Supplemental Table 1. Amino acid differences in swine-origin A(H1N1)v viruses detected in humans compared to A(H1N1)pdm09 vaccine strains.

A.

Amino acid differences in H1N1v compared to nearest CVV (pdm09 6B.1)

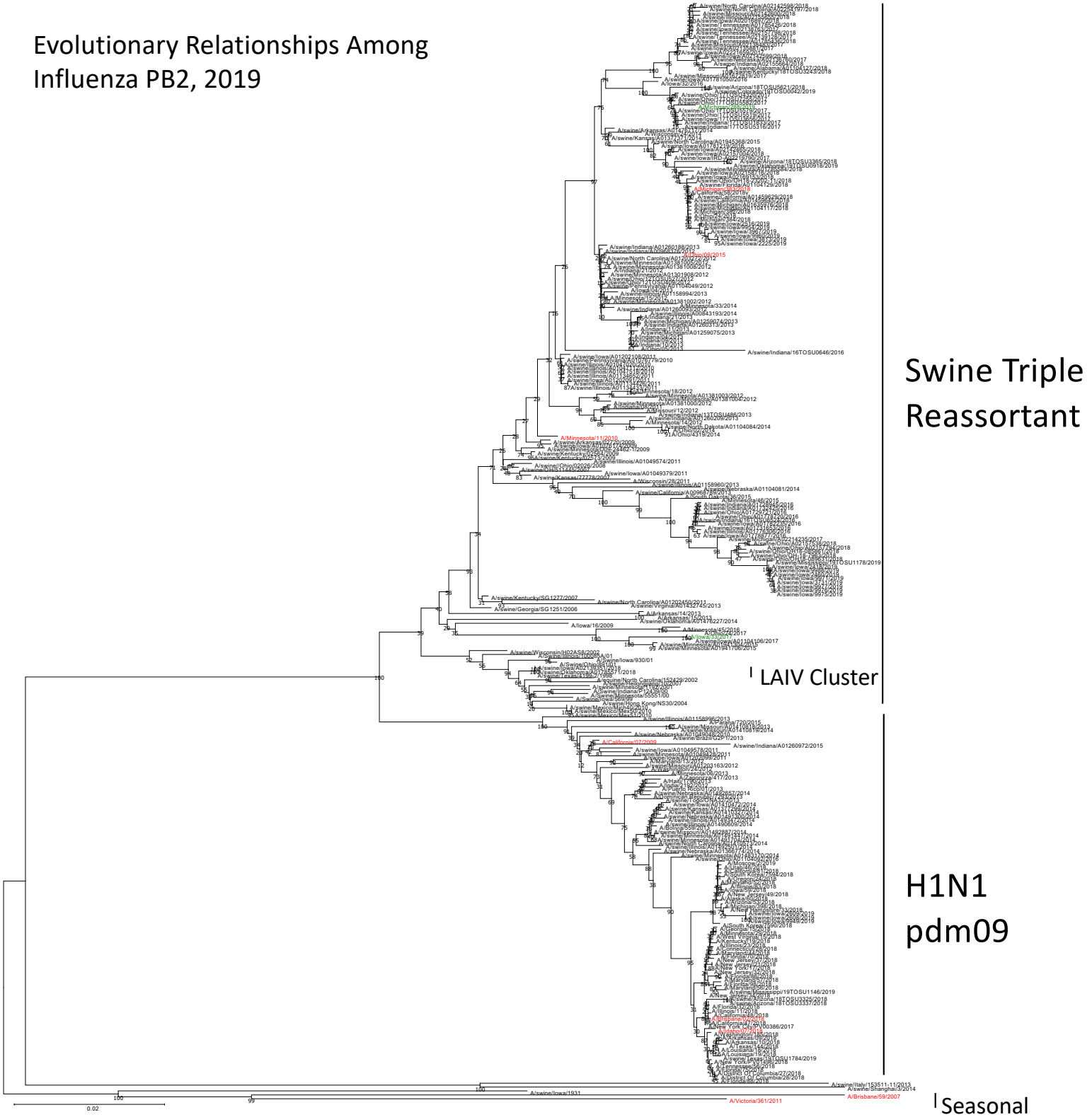
Mature H1 HA	A/Michigan/45/2015	A/Ohio/09/2015	A/swine/Iowa/A01672518/2017	A/swine/Nebraska/A02216645/2017	A/swine/Manitoba/D0455/2016	A/Massachusetts/31/2016	A/Wyoming/24/2016	A/Iowa/33/2017	Annotation
2	T	K							
3	L	I							
35	D	N							
47	V		I	I				I	
48	A							A/T	Antigenic site E
61	I	L							
71	S	A							
84	N	S							Antigenic site E
86	D	N							
113	R	K							
127	D/E	E	D	D	D	D	D	D	Antigenic site A
128	S	T							
141	A	T							Antigenic site A
142	K	N			R				Antigenic site A
155	G	E							Antigenic site B
156	N							I/N	Antigenic site B
161	L	I							
163	Q	I							
166	I	T							
168	D	N							Antigenic site D
169	K	R							Antigenic site D
178	G	A							
183	S	P							Antigenic site B
185	T	S							Antigenic site B
186	A	T							Antigenic site B
191	I/L	L	L	L	L	L	L	L	Antigenic site B
196	D	N		D/H/N/Y					
197	A	S							
203	T	S							
208	K	R							
211	K	E							Antigenic site D
222	D	G							Receptor binding
224	E	A							Receptor binding; Antigenic site D
256	T	A							
257	M	L							
258	E	K							
260	N							K/N	
261	A	S							
269	D	E							
276	N	D							Antigenic site C
283	E	N							
298	I	V							
# diff		38	3	4	3	2	2	6	vs A/Michigan/45/2015_H1N1v

