Minneapolis, MN 55455, USA

[3]Technical contact

[4]Lead contact

*Correspondence: tywang@umn.edu (T.-Y.W.), yang4414@umn.edu (R.Y.)
https://doi.org/10.1016/j.xpro.2021.100788

SUMMARY

Exitron splicing (EIS) events in cancers can disrupt functional protein domains to cause cancer driver effects. EIS has been recognized as a new source of tumor neoantigens. Here, we describe an integrated protocol for EIS and EIS-derived neoantigen identification using RNA-seq data. The protocol constitutes a step-by-step guide from data collection to neoantigen prediction.
For complete details on the use and execution of this protocol, please refer to Wang et al. (2021).

BEFORE YOU BEGIN

Data collection

This integrated protocol to analyze RNA sequencing (RNA-seq) data includes two components: ScanExitron (Wang et al., 2021) and ScanNeo (Wang et al., 2019) (Figure 1). ScanExitron was designed to detect exitron splicing events from short-read RNA-seq data, such as those produced by the Illumina sequencing platform from The Cancer Genome Atlas (TCGA) study (Wang et al., 2021). ScanNeo was originally developed for insertion and deletion (indel) derived neoantigen detection. Because of the similarity between deletions and EIS events in their effects changing protein sequences, ScanNeo is capable of detecting exitron-derived neoantigen directly. By definition, exitrons are cryptic introns with both their splice sites inside an annotated protein-coding exon. Therefore, human reference gene annotation is needed to identify *bona fide* exitrons. We recommend using the GRCh38 gene annotation GTF file from the GENCODE project (Frankish et al., 2019).

The protocol below describes ScanExitron applications analyzing a toy example data set and a real data set from the TCGA prostate cancer (PRAD) cohort, respectively.

> ***Note:*** Example data can be found at https://github.com/ylab-hi/ScanExitron/tree/master/example_data.

The RNA-seq alignment files in BAM format for TCGA PRAD cohort can be downloaded from NCI Genomic Data Commons (https://portal.gdc.cancer.gov/). A single representative aliquot was selected per participant for cases where more than one aliquot was available. Thus, 496 PRAD primary tumor samples and 52 normal samples were kept.

