Taller: Análisis bioinformático de SARS-CoV-2 1-3 septiembre 2021

Tutorial
Anotación Coronavirus usando VADR v1.3

Tutor: Adrián C. Rodriguez Ararat
Asistente de investigación grupo Natura, Universidad Icesi

Basado en el desarrollo de Eric Nawrocki

Cómo anotar secuencias de SARS-CoV-2 con VADR v1.3 o una versión posterior

1. Descargue e instale la última versión de VADR, siguiendo las instrucciones de este enlace (https://github.com/ncbi/vadr/blob/master/documentation/install.md)

Crear un directorio de trabajo
`mkdir /home/su_usuario/vard_sars_cov_2`

Descargar el ejecutable del instalador
`wget https://raw.githubusercontent.com/ncbi/vadr/master/vadr-install.sh`

2. Descargue los últimos modelos vadr de SARS-CoV-2 (versión 1.3-1, tarball gzipped) desde (https://ftp.ncbi.nlm.nih.gov/pub/nawrocki/vadr-models/sarscov2/1.3-1/vadr-models-sarscov2-1.3-1.tar.gz), descomprímalos (por ejemplo, tar xfz <tarball.gz>). Tenga en cuenta la ruta del nombre del directorio creado (<**sarscov2-models-dir-path**>) para el paso 3.

3. Elimine los nucleótidos ambiguos terminales de su archivo de secuencia fasta de entrada utilizando el `script fasta-trim-terminal-ambigs.pl` en **$VADRSCRIPTSDIR/miniscripts/**. La ruta de procesamiento del SARS-CoV-2 de GenBank elimina los nucleótidos ambiguos del principio y del final de las secuencias y también las secuencias que tienen menos de 50nt o más de 30.000nt (después del *trimming*) antes de ejecutar VADR, así que para asegurarse de que los resultados de su VADR local son coherentes con los resultados de VADR de GenBank, debería recortar primero los nucleótidos ambiguos terminales.

   ADVERTENCIA: el script `fasta-trim-terminal-ambigs.pl` no reproducirá exactamente el recorte que hace el pipeline de GenBank en algunos casos raros, pero debería solucionar la gran mayoría de las discrepancias que pueda ver entre los resultados locales de VADR y los de GenBank.

4. Para eliminar los nucleótidos ambiguos terminales de su archivo de secuencia <**input-fasta-file**> y para eliminar las secuencias cortas y largas para crear un nuevo archivo recortado <trimmed-fasta-file>, ejecute:
   `$VADRSCRIPTSDIR/miniscripts/fasta-trim-terminal-ambigs.pl --minlen 50 --maxlen 30000 <input-fasta-file> > <trimmed-fasta-file>`

5. Ejecute el programa `v-annotate.pl` en un archivo fasta de entrada recortado con secuencias de SARS-CoV-2 utilizando el comando y las opciones recomendadas a continuación.

   NOTA: Las opciones de abajo han cambiado para vadr 1.3. El siguiente comando se ejecuta con múltiples hilos en hasta 8 CPUs, por lo que sólo se recomienda si tiene al menos 8 CPUs y 16Gb de RAM disponibles. Para ejecutar en <n> CPUs, sustituya **--cpu 8** por **--cpu <n>**. Para ejecutar un solo hilo en una sola CPU, elimine la opción --cpu 8. Las opciones --split y --cpu son incompatibles con -p.

v-annotate.pl --split --cpu 8 --glsearch -s -r --nomisc --mkey sarscov2 --lowsim5seq 6 --lowsim3seq 6 --alt_fail lowscore,insertnn,deletinn --mdir <sarscov2-models-dir-path> <fasta-file-to-annotate> <output-directory-to-create>

(MÁS RELEVANTE PARA LOS USUARIOS AVANZADOS) OPCIONALMENTE mapee las coordenadas del modelo que no son NC_045512 en el archivo de salida. alt.list a coordenadas NC_045512 usando el script vadr-map-model-coords.pl en **$VADRSCRIPTSDIR/miniscripts/**. El archivo alt.list de salida incluye información sobre todas las alertas fatales que hacen que las secuencias fallen junto con las coordenadas del modelo relevantes para esas alertas. Algunos usuarios pueden estar interesados en convertir todos los datos vinculados a coordenadas del modelo NC_045512 a coordenadas para ayudar en el análisis posterior. Para añadir un campo adicional delimitado por tabulaciones con coordenadas NC_045512 para cada alerta, ejecute:

