

Version 2 ▾

Sep 12, 2022

Phylogenetic Analysis of Complete Bovine Coronavirus Genome Sequences V.2

Aspen M Workman¹, Tara G. McDaneld¹, Gregory P P Harhay¹, John Dustin Loy², Benjamin M. Hause³, Subha das³

¹United States Department of Agriculture (USDA) Agricultural Research Service (ARS), US Meat Animal Research Center (U-SMARC), State Spur 18D, Clay Center, NE 68933, USA;

²Nebraska Veterinary Diagnostic Center, School of Veterinary Medicine and Biomedical Sciences, University of Nebraska-Lincoln, 4040 East Campus Loop N, Lincoln, NE 68503-0907;

³South Dakota State University, Brookings, SD 57007, USA

1 Works for me

Share

dx.doi.org/10.17504/protocols.io.kqdg3pyeql25/v2

 Gregory P P Harhay
United States Department of Agriculture (USDA) Agricultural ...

DISCLAIMER

- The findings and conclusions in this presentation are those of the authors and should not be construed to represent any official USDA or U.S. Government determination or policy.
- Mention of trade names or commercial products in this presentation is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the USDA.
- USDA is an equal opportunity provider and employer

ABSTRACT

Bovine coronavirus (BCoV) has spilled over to many species, including humans, where the host range variant coronavirus OC43 is endemic. Balance of the opposing activities of the surface spike (S) and hemagglutinin esterase (HE) glycoproteins control virion avidity which is critical for interspecies transmission and host adaptation. Here, 78 genomes were sequenced directly from clinical samples collected between 2013 and 2022 from cattle in 12 states, primarily in the Midwestern U.S. Relatively little genetic diversity was observed, with genomes having >98% nucleotide identity. Eleven isolates collected between 2020 and 2022 from four states (Nebraska, Colorado, California, and Wisconsin) contained a 12-nucleotide insertion in the receptor-binding domain (RBD) of the HE gene identical to one recently reported in China, and a single genome from Nebraska collected in 2020 contained a novel 12-nucleotide deletion in the HE gene RBD. Isogenic HE proteins containing either the insertion or deletion in the HE RBD maintained esterase activity and the ability to bind bovine submaxillary mucin, a substrate enriched in the receptor 9-O-acetylated-sialic acid, despite modeling that predicted structural changes in the HE R3 loop critical for receptor binding. The emergence of BCoV with structural variants in the RBD raises the possibility of further interspecies transmission.

DOI

dx.doi.org/10.17504/protocols.io.kqdg3pyeql25/v2

PROTOCOL CITATION

Aspen M Workman, Tara G. McDaneld, Gregory P P Harhay, John Dustin Loy, Benjamin M. Hause,

Subha das 2022. Phylogenetic Analysis of Complete Bovine Coronavirus Genome Sequences.

protocols.io

<https://protocols.io/view/phylogenetic-analysis-of-complete-bovine-coronavirus-cgcktsuw>

Version created by [Gregory P P Harhay](#)

FUNDERS ACKNOWLEDGEMENT

National Institute of Food and Agriculture

Grant ID: 2019-67015-29847

USDA ARS CRIS

Grant ID: 3040-32000-036-00D

by National Institutes of Allergy and Infectious Diseases

Grant ID: 1R21AI162594-01

MANUSCRIPT CITATION please remember to cite the following publication along with this protocol

Recent emergence of bovine coronavirus variants with mutations in the hemagglutinin-esterase receptor binding domain in U.S. cattle (submitted)

KEYWORDS

coronavirus, hemagglutinin-esterase, spike protein, bovine respiratory disease, calf diarrhea, variant

LICENSE

This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

CREATED

Sep 08, 2022

LAST MODIFIED

Sep 12, 2022

PROTOCOL INTEGER ID

69740

DISCLAIMER:

- The findings and conclusions in this presentation are those of the authors and should not be construed to represent any official USDA or U.S. Government determination or policy.
- Mention of trade names or commercial products in this presentation is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the USDA.
- USDA is an equal opportunity provider and employer

Software

1

MAFFT 7.450 

by Kazutaka Katoh

2

RAXML 8.2.12 [🔗](#)

UBUNTU Linux ` 20.4

[source](#) by Alexandros Stamatakis

3

FAST --- Fast Analysis of Sequences Toolbox 1.7 [🔗](#)

UBUNTU Linux 22 LTS

[source](#) by Travis Lawrence

MAFFT Alignment

4 Align using MAFFT v7.450

The sequences were aligned using the MAFFT accuracy methods ---globalpair as defined in the mafft manual page below

DESCRIPTION

MAFFT is a multiple sequence alignment program for unix-like operating systems. It offers a range of multiple alignment methods.