B.

H1N1pdm09 amino acid differences compared to nearest CVV

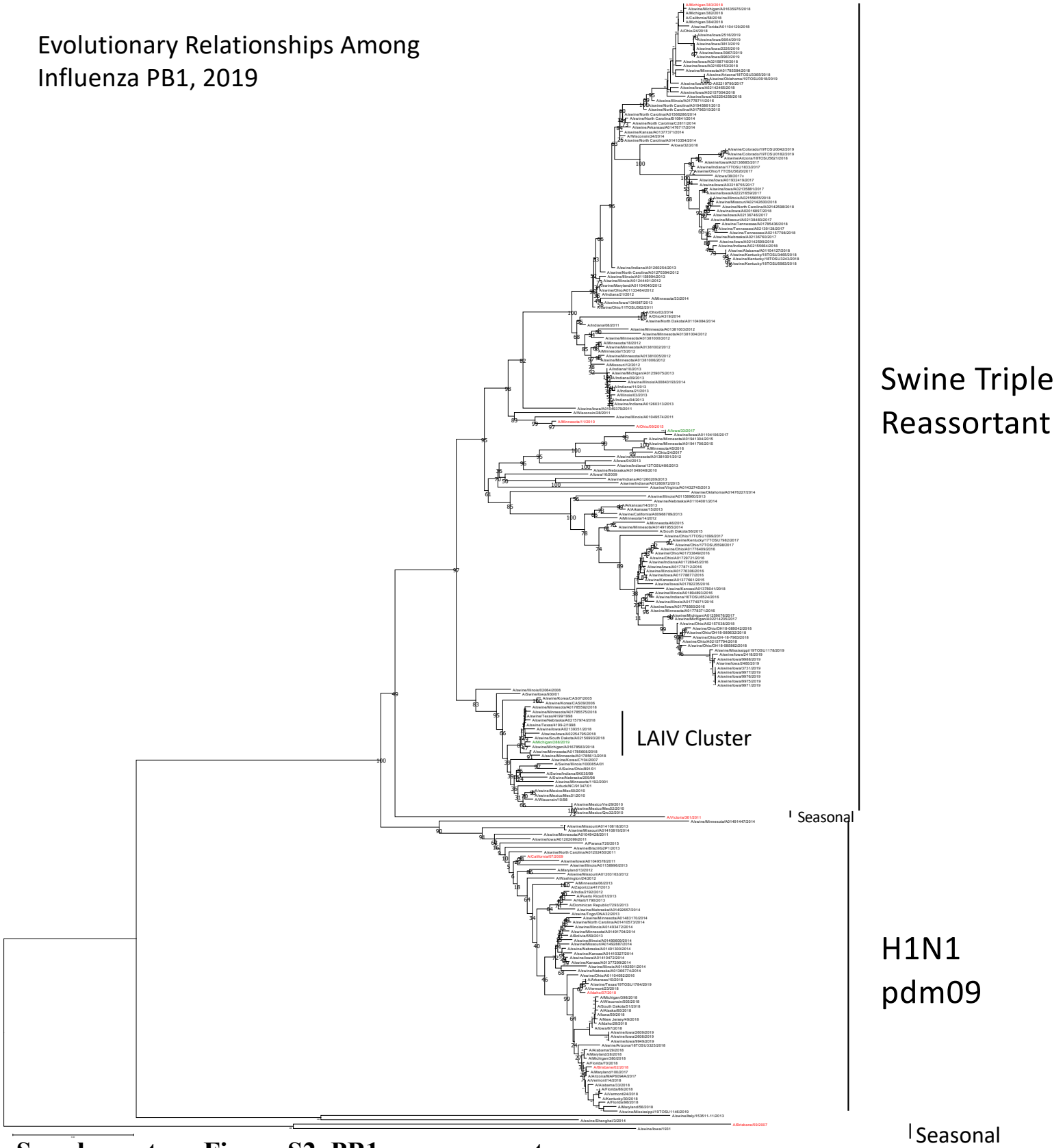
	A/Idaho/07/2018	A/Brisbane/02/2018	A/Michigan/45/2015	A/Michigan/288/2019	A/swine/Texas/19TOSU1784/2019	A/swine/Mississippi/19TOSU1146/2019	A/swine/Arizona/18TOSU3337/2018	A/swine/Arizona/18TOSU3325/2018	A/swine/Iowa/2608/2019	A/swine/Iowa/2609/2019	A/swine/Iowa/9949/2019	A/swine/Iowa/9952/2019	A/swine/Iowa/9955/2019	A/Iowa/33/2017	Annotation
17	D				N										
45	R	G			K		G	G							
47	V													I	
48	A				S									T/A	Antigenic site E
73	A			S											Potential glycosylation
74	R		S											S	
83	S														
84	N														Antigenic site E
97	N														
120	T			A	A										
127	D		E/D												Antigenic site A
129	N								D	D	D	D	D		Antigenic site A
137	P					S									Antigenic site A
142	K			R											Antigenic site A
156	N													I/N	Antigenic site B
162	N														
163	Q														
164	T		S											S	
173	V					I									
183	P		S											S	Antigenic site B
185	T						I	I	I	I	I	I	I		Antigenic site B
191	L		I/L												Antigenic site B; Receptor binding
203	T														
216	T														Antigenic site D
222	D														Receptor binding
223	Q	R													Receptor binding; Antigenic site D
256	T														
260	N								D	D	D	D	D	K/N	
282	P	A					A	A							
283	E														
295	V		I											I	
298	I	V					V	V							
#aadiff		4	6	3	4	2	4	4	3	3	3	3	3	8	vs A/Idaho/07/2018_QMC