$VADRSCRIPTSDIR/miniscripts/vadr-map-model-coords.pl <output-directory>/<output-alt-list-file> <sarscov2-model-dir-path>/sarscov2.mmap NC_045512

Modelos VADR de SARS-CoV-2

Desde el 13 de abril de 2021, la biblioteca de modelos VADR utilizada por GenBank para la anotación del SARS-CoV-2 (biblioteca de modelos vadr-models-sarscov2-1.3-1) incluye cuatro modelos de SARS-CoV-2: NC_045512, NC_045512-del28254, NC045512-MW422255 (B.1.1.7) y NC_045512-MW809059 (B.1.525). Puede determinar qué modelo se utilizó para anotar cualquier secuencia de entrada en los archivos de salida.sqa vadr descritos más adelante.

Modelo NC_045512: basado en la secuencia RefSeq NC_045512.2, tiene una longitud de 29903 nt.

Modelo NC_045512-del28254: idéntico al modelo NC_045512, salvo una deleción de único nucleótido en la posición 28254. Esta supresión de un solo nucleótido afecta al codón de parada del CDS de ORF8 en relación con el NC_045512 RefSeq, ampliando la longitud de ORF8 en cuatro aminoácidos en el modelo NC_045512-del28254 en relación con el modelo NC_045512 RefSeq. La longitud es de 29902 nt.

Modelo NC_045512-MW422255: destinado a facilitar la presentación de secuencias del linaje B.1.1.7. La longitud de este modelo es de 29884 nt. Se basa en la secuencia MW422255.1 pero modificada como sigue:

- Amplia la secuencia MW422255 en 54 nt en el extremo 5' y 67 nt en el extremo 3' para que los extremos 5' y 3' coincidan con la secuencia RefSeq NC_045512, utilizando los nucleótidos de NC_045512
- replacing the 8 N nucleotides with the corresponding nucleotide from NC_045512.
- sustituyendo los 8 nucleótidos N por el correspondiente nucleótido de NC_045512

Modelo NC_045512-MW809059: destinado a facilitar la presentación de secuencias del linaje actualmente denominado B.1.525. La longitud de este modelo es de 29830 nt. Se basa en la secuencia MW809059.1 pero modificada como sigue:

- ampliar la secuencia MW809059 en 2 nt en el extremo 5' y 50 nt en el extremo 3' para que los extremos 5' y 3' coincidan con la secuencia NC_045512 RefSeq, utilizando los nucleótidos de NC_045512

Anotación de secuencias de SARS-CoV-2

Esta sección es un ejemplo de anotación en v-annotate.pl de secuencias de SARS-CoV-2 de GenBank utilizando el mismo comando y opciones que GenBank utiliza actualmente para examinar las secuencias de SARS-CoV-2 entrantes.
Para descargar archivo fasta de tres secuencias
$wget -v https://ftp.ncbi.nlm.nih.gov/pub/nawrocki/vadr-models/sarscov2/pretrim.sars-cov2.4.fa

Directorio del modelo de SARS-CoV-2
/usr/local/vadr-models-sarscov2-1.3-1

pretrim.sars-cov2.4.fa

Crearemos un archivo con las secuencias depuradas sars-cov2.4.fa
$VADRSCRIPTSDIR/miniscripts/fasta-trim-terminal-ambigs.pl --minlen 50 --maxlen 30000 pretrim.sars-cov2.4.fa > sars-cov2.4.fa

Anotar las secuencias recortadas utilizando las opciones recomendadas de v-annotate.pl para el SARS-CoV-2, ejecute el siguiente comando y creara un directorio llamado my4
v-annotate.pl --split --cpu 8 --glsearch -s -r --nomisc --mkey sarscov2 --lowsim5seq 6 --lowsim3seq 6 --alt_fail lowscore,insertnn,deletinn --mdir /usr/local/vadr-models-sarscov2-1.3-1 sars-cov2.4.fa my4