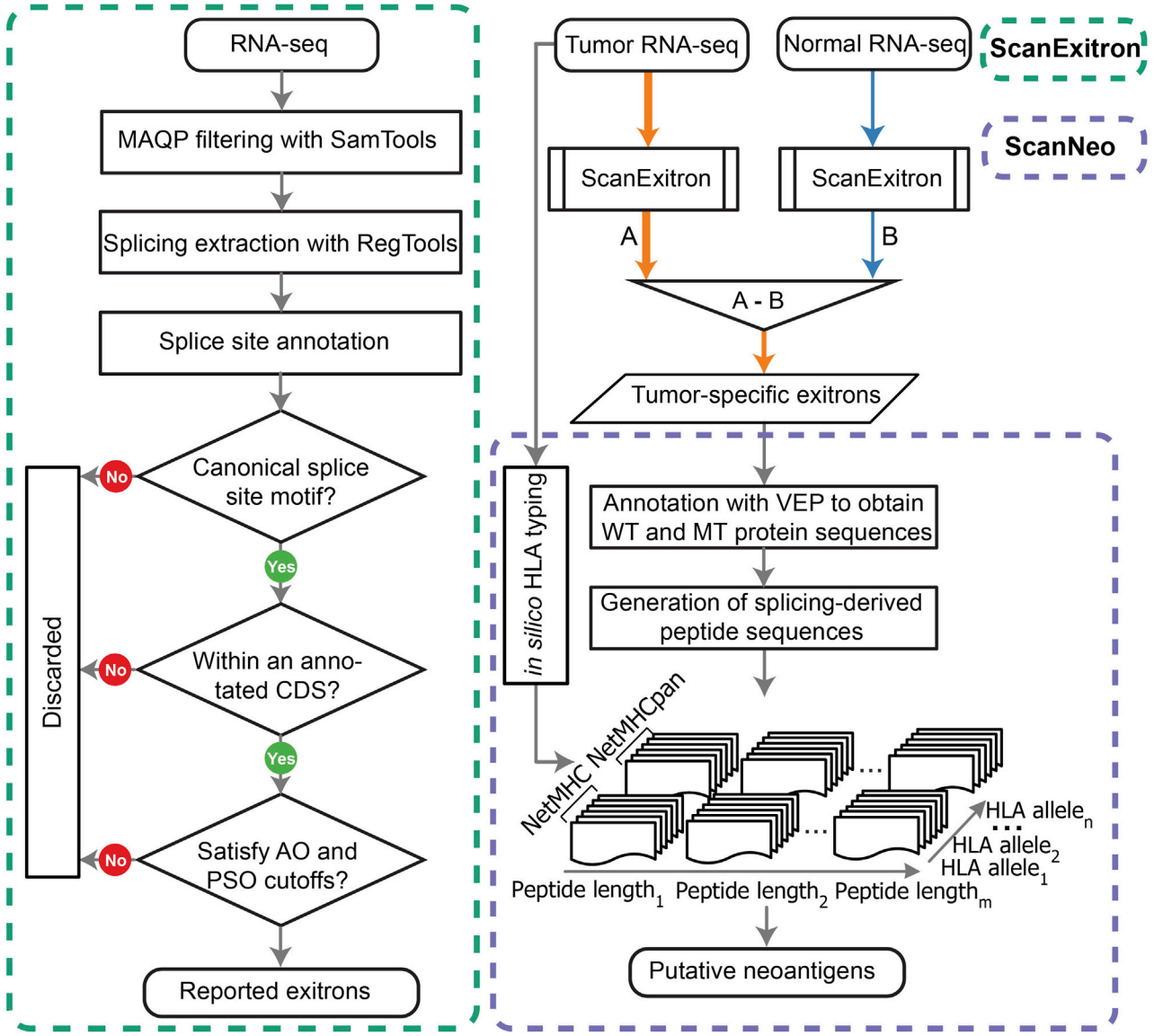


Figure 1. Flow chart showing exitron and exitron-derived neoantigens detection with ScanExitron and ScanNeo

HLA class I four-digit types of 495 out of 496 TCGA PRAD samples were obtained from (Thorsson et al., 2018) (https://gdc.cancer.gov/about-data/publications/panimmune). For the remaining one sample used in this study, ScanNeo was employed for HLA class I typing.

Optional reads alignment

If the users are dealing with their in-house RNA-seq data in raw FASTQ format, the alignment step will be needed before running the protocol. ScanExitron requires the input to be a BAM file, which is provided by a splice-aware aligner, such as HISAT2 (Kim et al., 2019). We recommend aligning the raw read FASTQ file using HISAT2 with Hierarchical Graph Ferragina-Manzini (HGFM) index built with known transcripts annotations. Users can build the HGFM index on their own (http://daehwankimlab.github.io/hisat2/howto/#build-hgfm-index-with-transcripts) or download the HGFM index (genome_tran) directly (http://daehwankimlab.github.io/hisat2/download/).

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Example data (example.bam file)	This paper	https://github.com/ylab-hi/ScanExitron/blob/master/example_data/
RNA-Seq data from TCGA PRAD cohort	NCI Genomic Data Commons	https://portal.gdc.cancer.gov
HLA types for TCGA cohort	Thorsson et al., 2018	https://gdc.cancer.gov/about-data/publications/panimmune
GENCODE human gene annotations	Frankish et al., 2019	https://www.gencodegenes.org/human/
Human reference genome NCBI build 38, GRCh38	Genome Reference Consortium	http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/human/
Software and algorithms		
HISAT2	Kim et al., 2019	RRID:SCR_015530; http://daehwankimlab.github.io/hisat2/
ScanExitron	Wang et al., 2021	https://github.com/ylab-hi/ScanExitron
Pyfaidx v0.5.9.2	Shirley et al., 2015	https://github.com/mdshw5/pyfaidx
SamTools v1.12	Li et al., 2009	RRID:SCR_00210; http://www.htslib.org/
BEDTools v2.26.0	Quinlan, 2014	RRID:SCR_006646; https://github.com/arq5x/bedtools2
RegTools v0.4.2	Feng et al., 2018	https://github.com/griffithlab/regtools
ScanNeo	Wang et al., 2019	RRID:SCR_019253; https://github.com/ylab-hi/ScanNeo
transIndel v2.0	Yang et al., 2018	https://github.com/cauyrd/transIndel
OptiType v1.2	Szolek et al., 2014	https://github.com/FRED-2/OptiType
Yara aligner v1.0.2	Siragusa et al., 2013	https://github.com/seqan/seqan/tree/master/apps/yara
Variant Effect Predictor v102.0	McLaren et al., 2016	RRID:SCR_007931; https://useast.ensembl.org/info/docs/tools/vep/script/index.html
Sambamba v0.8.0	Tarasov et al., 2015	https://lomereiter.github.io/sambamba/
IEDB MHC class I peptide binding prediction tools v3.1	Vita et al., 2019	https://downloads.iedb.org/tools/mhci/3.1/
BWA v0.7.17	Li and Durbin, 2009	RRID:SCR_010910; http://bio-bwa.sourceforge.net/
PyVCF v0.6.8	N/A	https://github.com/jamescasbon/PyVCF/
Picard v2.24.0	Broad Institute	https://broadinstitute.github.io/picard/
HDF5 v1.10.4	The HDF Group	http://www.hdfgroup.org/HDF5/
Tabix v1.12	Li et al., 2009	RRID:SCR_00210; http://www.htslib.org/
Other		
PC with 4 CPU cores and 16GB RAM	AMD	N/A
HPC system with 16 CPU cores and 64GB RAM	AMD	N/A

MATERIALS AND EQUIPMENT

Data (RNA-seq alignment files in BAM format – see data collection in before you begin)

Software

ScanExitron and its dependencies. ScanExitron is implemented in Python 3. While different versions of the Python software and associated packages may work correctly with ScanExitron, the authors use Python 3.7 and the following packages at the indicated versions when writing this protocol:

- pyfaidx (v0.5.9.2)
- SamTools (v1.12)

- BEDTools (v2.26.0)
- RegTools (v0.4.2)

Note: ScanExitron is not compatible with RegTools (v0.5 or above) in its current design.