Accuracy-oriented methods:

- L-INS-i (probably most accurate; recommended for <200 sequences; iterative refinement method incorporating local pairwise alignment information):

mafft --localpair --maxiterate 1000input [> output]

linsi input [> output]

- G-INS-i (suitable for sequences of similar lengths; recommended for <200 sequences; iterative refinement method incorporating global pairwise alignment information):

mafft--globalpair--maxiterate1000input [> output]

ginsi input [> output]

Keep in mind ...

1. The commands below generate files that contain both STDOUT (mafft output to the "terminal") as well as the multifasta alignment file (.afa). The STDOUT is found at the beginning of the file and the alignments follow the STDOUT
2. All .afa_out files were split in a text editor into two separate .out and .afa files for downstream analysis.
3. The input FASTA file headers contain description information; this header information should be stripped out to facilitate readable, compact leaf names/identifiers. Use the following command to strip out the description text.

sed to strip out the description in the header from FASTA

```
sed '/^>/ s/ .*//' input_alignment_multifasta_file.afa >  
output_alignment_multifasta_file.SEQID_Only.afa
```

Strip out description text from multifasta alignment file

UBUNTU Linux

4.1 The input FASTA files are:

[**192_Spike_6_14_22.fasta**](#) [**192_HE_6_14_22.fasta**](#)
[**192_Genomes_Final_6_14_22.fasta**](#)

The spike and HE genes were extracted from the annotated genomes into the file of fasta sequences above. There are duplicated sequences within 192_HE_6_22.fasta as well as within 192_Spike_6_14_22.fasta. The genome sequences within 192_Genomes_Final_6_14_22.fasta are unique.

The spike and HE file of fasta sequence files were deduplicated with **fassort** & **fasuniq** from the FAST Analysis of Sequences Toolbox. The unique sequences present in multiple genomes will be designated with a sequence identifier comprised of a concatenation of the sequence identifiers used in the multiple genomes separated by a ":". For example, the HE fasta sequence id IWT-11:SHG-6:TCG-21:TCG-23:TCG-22 represents the identical HE fasta sequences IWT-11, SHG-6, TCG-21, TCG-23 and TCG-22.

Create file of unique spike fasta sequences

```
fassort -s 192_Spike_6_14_22.fasta | fasuniq --concat >  
unique_Spike_6_14_22.fasta
```

Create a file of unique Spike fasta sequences. The unique sequences present in multiple genomes will be designated with a sequence identifier comprised of a concatenation of the sequence identifiers used in the multiple genomes, separated by a ":".

UBUNTU Linux

 **unique_HE_6_14_22.fasta**

Create file of unique spike fasta sequences

```
fassort -s 192_Spike_6_14_22.fasta | fasuniq --concat >
unique_Spike_6_14_22.fasta
```

Create a file of unique Spike fasta sequences. The unique sequences present in multiple genomes will be designated with a sequence identifier comprised of a concatenation of the sequence identifiers used in the multiple genomes, separated by a ":".

UBUNTU Linux

 **unique_Spike_6_14_22.fasta**

4.2 Remove description text (all text after sequence id) in HE , Spike, and Genomes fasta fileps.

sed to strip out the description in the header from FASTA

```
sed '/^>/ s/ .*//' unique_HE_6_14_22.fasta >
unique_HE_6_14_22.SEQID_Only.fasta
```

UBUNTU Linux

output :  **unique_HE_6_14_22.SEQID_Only.fasta**

sed to strip out the description in the header from FASTA

```
sed '/^>/ s/ .*//' unique_HE_6_14_22.fasta >
unique_HE_6_14_22.SEQID_Only.fasta
```