A. Differences in the mature HA1 portion of the HA protein identified between A/Iowa/33/2017 and A/Michigan/45/2015, the A(H1N1)pdm09 seasonal influenza vaccine component in the 2017-2018 Northern Hemisphere influenza season in use when this case was identified. **B.** Differences in the mature HA1 portion of the HA protein identified between A/Michigan/288/2019 and A/Idaho/07/2018, the A(H1N1)pdm09 seasonal influenza vaccine component in the 2019-2020 Northern Hemisphere influenza season.



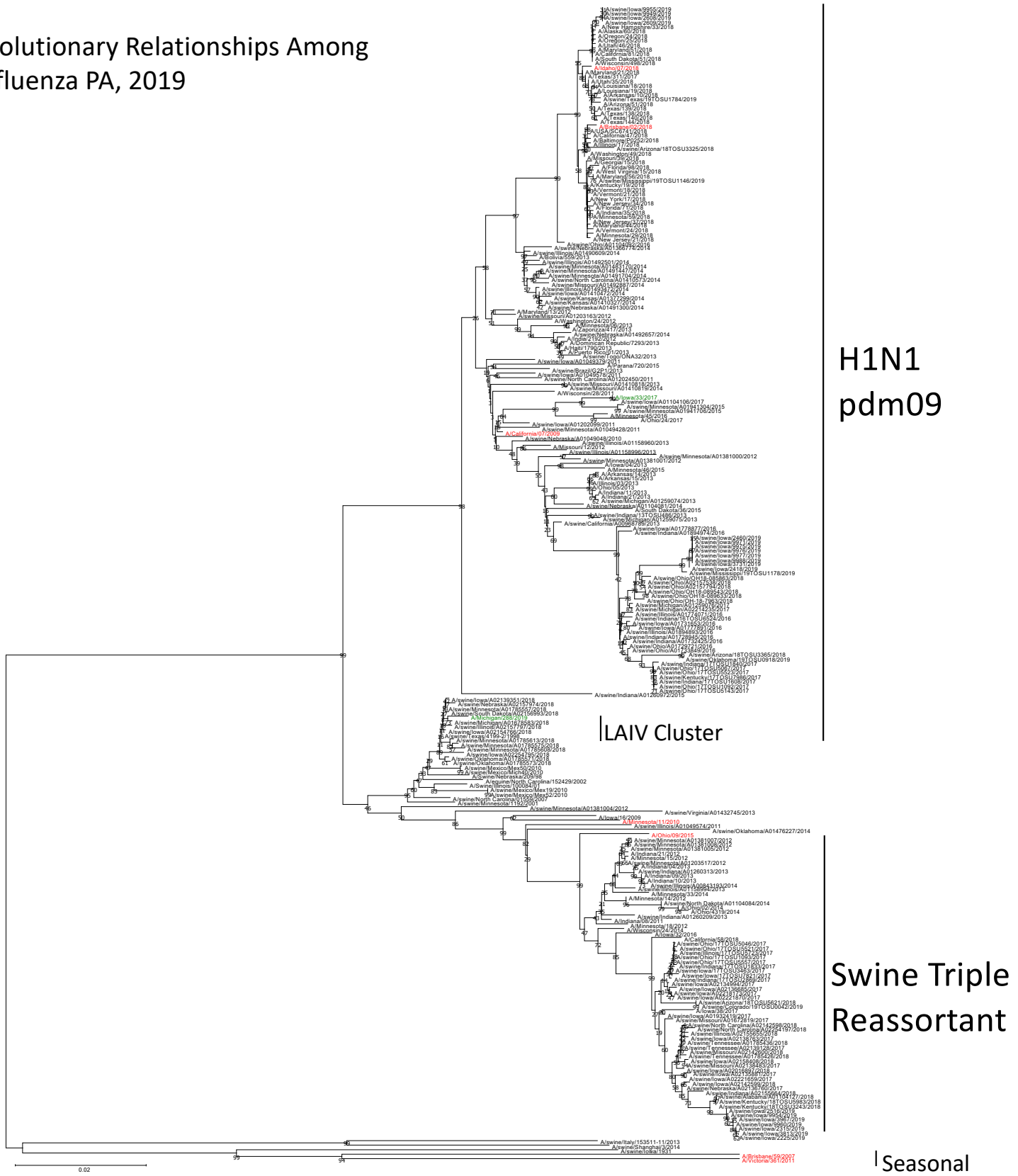
Supplementary Figure S1. PB2 gene segment

The evolutionary history was inferred using the Neighbor-Joining method (26). The optimal tree with the sum of branch length = 1.36953295 is shown. The percentage of replicate trees in which the associated taxa clustered together in the bootstrap test (1000 replicates) are shown next to the branches (27). The tree is drawn to scale, with branch lengths in the same units as those of the evolutionary distances used to infer the phylogenetic tree. The evolutionary distances were computed using the Jukes-Cantor method (28) and are in the units of the number of base substitutions per site. The analysis involved 284 nucleotide sequences. All ambiguous positions were removed for each sequence pair. There were a total of 2335 positions in the final dataset. Evolutionary analyses were conducted in MEGA7 (17).