ScanNeo and its dependencies. ScanNeo is also implemented in Python 3. When writing this protocol, the authors use Python 3.7 and the following packages at the indicated versions:

- transIndel (v2.0)
- IEDB MHC class I peptide binding prediction tools (v3.1)
- optitype (v1.3.5)
- BWA (v0.7.17)
- Sambamba (v0.8.0)
- BEDTools (v2.26.0)
- Variant Effect Predictor (v102.0)
- coincbc (v2.10.5)
- razers3 (v3.5.8)
- Picard (v2.24.0)
- Yara (v1.0.2)
- pyomo (v5.7.3)
- PyVCF (v0.6.8)
- HDF5 (v1.10.4)
- tabix (v1.12)
- pyfaidx (v0.5.9.2)

STEP-BY-STEP METHOD DETAILS

Step 1: Installing ScanExitron and ScanNeo

Timing: 60 min

Full installation of ScanExitron and ScanNeo includes downloading the ScanExitron and ScanNeo packages from GitHub. An example of how to perform all steps of this protocol using example data is available on the project GitHub at https://github.com/ylab-hi/ScanExitron/wiki/Exitron-and-exitron-derived-neoantigen-identification-with-ScanExitron-and-ScanNeo

1. Installing ScanExitron
   a. Install ScanExitron dependencies
      i. Install RegTools v0.4.2

```
$ git clone –depth 1 –branch 0.4.2 https://github.com/griffithlab/regtools.git
```

      ii. Install other dependent packages via conda.

```
$ conda install -c bioconda samtools bedtools pyfaidx
```

   b. Install ScanExitron by running the following code:

```
$ git clone https://github.com/ylab-hi/ScanExitron.git
```

⚠ CRITICAL: Check if all required dependencies are downloaded and installed correctly. Originally, installing packages via conda will automatically check for and install the required dependencies. However, errors during installation could occur when installing on computational environments (Troubleshooting 1 and Troubleshooting 2).

2. Installing ScanNeo
   a. Install ScanNeo dependencies
      i. Install transIndel v2.0

```
$ git clone https://github.com/cauyrd/transIndel
```

Add the directory of transIndel_build_RNA.py and transIndel.py to the $PATH environment variable.

ii. Install IEDB HLA class I binding prediction tools (https://downloads.iedb.org/tools/mhci/3.1/IEDB_MHC_I-3.1.tar.gz)

iii. Install other dependent packages via conda.

```
$ conda install -c bioconda optitype ensembl-vep sambamba bedtools picard bwa yara razers3 pyfaidx pyvcf

$ conda install -c conda-forge coincbc

$ conda install -c anaconda hdf5
```

iv. Install VEP annotations and plugins
Install VEP annotations using the following command.

```
$ vep_install -a cf -s homo_sapiens -y GRCh38 –CONVERT
```

Note: Before install VEP annotations, make sure the directory of executable file vep_install is in the $PATH environment variable (Troubleshooting 3).

Install two VEP plugins for ScanNeo.

```
$ git clone https://github.com/ylab-hi/ScanNeo.git

$ cd VEP_plugins

$ cp Downstream.pm ~/.vep/Plugins

$ cp Wildtype.pm ~/.vep/Plugins
```

v. Configure optitype and yara index according to the ScanNeo manual (https://github.com/ylab-hi/ScanNeo).

   b. Install ScanNeo by running the following code:

```
$ git clone https://github.com/ylab-hi/ScanNeo.git
```

Note: Make sure the directories of all the executable files are in the $PATH environment variable.

Step 2: Preparing the reference genome sequences and gene annotation files

Timing: 15 min

ScanExitron utilized annotated coding sequence (CDS) regions to probe the exitrons, and it also extracted splice sites using the reference genome sequences. The human reference genome sequences and gene annotation will be used.

3. Preparing human reference genome sequences in FASTA format.

Download hg38 FASTA human reference genomes from UCSC genome browser (https://hgdownload.cse.ucsc.edu/goldenpath/hg38/bigZips/hg38.fa.gz) and unzip it.