UBUNTU Linux

output:  **unique_Spike_6_14_22.SEQID_Only.fasta**

sed to strip out the description in the header from FASTA

```
sed '/^>/ s/ .*//' 192_Genomes_Final_6_14_22.fasta >
192_Genomes_Final_6_14_22.SEQID_Only.fasta
```

UBUNTU Linux

output: 192_Genomes_Final_6_14_22.SEQID_Only.fasta

- 4.3 192_Genomes_Final_6_14_22.SEQID_Only.jtt_200.globalpair.afa_out Align 192 genomes using the default PAM (JTT) 200 substitution matrix

```
nohup mafft --thread 25 --globalpair --maxiterate 1000 --jtt 200 --reorder  
192_Genomes_Final_6_14_22.SEQID_Only.fasta >  
192_Genomes_Final_6_14_22.SEQID_Only.jtt_200.globalpair.afa_out &
```

outputs : 192_Genomes_Final_6_14_22.jtt_200.globalpair.SEQID_Only.afa_out

This was split into 192_Genomes_Final_6_14_22.jtt_200.globalpair.SEQID_Only.afa & 192_Genomes_Final_6_14_22.jtt_200.globalpair.SEQID_Only.out

- 4.4 Align 192 BCV genomes using the PAM (JTT) 100 substitution matrix

```
nohup mafft --thread 30 --globalpair --maxiterate 1000 --jtt 100 --reorder  
192_Genomes_Final_6_14_22.SEQID_Only.fasta >  
192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.afa_out &
```

outputs : 192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.afa_out

This was split into 192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.afa & 192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.out

- 4.5 Align SPIKE using the default PAM (JTT) 200 substitution matrix

```
nohup mafft --thread 15 --globalpair --maxiterate 1000 --jtt 200 --reorder  
unique_Spike_6_14_22.SEQID_Only.fasta >  
unique_Spike_6_14_22.SEQID_Only.jtt_200.globalpair.afa_out &
```

Outputs : unique_Spike_6_14_22.SEQID_Only.jtt_200.globalpair.afa_out

This was split into unique_Spike_6_14_22.SEQID_Only.jtt_200.globalpair.afa & unique_Spike_6_14_22.SEQID_Only.jtt_200.globalpair.out

4.6 Align SPIKE using the PAM (JTT) 100 substitution matrix

```
nohup mafft --thread 15 --globalpair --maxiterate 1000 --jtt 100 --reorder  
unique_Spike_6_14_22.SEQID_Only.fasta >  
unique_Spike_6_14_22.SEQID_Only.jtt_100.globalpair.afa_out &
```

Outputs: [unique_Spike_6_14_22.SEQID_Only.jtt_100.globalpair.afa_out](#)

This was split into [unique_Spike_6_14_22.SEQID_Only.jtt_100.globalpair.afa](#) &
 [unique_Spike_6_14_22.SEQID_Only.jtt_100.globalpair.out](#)

4.7 Align HE using the default PAM (JTT) 200 substitution matrix

```
nohup mafft --thread 15 --globalpair --maxiterate 1000 --jtt 200 --reorder  
unique_HE_6_14_22.SEQID_Only.fasta >  
unique_HE_6_14_22.SEQID_Only.jtt_200.globalpair.afa_out
```

Outputs: [unique_HE_6_14_22.SEQID_Only.jtt_200.globalpair.afa_out](#)

This was split into [unique_HE_6_14_22.SEQID_Only.jtt_200.globalpair.afa](#) &
 [unique_HE_6_14_22.SEQID_Only.jtt_200.globalpair.out](#)

4.8 Align HE using the default PAM (JTT) 100 substitution matrix

```
nohup mafft --thread 15 --globalpair --maxiterate 1000 --jtt 100 --reorder  
unique_HE_6_14_22.SEQID_Only.fasta >  
unique_HE_6_14_22.SEQID_Only.jtt_100.globalpair.afa_out
```

Output: [unique_HE_6_14_22.SEQID_Only.jtt_100.globalpair.afa_out](#)

This was split into: [unique_HE_6_14_22.SEQID_Only.jtt_100.globalpair.afa](#) &
 [unique_HE_6_14_22.SEQID_Only.jtt_100.globalpair.out](#)

5 RaXML Tee Building

Notes below are cut from the [RaXML manual](#) to put the command line parameters used in context