Supplementary Figure S2. PB1 gene segment

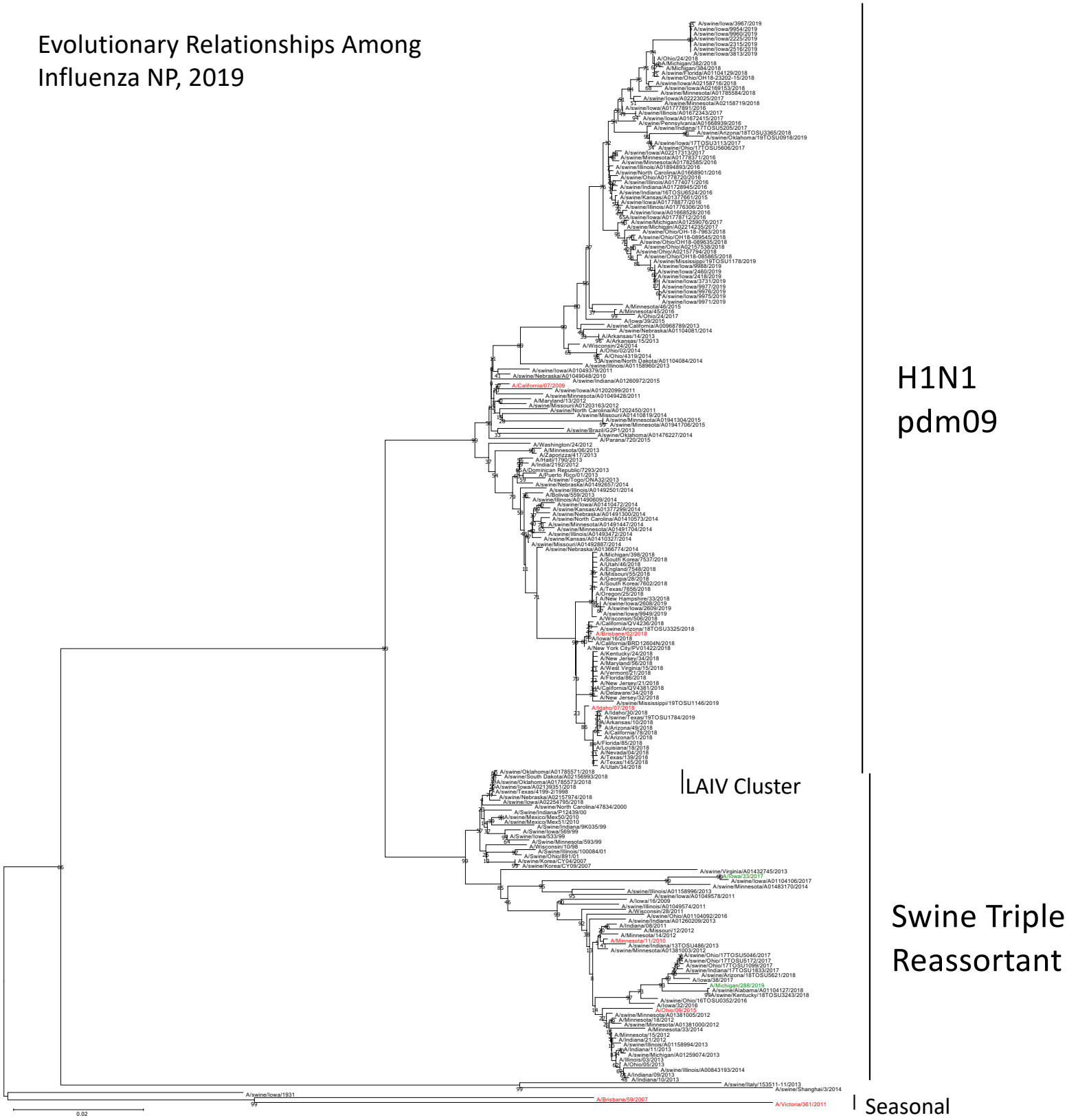
The evolutionary history was inferred using the Neighbor-Joining method (26). The optimal tree with the sum of branch length = 1.30929729 is shown. The percentage of replicate trees in which the associated taxa clustered together in the bootstrap test (1000 replicates) are shown next to the branches (27). The tree is drawn to scale, with branch lengths in the same units as those of the evolutionary distances used to infer the phylogenetic tree. The evolutionary distances were computed using the Jukes-Cantor method (28) and are in the units of the number of base substitutions per site. The analysis involved 257 nucleotide sequences. All ambiguous positions were removed for each sequence pair. There were a total of 2328 positions in the final dataset. Evolutionary analyses were conducted in MEGA7 (17).