4. Preparing reference gene annotation in GTF format.
   a. Download hg38 annotation file from GENCODE project (ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_37/gencode.v37.annotation.gtf.gz)
   b. Extract the protein-coding CDS regions

In Unix/Linux system, the protein-coding exons regions can be extracted using "cat", "awk" and "tr" commands, as followed.***Note:*** Make sure the input RNA-seq BAM files used the same coordinate

```
$ cat gencode.v37.annotation.gtf | awk 'OFS="\t" {if ($3=="CDS") {print $1,$4-1,$5,$10,$16,$7}}' | tr -d '";' > gencode.hg38.CDS.bed
```

system as the reference genome and the reference annotations files. Otherwise, you have to remap the RNA-seq reads with the corresponding reference genome.

Step 3: Running ScanExitron

Timing: 5 min

After installing all of the dependencies and preparing the reference genome sequences and annotation files, it is time to run ScanExitron. ScanExitron can be used only in UNIX/Linux systems currently. Additional details for running ScanExitron and updates to the parameters can be found at the project GitHub repository (https://github.com/ylab-hi/ScanExitron).

Note: Here we only provided the running time for the toy example dataset, which contains three exitrons. The actual running time for the real sample is dependent on the number of junction reads and the number of exitrons in it.

5. Make necessary modifications to the configuration file of ScanExitron.

Replace the items in config.ini with the reference genome sequences and annotation files prepared in step 2: preparing the reference genome sequences and gene annotation files. The example config.ini file can be found at https://github.com/ylab-hi/ScanExitron/blob/master/config.ini.example (Troubleshooting 4).

6. Run ScanExitron with the following command:

```
$ ScanExitron.py -i example.bam -ao 3 -pso 0.05 -m 50 -r hg38
```

⚠ CRITICAL: Make sure the input RNA-seq BAM files used the same coordinate system as the reference genome and the reference annotations files (Troubleshooting 5).

Note: In practice, the different parameter settings will result in the different number of exitrons identified. For example, if you set a higher alternate allele observation (AO) and percent spliced out (PSO) (Wang et al., 2021), you will get a smaller number of exitrons. The details for these two metrics are described in quantification and statistical analysis. Additional details for running ScanExitron and updates to the parameters can be found at the project GitHub repository (https://github.com/ylab-hi/ScanExitron).

Multiple files will be generated in this step, including "example.hq.bam", "example.hq.bam.bai", "example.hq.janno" and "example.exitron".

The identified exitrons are stored in the example.exitron file (Table 1). Figure 2 illustrates these detected EIS events using Integrative Genomics Viewer (IGV) (Robinson et al., 2011).

Note: Differential analysis will be available if researchers have groups of samples of interest (Troubleshooting 6).

In order to feed the ScanExitron results to ScanNeo, output files of ScanExitron are required to be converted to VCF format using the utility script named exitron2vcf.py contained in the ScanExitron utils folder with the following command:***Note:*** The directory of exitron2vcf.py should be in the

```
$ exitron2vcf.py -i example.exitron -o example.vcf
```

$PATH environment variable.

Step 4: Running ScanNeo

Timing: 15 min

After running ScanExitron for the sample dataset, we get a list of exitron splicing events in the example.vcf file. In practice, you have to also run ScanExitron for the corresponding normal samples aiming to obtain exitrons that are tumor specific. Here, we assume all the exitrons identified in the sample dataset are tumor-specific exitrons (TSEs).

It is time to run ScanNeo to generate exitron-derived neoantigens. ScanNeo can be used only in UNIX/Linux systems currently. Additional details for running ScanNeo and updates to the parameters can be found at the project GitHub repository (https://github.com/ylab-hi/ScanNeo).

7. Make necessary modifications to the configuration file of ScanNeo.

Replace the items in config.ini with the reference genome sequences and gene annotation files prepared in step 2: preparing the reference genome sequences and gene annotation files. The example config.ini file can be found at https://github.com/ylab-hi/ScanNeo/blob/master/config.ini.example. (Troubleshooting 4)

Note: The reference genome sequences field is mandatory for this protocol. The gene annotation field is necessary when calling indels using ScanNeo. Yara HLA index field is necessary when HLA typing using ScanNeo.

8. Run ScanNeo
   a. ScanNeo first added corresponding reference and alternate allele sequences to each EIS event. Next, these events were annotated with variant effect predictor (VEP) (McLaren et al., 2016). Run this annotation step of ScanNeo using the following command.

```
$ ScanNeo.py anno -i example.vcf -o example.vep.vcf
```

   b. Neoantigen prediction step of ScanNeo used VEP annotated VCF file as input to predict neoantigens using the following command.