"This is a how-to, which describes how RAxML should best be used for a simple real-world biological analysis, given an example alignment named ex_al"

Now, if you want to run a full analysis, i.e., BS and ML search type:

```
raxmlHPC -f a -x 12345 -p 12345 -# 100 m GTRGAMMA -s ex_al -n TEST
```

This will first conduct a BS search and once that is done a search for the best-scoring ML tree. Such a program run will return the bootstrapped trees (RAxML_bootstrap.TEST), the best scoring ML tree(RAxML_bestTree.TEST), and the BS support values drawn on the best-scoring tree as node labels (RAxML_bipartitions.TEST) as well as, more correctly since support values refer to branches as branch labels (RAxML_bipartitionsBranchLabels.TEST).

Finally, note that by increasing the number of BS replicates via -# you will also make the ML thorough since for ML optimization every 5th BS tree is used as a starting point to for ML trees.

raxmlHPC reports 192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.afa has 3438 DNA alignment patterns

Thus, if you run RAxML with 32 instead of 1 thread this does not mean that it will automatically become 32 times faster, it may actually even become slower. As I already mentioned, the parallel efficiency, that is, with how many threads/cores you can still execute it efficiently in parallel depends on the alignment length, or to be more precise on the number of distinct patterns in your alignment.

This number is printed by RAxML to the terminal and into the RAxML_info.runID file" and look like this:

Alignment has 70 distinct alignment patterns

As a rule of thumb **I'd use one core/thread per 500 DNA site patterns**, i.e., if you have less, than it's probably better to just use the sequential version. Single-gene DNA alignments with around 1000 sites can be analyzed with 2 or at most 4 threads. Thus, the more patterns your alignment has, the more threads/cores you can use efficiently.

Given the directions above from the [RaXML manual](#), use 8 threads for raxmlHPC tree building

```
nohup raxmlHPC -T 8 -f a -x 12345 -p 12345 -N 2000 -m GTRGAMMA -s
192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.afa -n
192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.T8.N2000 >
192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.T8.N2000.out &
```

Run raxmlHPC command using nohup on 'nix platform to protect from premature job termination in case of remote connection loss ... and run in background with & at end of command line. Perform a 2000 bootstrap search followed by a search for the best-scoring maximum likelihood tree.

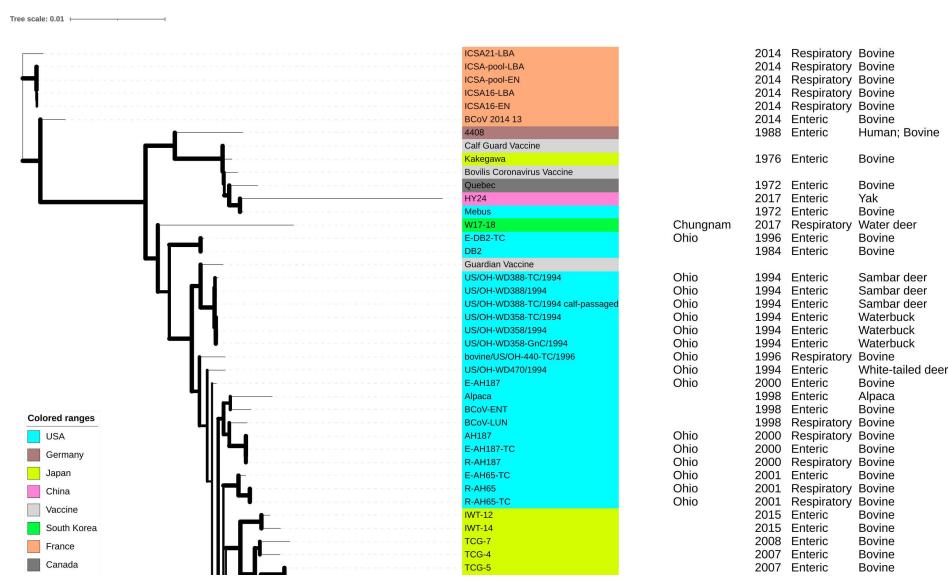
- T 8 (use 8 threads)
- f a
- x 12345
- p 12345
- m GTRGAMMA
- N 2000