Supplementary Figure S3. PA gene segment

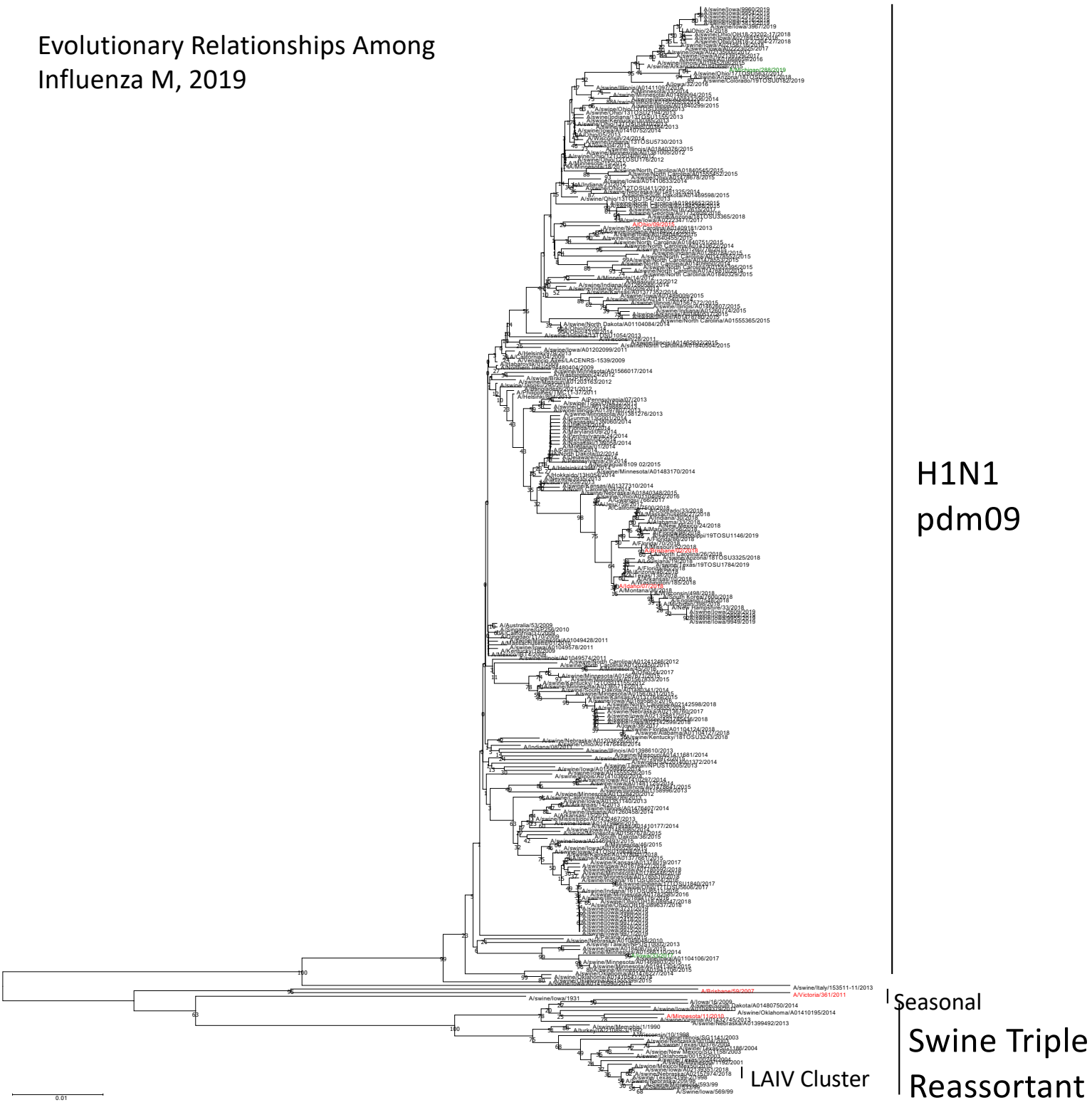
The evolutionary history was inferred using the Neighbor-Joining method (26). The optimal tree with the sum of branch length = 1.23524684 is shown. The percentage of replicate trees in which the associated taxa clustered together in the bootstrap test (1000 replicates) are shown next to the branches (27). The tree is drawn to scale, with branch lengths in the same units as those of the evolutionary distances used to infer the phylogenetic tree. The evolutionary distances were computed using the Jukes-Cantor method (28) and are in the units of the number of base substitutions per site. The analysis involved 256 nucleotide sequences. All ambiguous positions were removed for each sequence pair. There were a total of 2209 positions in the final dataset. Evolutionary analyses were conducted in MEGA7 (17).