Table 1. The identified exitrons in the example data set

chrm:start-end	ao	strand	gene_symbol	Length	splice_site	pso	psi	dp
chr22:29489329–29489390	169	+	NEFH	60	GC-AG	0.261	0.739	648
chr22:29489371–29489432	80	+	NEFH	60	GT-AG	0.115	0.885	696
chr22:29489593–29489618	36	+	NEFH	24	GC-AG	0.0848	0.915	424

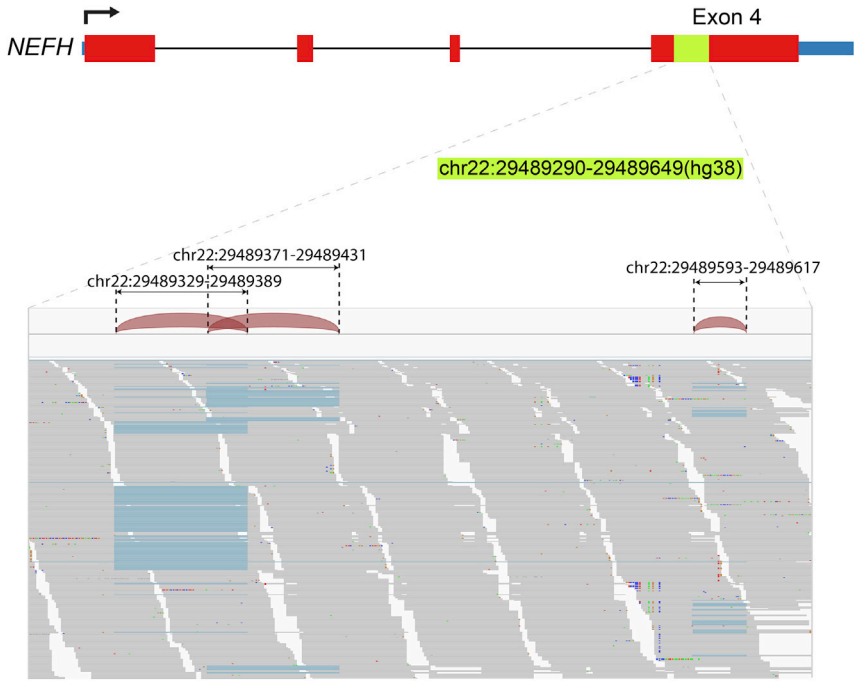


Figure 2. Three exitron splicing (EIS) events identified in *NEFH* gene loci by ScanExitron from the example RNA-seq data

```
$ ScanNeo.py hla -i example.vep.vcf -alleles HLA-A*68:02,HLA-A*23:01,HLA-B*07:02,HLA-B*53:01,HLA-C*07:02,HLA-C*04:01 -t 16 -af PSO -e 9 -p /path/to/iedb/ -o example.tsv
```

The putative exitron-derived neoantigens are stored in the example.tsv file (Table 2).

Note: This is a good time to compare your output results files to example files provided in the ScanExitron GitHub repository (https://github.com/ylab-hi/ScanExitron/tree/master/example_data) to ensure that you have run the protocol correctly.

Pause point: Once you know the parameters you wish to use and have successfully run ScanExitron and ScanNeo, you may find this to be a good place to pause and evaluate the results before proceeding with the optional steps.

Optional step 5: Running this protocol for TCGA PRAD cohort

Timing: 15 h

As a matter of fact, we have to use exitrons that are tumor-specific to predict neoantigens. We used TCGA PRAD cohort that includes 496 tumor and 52 tumor-adjacent normal samples to demonstrate how to use this protocol.

9. For every sample in TCGA PRAD cohort, we identified EIS events of PRAD tumor and normal samples following the instructions in step 3: running ScanExitron. Then we generated a list

Table 2. The predicted exitron-derived neoantigens in the example data set

Chrom	Start	Stop	Gene name	HLA allele	Peptide length	MT epitope seq	WT epitope seq	Best MT score method	Best MT score	Corresponding WT score
chr22	29489329	29489389	NEFH	HLA-B*07:02	9	SPPEAKSPA	SPPEAKSPE	NetMHCpan	399.52	7247.45