Cuando ejecute el comando anterior, debería ver una salida similar al siguiente bloque que enumera los valores de las variables de entorno relevantes, y los argumentos y opciones de entrada:

```
# v-annotate.pl :: classify and annotate sequences using a model library
# VADR 1.3 (Aug 2021)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# date:             Tue Aug  3 09:49:47 2021
# $VADRBIOEASELDIR:  /usr/local/vadr-install-1.3/Bio-Easel
# $VADRBLASTDIR:     /usr/local/vadr-install-1.3/ncbi-blast/bin
# $VADREASELDIR:     /usr/local/vadr-install-1.3/infernal/binaries
# $VADRINFERNALDIR:  /usr/local/vadr-install-1.3/infernal/binaries
# $VADRMODELDIR:     /usr/local/vadr-install-1.3/vadr-models-calici
# $VADRSCRIPTSDIR:   /usr/local/vadr-install-1.3/vadr
#
# sequence file:                                         sars-cov2.4.fa
# output directory:                                       my4
# specify that alert codes in <s> cause FAILure:                        lowscore,insertnn,deletinn [--alt_fail]
# .cm, .minfo, blastn .fa files in $VADRMODELDIR start with key <s>, not 'vadr':  sarscov2 [--mkey]
# model files are in directory <s>, not in $VADRMODELDIR:                  /usr/local/vadr-models-
sarscov2-1.3-1 [--mdir]
# in feature table for failed seqs, never change feature type to misc_feature:    yes [--nomisc]
# lowsim5s/LOW_SIMILARITY_START minimum length is <n>:                      6 [--lowsim5seq]
# lowsim3s/LOW_SIMILARITY_END minimum length is <n>:                       6 [--lowsim3seq]
# use max length ungapped region from blastn to seed the alignment:          yes [-s]
# replace stretches of Ns with expected nts, where possible:                yes [-r]
# split input file into chunks, run each chunk separately:                  yes [--split]
# parallelize across <n> CPU workers (requires --split or --glsearch):          8 [--cpu]
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

A continuación, v-annotate.pl emitirá información a medida que avanza por los diferentes pasos del análisis:

```
# Validating input                                                    ... done. [    0.3 seconds]
# Splitting sequence file into chunks to run independently in parallel on 8 processors    ... done. [    0.9
seconds]
# Executing 4 scripts in parallel on 8 processors to process 4 partition(s) of all 4 sequence(s) ...
#             4 of    4 jobs finished (0.1 minutes spent waiting)
# done. [   12.4 seconds]
# Merging and finalizing output                                       ... done. [    0.8 seconds]
```

Con las opciones --split --cpu 8, el script fasta de entrada se fragmenta, en este caso colocando una secuencia por fragmento pero normalmente unas 10 secuencias por fragmento para archivos más grandes, y ejecuta v-annotate.pl por separado en esos trozos en 8 CPUs diferentes en paralelo. Cuando todas las secuencias terminan de procesarse, el script principal fusiona la salida.
La salida de v-annotate.pl incluye un resumen de la clasificación de las secuencias y las alertas reportadas:

```
# Summary of classified sequences:
#
#                                  num   num   num
#idx  model            group        subgroup    seqs  pass  fail
#---  ------------------  ------------  ----------  ----  ----  ----
1    NC_045512          Sarbecovirus  SARS-CoV-2     3     2     1
```

```
2    NC_045512-MW422255  Sarbecovirus  SARS-CoV-2    1    0    1
#--- ------------------ ------------ ---------- ---- ---- ----
-    *all*          -        -         4    2    2
-    *none*         -        -         0    0    0
#--- ------------------ ------------ ---------- ---- ---- ----
#
# Summary of reported alerts:
#
#    alert    causes   short                     per    num   num  long
#idx  code     failure  description               type   cases  seqs  description
#--- -------- ------- -------------------------- ------- ----- ---- -----------
1    cdsstopn  yes*    CDS_HAS_STOP_CODON         feature    3     2  in-frame stop codon exists 5' of stop position predicted by homology to reference
2    cdsstopp  yes*    CDS_HAS_STOP_CODON         feature    3     2  stop codon in protein-based alignment
3    peptrans  yes*    PEPTIDE_TRANSLATION_PROBLEM  feature    26    1  mat_peptide may not be translated because its parent CDS has a problem
#--- -------- ------- -------------------------- ------- ----- ---- -----------
```

Y finalmente una lista de los archivos de salida creados:

```
# Output printed to screen saved in:
my4.vadr.log
# List of executed commands saved in:
my4.vadr.cmd
# List and description of all output files saved in:
my4.vadr.filelist
# esl-seqstat -a output for input fasta file saved in:
my4.vadr.seqstat
# 5 column feature table output for passing sequences saved in:
my4.vadr.pass.tbl
# 5 column feature table output for failing sequences saved in:
my4.vadr.fail.tbl
# list of passing sequences saved in:
my4.vadr.pass.list
# list of failing sequences saved in:
my4.vadr.fail.list
# list of alerts in the feature tables saved in:
my4.vadr.alt.list
# fasta file with passing sequences saved in:
my4.vadr.pass.fa
# fasta file with failing sequences saved in:
my4.vadr.fail.fa
# per-sequence tabular annotation summary file saved in:
my4.vadr.sqa
# per-sequence tabular classification summary file saved in:
my4.vadr.sqc
# per-feature tabular summary file saved in:
my4.vadr.ftr
# per-model-segment tabular summary file saved in:
my4.vadr.sgm
# per-alert tabular summary file saved in:
my4.vadr.alt
# alert count tabular summary file saved in:
my4.vadr.alc
# per-model tabular summary file saved in:
my4.vadr.mdl
# alignment doctoring tabular summary file saved in:
my4.vadr.dcr
# ungapped seed alignment summary file (-s) saved in:
my4.vadr.sda
```

```
# replaced stretches of Ns summary file (-r) saved in:
my4.vadr.rpn
#
# All output files created in directory ./my4/
#
# Elapsed time:  00:00:13.71
#                hh:mm:ss
#
[ok]
```

Tenga en cuenta que todos los archivos de salida estarán en el directorio recién creado my4. El resumen de las secuencias clasificadas muestra que dos secuencias pasaron y dos fallaron. El archivo my4.vadr.pass.list, lista las dos secuencias que pasaron:

```
MT159720.1
MT308693.1
```

y my4.vadr.fail.list enumera las dos secuencias que fallaron:

```
MT159720.1/1406-G-to-T
MW422255.1/21610-T-to-A
```

Además, los archivos de secuencia con formato FASTA para cada una de las secuencias que pasan y fallan son my4.vadr.pass.fa y my4.vadr.fail.fa.

Para las dos secuencias que pasaron, la anotación está disponible en el archivo de salida my4.vadr.pass.tbl y para las dos secuencias que fallaron la anotación está en el archivo my4.vadr.fail.tbl.

Archivo mi.vadr.pass.tbl: (con la mitad de la tabla para cada secuencia eliminada por razones de brevedad)

```
>Feature MT159720.1
266     21555   gene
                        gene    ORF1ab
266     13468   CDS
13468   21555
                        product ORF1ab polyprotein
                        exception       ribosomal slippage
                        protein_id      MT159720.1_1
266     13483   CDS
                        product ORF1a polyprotein
                        protein_id      MT159720.1_2
266     805     mat_peptide
                        product leader protein
                        protein_id      MT159720.1_1

...snip...

28274   29533   gene
                        gene    N
28274   29533   CDS
                        product nucleocapsid phosphoprotein
                        protein_id      MT159720.1_11
29558   29674   gene
                        gene    ORF10
29558   29674   CDS
                        product ORF10 protein
                        protein_id      MT159720.1_12
29609   29644   stem_loop
                        note    Coronavirus 3' UTR pseudoknot stem-loop 1
29629   29657   stem_loop
                        note    Coronavirus 3' UTR pseudoknot stem-loop 2
29728   29768   stem_loop
                        note    Coronavirus 3' stem-loop II-like motif (s2m)
>Feature MT308693.1
217     21506   gene
                        gene    ORF1ab
```

```
217     13419   CDS
13419   21506
                        product  ORF1ab polyprotein
                        exception        ribosomal slippage
                        protein_id       MT308693.1_1
217     13434   CDS
                        product  ORF1a polyprotein
                        protein_id       MT308693.1_2
217     756     mat_peptide
                        product  leader protein
                        protein_id       MT308693.1_1

...snip...

28225   29484   gene
                        gene     N
28225   29484   CDS
                        product  nucleocapsid phosphoprotein
                        protein_id       MT308693.1_11
29509   29625   gene
                        gene     ORF10
29509   29625   CDS
                        product  ORF10 protein
                        protein_id       MT308693.1_12
29560   29595   stem_loop
                        note     Coronavirus 3' UTR pseudoknot stem-loop 1
29580   29608   stem_loop
                        note     Coronavirus 3' UTR pseudoknot stem-loop 2
29679   29719   stem_loop
                        note     Coronavirus 3' stem-loop II-like motif (s2m)
```

Y la segunda secuencia en el mi.vadr.fail.tbl:

```
>Feature MW422255.1/21610-T-to-A
212     21492   gene
                        gene     ORF1ab
212     13405   CDS
13405   21492
                        product  ORF1ab polyprotein
                        exception        ribosomal slippage
                        protein_id       MW422255.1/21610-T-to-A_1
212     13420   CDS
                        product  ORF1a polyprotein
                        protein_id       MW422255.1/21610-T-to-A_2
212     751     mat_peptide
                        product  leader protein
                        protein_id       MW422255.1/21610-T-to-A_1

...snip...