Evolutionary Relationships Among
Influenza NP, 2019



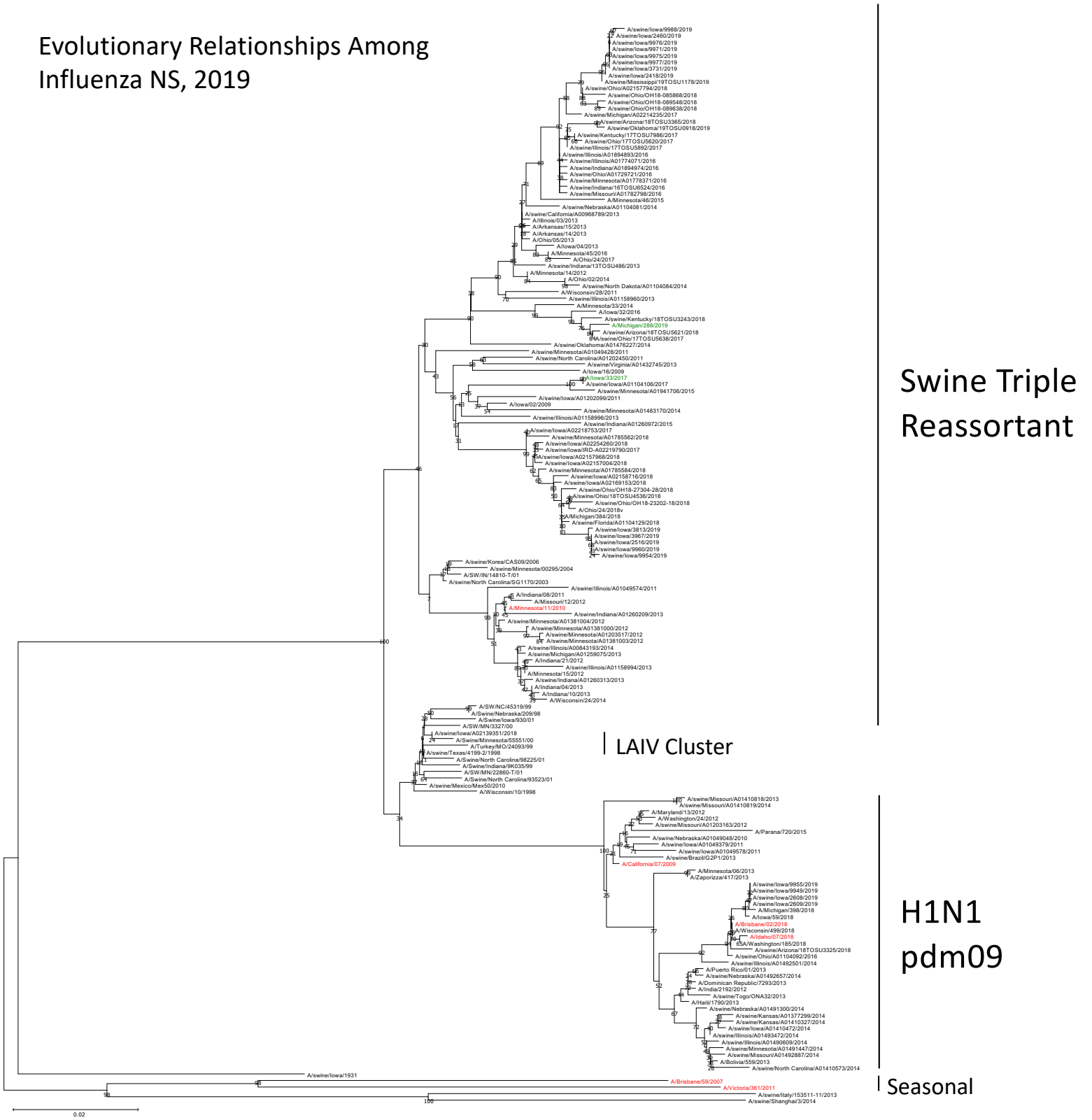
Supplementary Figure S4. NP gene segment.

The evolutionary history was inferred using the Neighbor-Joining method (26). The optimal tree with the sum of branch length = 1.11312984 is shown. The percentage of replicate trees in which the associated taxa clustered together in the bootstrap test (1000 replicates) are shown next to the branches (27). The tree is drawn to scale, with branch lengths in the same units as those of the evolutionary distances used to infer the phylogenetic tree. The evolutionary distances were computed using the Jukes-Cantor method (28) and are in the units of the number of base substitutions per site. The analysis involved 219 nucleotide sequences. All ambiguous positions were removed for each sequence pair. There were a total of 1540 positions in the final dataset. Evolutionary analyses were conducted in MEGA7 (17).



Supplementary Figure S5. M gene segment

The evolutionary history was inferred using the Neighbor-Joining method (26). The optimal tree with the sum of branch length = 1.26408353 is shown. The percentage of replicate trees in which the associated taxa clustered together in the bootstrap test (1000 replicates) are shown next to the branches (27). The tree is drawn to scale, with branch lengths in the same units as those of the evolutionary distances used to infer the phylogenetic tree. The evolutionary distances were computed using the Jukes-Cantor method (28) and are in the units of the number of base substitutions per site. The analysis involved 303 nucleotide sequences. All ambiguous positions were removed for each sequence pair. There were a total of 1027 positions in the final dataset. Evolutionary analyses were conducted in MEGA7 (17).



Supplementary Figure S6. NS gene segment

The evolutionary history was inferred using the Neighbor-Joining method (26). The optimal tree with the sum of branch length = 1.17233975 is shown. The percentage of replicate trees in which the associated taxa clustered together in the bootstrap test (1000 replicates) are shown next to the branches (27). The tree is drawn to scale, with branch lengths in the same units as those of the evolutionary distances used to infer the phylogenetic tree. The evolutionary distances were computed using the Jukes-Cantor method (28) and are in the units of the number of base substitutions per site. The analysis involved 164 nucleotide sequences. All ambiguous positions were removed for each sequence pair. There were a total of 867 positions in the final dataset. Evolutionary analyses were conducted in MEGA7 (17).