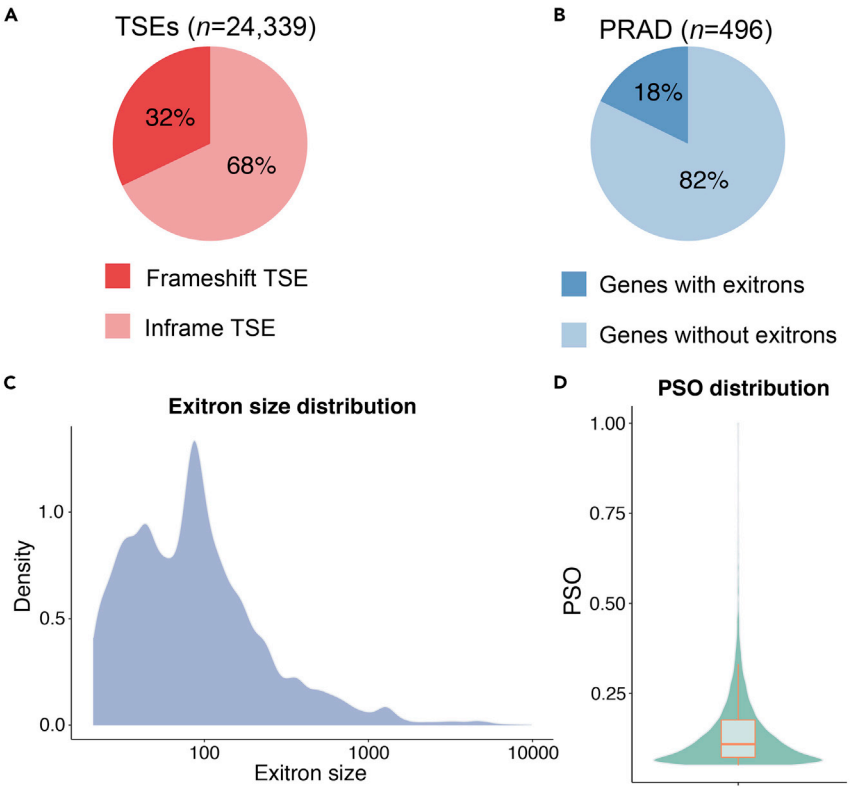


Figure 3. Tumor-specific exitron (TSE) splicing events detection in PRAD cohort

(A) The proportion of frameshift and inframe TSEs in PRAD tumors.

(B) The proportion of genes with and without exitrons in PRAD tumors.

(C) Exitron size distribution of TSEs identified in PRAD tumors.

(D) PSO distribution of TSEs identified in PRAD tumors.

of tumor-specific exitrons (TSEs) by excluding the EIS events in tumor samples that were also found in more than three normal samples. We achieved this filtering process using in-house Python scripts, which are available at https://github.com/ylab-hi/ScanExitron/wiki/Exitron-and-exitron-derived-neoantigen-identification-with-ScanExitron-and-ScanNeo. A summary of identified exitrons and TSEs in PRAD is described in Figure 3.

10. Run step 8 using the same parameters for TSEs of every sample in VCF format, we identified exitron-derived neoantigens for PRAD cohort (Figure 4).

Note: The timing didn't include downloading PRAD BAM files. In step 9, we submitted 16 jobs in the Slurm queue system. Every job only required one CPU core. In step 10, we used 20 jobs, every job required 16 CPU cores. Because ScanNeo implemented a parallel computing architecture, we highly recommend users set more CPU cores for it.

EXPECTED OUTCOMES

At the end of the process of the example dataset, you will have two main text files; (1) the exitron splicing events identified (Data showed in Table 1 and Figure 2) and (2) the predicted exitron-derived neoantigens (Data showed in Table 2). At the end of the process of the PRAD RNA-seq dataset, you will have TSE events for 496 PRAD patients and the corresponding predicted neoantigens (Data plotted in Figures 3 and 4).

QUANTIFICATION AND STATISTICAL ANALYSIS

For every exitron splicing event identified, we used two measurements to quantify the exitron splicing event, that is, AO and PSO (Wang et al., 2021) . AO is the number of splice junction reads

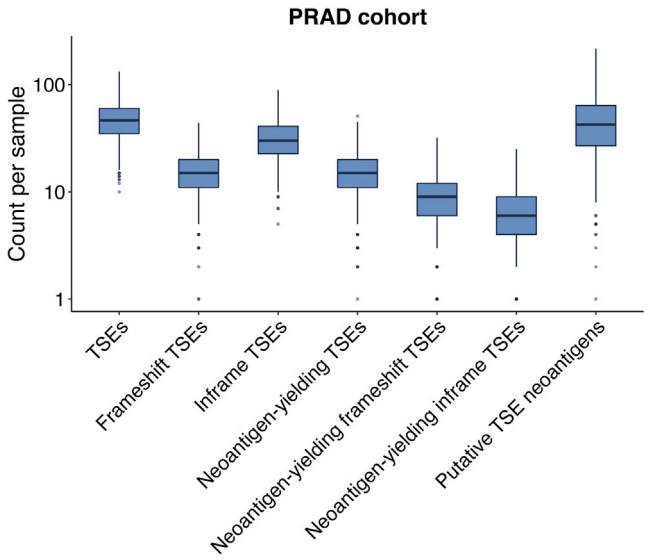


Figure 4. The loads of TSEs, frameshift TSEs, inframe TSEs, neoantigen-yielding TSEs, neoantigen-yielding frameshift TSEs, neoantigen-yielding inframe TSEs, and putative TSE neoantigens in PRAD tumors

supporting exitron splicing. PSO metric was used to measure the percentage of transcripts in which a given exitron is spliced. Generally speaking, higher AO and PSO metrics indicated exitron splicing events with high confidence. Besides AO and PSO, we also reported percent spliced-in (PSI) (Schafer et al., 2015) as the counterpart of PSO and the average depth of the identified exitron splicing event in the ScanExitron output. Additional details for ScanExitron results can be found at the project GitHub repository (https://github.com/ylab-hi/ScanExitron).