29556   29584   stem_loop
                        note     Coronavirus 3' UTR pseudoknot stem-loop 2
29655   29695   stem_loop
                        note     Coronavirus 3' stem-loop II-like motif (s2m)

Additional note(s) to submitter:
ERROR: CDS_HAS_STOP_CODON: (CDS:surface glycoprotein) in-frame stop codon exists 5' of stop
position predicted by homology to reference [TAA, shifted S:3702,M:3702]; seq-coords:21608..21610:+;
mdl-coords:21662..21664:+; mdl:NC_045512-MW422255;
ERROR: CDS_HAS_STOP_CODON: (CDS:surface glycoprotein) stop codon in protein-based alignment
[-]; seq-coords:21608..21610:+; mdl-coords:21662..21664:+; mdl:NC_045512-MW422255;
```

Opción de convertir las coordenadas del modelo que no son NC_045512 en coordenadas NC_045512 en los archivos de salida .alt.list y GenBank detailed error report .tsv

VADR utiliza cuatro modelos diferentes de SARS-CoV-2, e informa de qué modelo se utiliza para anotar cada secuencia de entrada en varios archivos de salida, incluido el archivo .alt.list y el archivo "detailed error report .tsv" creado por el portal de envío de GenBank para cualquier envío que incluya al menos un fallo (para el que el usuario haya seleccionado no eliminar automáticamente las secuencias que fallen). Los archivos alt.list y .tsv incluyen información sobre cada alerta o error fatal.

Ejemplo de resultado de my4/my4.vadr.alt.list

```
#sequence model feature-type feature-name error seq-coords mdl-coords error-description
MT159720.1/1406-G-to-T NC_045512 CDS ORF1ab polyprotein
CDS_HAS_STOP_CODON 1406..1408:+ 1406..1408:+ in-frame stop codon exists 5' of stop position predicted by homology to reference [TAG, shifted S:20147,M:20147]
MT159720.1/1406-G-to-T NC_045512 CDS ORF1ab polyprotein
CDS_HAS_STOP_CODON 1406..1408:+ 1406..1408:+ stop codon in protein-based alignment [-]
MT159720.1/1406-G-to-T NC_045512 CDS ORF1a polyprotein
CDS_HAS_STOP_CODON 1406..1408:+ 1406..1408:+ in-frame stop codon exists 5' of stop position predicted by homology to reference [TAG, shifted S:12075,M:12075]

...snip...

MT159720.1/1406-G-to-T NC_045512 mat_peptide 2'-O-ribose methyltransferase
PEPTIDE_TRANSLATION_PROBLEM - - mat_peptide may not be translated because its parent CDS has a problem [-]
MT159720.1/1406-G-to-T NC_045512 mat_peptide nsp11
PEPTIDE_TRANSLATION_PROBLEM - - mat_peptide may not be translated because its parent CDS has a problem [-]
MW422255.1/21610-T-to-A NC_045512-MW422255 CDS surface glycoprotein
CDS_HAS_STOP_CODON 21608..21610:+ 21662..21664:+ in-frame stop codon exists 5' of stop position predicted by homology to reference [TAA, shifted S:3702,M:3702]
MW422255.1/21610-T-to-A NC_045512-MW422255 CDS surface glycoprotein
CDS_HAS_STOP_CODON 21608..21610:+ 21662..21664:+ stop codon in protein-based alignment [-]
```

Puede ejecutar el siguiente script para añadir un campo adicional delimitado por tabulaciones al final de cada línea con las coordenadas del modelo NC_045512 para cada alerta con el siguiente comando:

```
perl $GDIR/vadr/miniscripts/vadr-map-model-coords.pl my4/my4.vadr.alt.list <sarscov2-models-dir-path>/sarscov2.mmap NC_045512
```

Referencia

Alejandro A Schäffer, Eneida L Hatcher, Linda Yankie, Lara Shonkwiler, J Rodney Brister, Ilene Karsch-Mizrachi, Eric P Nawrocki; VADR: validation and annotation of virus sequence submissions to GenBank. BMC Bioinformatics 21, 211 (2020). https://doi.org/10.1186/s12859-020-3537-3