Supplementary Figure S7. R code for Random Forest Analysis

Included as an R formatted file.

```
#!/usr/bin/env Rscript

#Libraries
#Rscript vsurf.R HA_prot_set1_matrix.m
library(rpart)
library(dplyr)
library(VSURF)

(args = commandArgs(trailingOnly=TRUE))

matrixfilename <- "test"

if (length(args)==0) {
  stop("At least one argument must be supplied (input file).\n", call.=FALSE)
} else if (length(args)>=1) {
  # default output file
  matrixfilename <- args[1]
  coreNum <- args[2]
}

#Transform and Clean Data#
data1 <- read.csv(matrixfilename, header = T, stringsAsFactors = T, colClasses = "factor")

#data1[sample(nrow(data1), nrow(data1)),]
print("Number of columns, including aa position and host name")
length(data1)
print("Number of Organisms")
(samplelen <- length(data1$organism))
print("Number of Organism Included in the Model, approx 90% of full data set")
(trainlen <- as.integer(length(data1$organism) * 0.90))
print("Number of Organisms withheld for final testing, approx 10% of full data set")
(testlen <- as.integer(length(data1$organism) * 0.10))

#Training Data#
#?slice()
mld_train <- slice(data1, sample(1:samplelen)[1:trainlen])
#str(mld_train)

#Testing Data#
mld_test <- slice(data1, sample(1:samplelen)[1:testlen])
#str(mld_test)

#install.packages("VSURF")
####Using the VSURF package####
print("should match number of columns")
```

```
(mld_col_len <- length(mld_train))
df1 <- mld_train[,1:(mld_col_len-1)]
ar1 <- mld_train[,mld_col_len]
#str(ar1)
#ar1

set.seed(715, "L'Ecuyer-CMRG")

#A "5" for 5-fold cross validataion
K <- 5
x <- df1
y <- ar1
n <- length(y)
folds <- replicate(ceiling(n / K), sample(1:K))[1:n]

errtest.mat <- matrix(nrow = K, ncol = 2)
colnames(errtest.mat) <- c("interp", "pred")
res.cv <- vector("list", K)

vsurf.all.data <- list()
m = 1

for (k in 1:K) {
  print(paste("k",k))
  xtrain <- x[-which(folds == k),]
  ytrain <- y[-which(folds == k)]
  xtest <- x[which(folds == k),]
  ytest <- y[which(folds == k)]

  vsurf.fold <-
    VSURF(xtrain, ytrain, ntree = 2000, parallel = TRUE, clusterType = "FORK", ncores =
coreNum)
  errtest.mat[k, 1] <-
    sum(ytest != predict(vsurf.fold, newdata = xtest, step = "interp"))
  errtest.mat[k, 2] <-
    sum(ytest != predict(vsurf.fold, newdata = xtest, step = "pred"))
  res.cv[[k]] <- vsurf.fold
  print(vsurf.fold$varselect.pred)
  print(paste("k",k))
  current_list <- vsurf.fold$varselect.pred
  for (l in current_list) {
    vsurf.all.data[m] <- l
    m = m + 1
  }
}
```

```
(errtest <- colSums(errtest.mat) / n)
print("Error rate for the interp and predict classes")
print(errtest)
print(paste("from a",K,"-fold cross validation"))
print("5% of original data with held for final testing")

freq_of_pos <- table(unlist(vsurf.all.data[1:length(vsurf.all.data)]))
pos_keep <- freq_of_pos/K >= 0.8
print(pos_keep)
pos_keep_df <- as.data.frame(pos_keep)

(mld_train_k10 <- rownames(pos_keep_df)[which(pos_keep_df$pos_keep)])

xnam <- paste0("pos", mld_train_k10)
(fmla <- as.formula(paste("organism ~ ", paste(xnam, collapse= "+"))))

print("Begining the fitting")
#Fittin the Model
fit <- rpart(formula = fmla,
        data = mld_train,
        method = 'class',
        control = rpart.control(minsplit = 2, cp = 0))
print("Fit was made")

save(mld_train_k10,file=paste0(pname,"_model_sites.rdf"))

#Testing the Model against the training data all together
prediction_train <- predict(fit, mld_train, type = 'class')
results_train <- data.frame(trueHost=mld_train$organism, predictedHost = prediction)
print("table of training data to fit model")
table(results_train)

#Testing the Model against unused data
prediction_test <- predict(fit, mld_test, type = 'class')
results_test <- data.frame(trueHost=mld_test$organism, predictedHost = prediction)
print("table of testing data to fit model-unused in original fit")
table(results_test)
```