LIMITATIONS

The accuracy of exitron identification with ScanExitron is dependent on the accuracy of splice junctions from the RNA-seq BAM file and the completeness of CDS annotations. Firstly, due to the complexity of alternative splicing within a gene and the short-reads length, splice-aware aligners could produce large numbers of false-positive junctions (Engstrom et al., 2013). There is no optimal solution so far. But we can still mitigate it in two ways. One way is to make aligners prefer to use known splice sites by using the index built with known transcripts annotations, as we suggested in the optional reads alignment section. The other obvious way is to increase the read length when possible. Secondly, even for model organisms such as human, the reference annotations are incomplete, thus genuine exitrons with supporting junctions may be missed owing to the lack of overlapped annotated CDS annotations. Thus, in practice, we highly suggested using the latest gene annotations when possible.

Currently, the neoantigen prediction workhorse of this protocol, ScanNeo, only supports two well-established and popular MHC class I prediction algorithms, aka, NetMHC (Lundegaard et al., 2008) and NetMHCpan (Nielsen and Andreatta, 2016). Alternative versatile prediction algorithms should be used for neoantigen prediction. Thus, we plan to update ScanNeo to incorporate more MHC class I prediction approaches.

TROUBLESHOOTING

Problem 1

Install the software-dependent packages (Steps 1 and 2).

Potential solution

When possible, use Anaconda (https://www.anaconda.com/) to install Python 3 and its dependent packages. To order to avoid potential conflicts with installed Python packages, you can create a new conda environment to install all the necessary packages using the "conda create" command.

Problem 2

Software versions specific requirements (Steps 1 and 2).

Potential solution

Make sure that Python and other dependencies versions are appropriate.

You can use Anaconda to specify the version of the installed package, using the following commands:Or use GitHub tag to specify the package version.

```
$ conda install <package>=<version>
```

Problem 3

```
$ git clone –depth 1 –branch <version> https://github.com/<package>.git
```

You are receiving a "command not found" error message (Step 2), when you are trying to install VEP annotations using vep_install or run other conda installed executable files such as bedtools and sambamba. This indicated that the executable files are not in the $PATH environment variable.

Potential solution

Add Anaconda bin directory to the $PATH environment variable in the file ~/.bashrc.

```
export PATH="/path/to/Anaconda3/Python3/bin:$PATH"
```

Problem 4

You are receiving a "configparser.NoSectionError" error message (Step 5 and 7).

Potential solution

Place config.ini file to the location of ScanExitron or ScanNeo.

Problem 5

You are receiving an "Errors in BED line" error message (Step 6). This indicated the input RNA-seq BAM file used GRCh37/GRCh38 contig names, such as '1', '2', instead of hg37/hg38 contig names, such as 'chr1', 'chr2'.

Potential solution

If you have the raw RNA-seq reads in FASTQ format, you can realign the reads using hg38/hg19 reference genome sequences. Otherwise, you can extract the reads from the RNA-seq BAM file using Picard SamToFastq (https://broadinstitute.github.io/picard/command-line-overview.html#SamToFastq), then realign the reads.

Problem 6

How to perform a differential analysis of exitrons between two groups of samples (Step 6 and Table 1).

Potential solution

First, following steps 1–6, you can detect a list of exitrons for every sample. Second, organize the exitron results of all samples to form a table of PSO values. In this table, you should put PSO values in the cell for the corresponding row (exitron splicing event) and column (sample). Because you have two groups of samples, you can use a linear model or statistical tests (e.g., T-test) to calculate the statistical significance (p-value) for each exitron. If there are multiple exitrons in the table, multiple testing correction is needed to adjust the p-values.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Rendong Yang (yang4414@umn.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The example data set for this study is available at https://github.com/ylab-hi/ScanExitron.

ACKNOWLEDGMENTS

We acknowledge the following sources of funding: DoD (W81XWH-19-1-0161) to R.Y. and Eagles Telethon Postdoctoral Fellowship to T.-Y.W. We thank Dr. Jeffrey McDonald at The Hormel Institute for his technical support for computing facilities. Support from the Minnesota Supercomputer Institute (MSI) is also gratefully acknowledged.

AUTHOR CONTRIBUTIONS

Writing, T.-Y.W. and R.Y.; development and processing, T.-Y.W.; funding acquisition, R.Y.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Engstrom, P.G., Steijger, T., Sipos, B., Grant, G.R., Kahles, A., Ratsch, G., Goldman, N., Hubbard, T.J., Harrow, J., Guigo, R., et al. (2013). Systematic evaluation of spliced alignment programs for RNA-seq data. Nat. Methods *10*, 1185–1191.