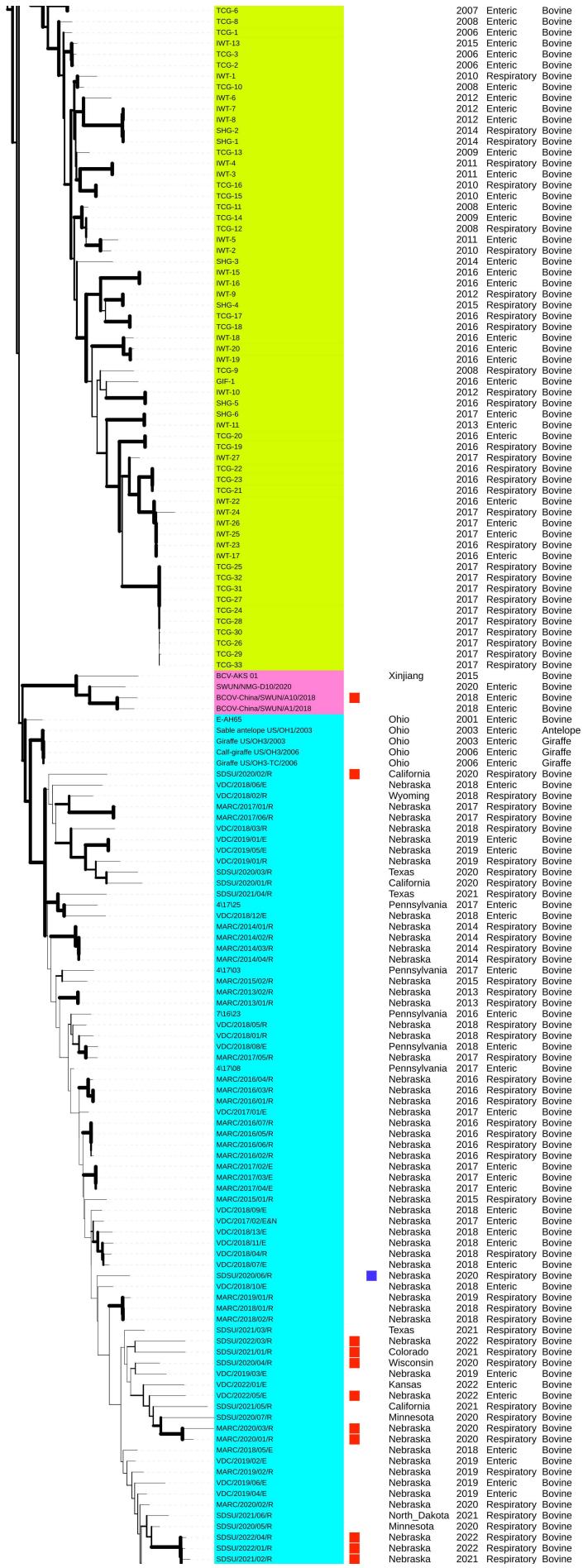
Output :

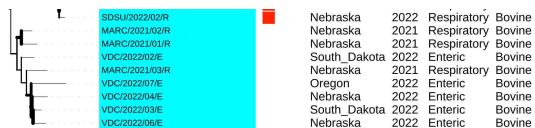
| |
|---|
| RAxML_bootstrap.192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.T8.N2000 |
| 192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.T8.N2000.out |
| RAxML_info.192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.T8.N2000 |
| RAxML_bipartitionsBranchLabels.192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.T8.N2000 |
| RAxML_bipartitions.192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.T8.N2000 |
| RAxML_bestTree.192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.T8.N2000 |

Use RAxML_bipartitionsBranchLabels.192_Genomes_Final_6_14_22.jtt_100.globalpair.SEQID_Only.T8.N2000 file to create phylogenetic tree with bootstrap support values provided as branch labels.

Visualize this tree of [192 Bovine coronavirus genomes, MAFFT Global Pair Alignment, PAM 100](#) in the [Interactive Tree of Life](#)







RAXML tree of 192 bovine coronavirus genomes. Bootstrap support values are proportional to the line thickness of the branches. Isolates with HE deletions are denoted with a filled red square while the single isolate with an HE insertion is denoted with a filled blue square. Tree metrics are best investigated in the [IToL tree](#).

The pdf suitable for download is

[RaXML tree of 192 livestock coronavirus genomes.pdf](#)