Feng, Y.-Y., Ramu, A., Cotto, K.C., Skidmore, Z.L., Kunisaki, J., Conrad, D.F., Lin, Y., Chapman, W.C., Uppaluri, R., Govindan, R., Griffith, O.L., and Griffith, M. (2018). RegTools: Integrated analysis of genomic and transcriptomic data for discovery of splicing variants in cancer. bioRxiv. https://doi.org/10.1101/436634.

Frankish, A., Diekhans, M., Ferreira, A.M., Johnson, R., Jungreis, I., Loveland, J., Mudge, J.M., Sisu, C., Wright, J., Armstrong, J., et al. (2019). GENCODE reference annotation for the human and mouse genomes. Nucleic Acids Res. *47*, D766–D773.

Kim, D., Paggi, J.M., Park, C., Bennett, C., and Salzberg, S.L. (2019). Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. Nat. Biotechnol. *37*, 907–915.

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

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R.; 1000 Genome Project Data Processing Subgroup (2009). The Sequence Alignment/Map format and SAMtools. Bioinformatics *25*, 2078–2079.

Lundegaard, C., Lamberth, K., Harndahl, M., Buus, S., Lund, O., and Nielsen, M. (2008). NetMHC-3.0: accurate web accessible predictions of human, mouse and monkey MHC class I affinities for peptides of length 8-11. Nucleic Acids Res. *36*, W509–W512.

McLaren, W., Gil, L., Hunt, S.E., Riat, H.S., Ritchie, G.R., Thormann, A., Flicek, P., and Cunningham, F. (2016). The ensembl variant effect predictor. Genome Biol. *17*, 122.

Nielsen, M., and Andreatta, M. (2016). NetMHCpan-3.0; improved prediction of binding to MHC class I molecules integrating information from multiple receptor and peptide length datasets. Genome Med. *8*, 33.

Quinlan, A.R. (2014). BEDTools: The Swiss-Army Tool for Genome Feature Analysis. Curr. Protoc. Bioinformatics *47*, 11.12.1–34.

Robinson, J.T., Thorvaldsdottir, H., Winckler, W., Guttman, M., Lander, E.S., Getz, G., and Mesirov, J.P. (2011). Integrative genomics viewer. Nat. Biotechnol. *29*, 24–26.

Schafer, S., Miao, K., Benson, C.C., Heinig, M., Cook, S.A., and Hubner, N. (2015). Alternative splicing signatures in RNA-seq data: percent spliced in (PSI). Curr. Protoc. Hum. Genet. *87*, 11 16 1–11 16 14.

Shirley, M.D., Ma, Z., Pedersen, B.S., and Wheelan, S.J. (2015). Efficient "pythonic" access to FASTA files using pyfaidx. PeerJ PrePrints. https://doi.org/10.7287/peerj.preprints.970v1.

Siragusa, E., Weese, D., and Reinert, K. (2013). Fast and accurate read mapping with approximate seeds and multiple backtracking. Nucleic Acids Res. *41*, e78.

Szolek, A., Schubert, B., Mohr, C., Sturm, M., Feldhahn, M., and Kohlbacher, O. (2014). OptiType: precision HLA typing from next-generation sequencing data. Bioinformatics *30*, 3310–3316.

Tarasov, A., Vilella, A.J., Cuppen, E., Nijman, I.J., and Prins, P. (2015). Sambamba: fast processing of

NGS alignment formats. Bioinformatics *31*, 2032–2034.

Thorsson, V., Gibbs, D.L., Brown, S.D., Wolf, D., Bortone, D.S., Ou Yang, T.H., Porta-Pardo, E., Gao, G.F., Plaisier, C.L., Eddy, J.A., et al. (2018). The Immune Landscape of Cancer. Immunity *48*, 812–830 e14.

Vita, R., Mahajan, S., Overton, J.A., Dhanda, S.K., Martini, S., Cantrell, J.R., Wheeler, D.K., Sette, A., and Peters, B. (2019). The Immune Epitope Database (IEDB): 2018 update. Nucleic Acids Res. *47*, D339–D343.

Wang, T.Y., Liu, Q., Ren, Y., Alam, S.K., Wang, L., Zhu, Z., Hoeppner, L.H., Dehm, S.M., Cao, Q., and Yang, R. (2021). A pan-cancer transcriptome analysis of exitron splicing identifies novel cancer driver genes and neoepitopes. Mol. Cell *81*, 2246–2260 e12.

Wang, T.Y., Wang, L., Alam, S.K., Hoeppner, L.H., and Yang, R. (2019). ScanNeo: identifying indel-derived neoantigens using RNA-Seq data. Bioinformatics *35*, 4159–4161.

Yang, R., Van Etten, J.L., and Dehm, S.M. (2018). Indel detection from DNA and RNA sequencing data with transIndel. BMC Genomics *19*, 270.