Population Genomics Final Project Report

Jacob Halle

Introduction

The threat of viral outbreaks is an ongoing and increasingly dangerous threat. In 2003, the SARS epidemic caused 774 deaths with an alarming 9.6% fatality rate[1]. Luckily, SARS lacked the transmissibility to result in a full global pandemic. In 2019, SARS mutated into the highly contagious COVID-19 coronavirus, which has been responsible for over 7 million deaths[1] and is still dangerous for sensitive populations. To develop effective vaccines preemptively, it is paramount to be able understand evolutionary pressure on viral genomes such as SARS and COVID-19. One such method for this task is evaluating the dN/dS ratio of viral protein coding genes. The dN/dS ratio provides valuables insights into the evolutionary pressure placed on genes. The ratio captures the proportion of non-synonymous mutations versus synonymous mutation. A non-synonymous mutation is a change in nucleotide that results in a different amino acid, while synonymous mutations do not affect the resulting protein. In general, a dN/dS ratio greater than 1 suggests positive selection, and less than 1 suggests purifying selection.

Materials and Methods

Coding DNA sequences (CDS) were obtained using the "biomartr" package (v1.0.9) in R (v4.4.1) and executed in Rstudio (v2024.04.2+764)[2]. CDSs were obtained from the NCBI refseq database, using "GCF_000864885.1" as the reference assembly for SARS, and "GCF_009858895.2" for COVID-19. The "orthologr" package (v0.4.2) was used to generate dN/dS values.[3] This package utilized BLAST (v2.16.0), "DIAMOND" (v2.1.10), and "KaKs_Calculator" v(2.0)[4, 5, 6]. All software was downloaded directly onto my personal computer running "Windows 10 Pro" v(10.0.19045). The "dNdS" function was run from "orthologr" with the COVID-19 assembly as the query file, SARS as the subject file, reciprocal best hit approach for identifying orthologs, pairwise alignment of sequences, pal2nal for protein alignment conversion to nucleotide alignments, Needleman-Wunsch method for amino acid alignment, and specified not to delete corrupt CDS. dNdS estimation was calculated three times using Nei, M. and Gojobori, T. (NG), Yang, Z. and Nielsen, R (YN), and Modified: Zhang, Z., et al (MYN). All other parameters were left at default value.

Results

The total number of CDS found in COVID-19 and SARS was 13 and 11, respectively. Of these, 9 proteins were identified as orthologs under the reciprocal best hit approach. Table 1. Shows the dNdS results for each of the methods tested.

Table 1. dNdS Results Using Three Different Estimation Methods

a. YN

Query ID	Subject ID	dN	dS	dNdS	Percent identity
lcl|NC_045512.2_cds_YP_009724390.1_3	lcl|NC_004718.3_cds_YP_009825051.1_3	0.139118	4.92275	0.0282602	74.8
lcl|NC_045512.2_cds_YP_009725295.1_2	lcl|NC_004718.3_cds_YP_009944365.1_2	0.113247	2.7329	0.0414383	79.9
lcl|NC_045512.2_cds_YP_009724389.1_1	lcl|NC_004718.3_cds_NP_828849.7_1	0.0773269	1.71398	0.0451153	85.8
lcl|NC_045512.2_cds_YP_009724393.1_6	lcl|NC_004718.3_cds_YP_009825055.1_7	0.0522319	0.629234	0.0830087	90.6
lcl|NC_045512.2_cds_YP_009724391.1_4	lcl|NC_004718.3_cds_YP_009825052.1_4	0.170543	1.54327	0.110508	72.5
lcl|NC_045512.2_cds_YP_009724395.1_8	lcl|NC_004718.3_cds_YP_009825057.1_9	0.0837364	0.72435	0.115602	85.4
lcl|NC_045512.2_cds_YP_009724397.2_11	lcl|NC_004718.3_cds_YP_009825061.1_13	0.0544983	0.380286	0.143309	91
lcl|NC_045512.2_cds_YP_009724394.1_7	lcl|NC_004718.3_cds_YP_009825056.1_8	0.170516	1.07937	0.157976	68.9
lcl|NC_045512.2_cds_YP_009724392.1_5	lcl|NC_004718.3_cds_YP_009825054.1_6	0.0264306	0.152792	0.172985	94.8

b. NG

Query ID	Subject ID	dN	dS	dNdS	Percent identity
lcl|NC_045512.2_cds_YP_009724389.1_1	lcl|NC_004718.3_cds_NP_828849.7_1	0.0949335	1.10837	0.0856519	85.8
lcl|NC_045512.2_cds_YP_009725295.1_2	lcl|NC_004718.3_cds_YP_009944365.1_2	0.135124	1.35038	0.100063	79.9
lcl|NC_045512.2_cds_YP_009724393.1_6	lcl|NC_004718.3_cds_YP_009825055.1_7	0.0633304	0.601353	0.105313	90.6
lcl|NC_045512.2_cds_YP_009724390.1_3	lcl|NC_004718.3_cds_YP_009825051.1_3	0.164546	1.5383	0.106966	74.8
lcl|NC_045512.2_cds_YP_009724395.1_8	lcl|NC_004718.3_cds_YP_009825057.1_9	0.0826541	0.739171	0.11182	85.4
lcl|NC_045512.2_cds_YP_009724397.2_11	lcl|NC_004718.3_cds_YP_009825061.1_13	0.0555083	0.386608	0.143578	91
lcl|NC_045512.2_cds_YP_009724394.1_7	lcl|NC_004718.3_cds_YP_009825056.1_8	0.17643	0.97249	0.181421	68.9
lcl|NC_045512.2_cds_YP_009724391.1_4	lcl|NC_004718.3_cds_YP_009825052.1_4	0.177568	0.97093	0.182885	72.5
lcl|NC_045512.2_cds_YP_009724392.1_5	lcl|NC_004718.3_cds_YP_009825054.1_6	0.030707	0.150084	0.204599	94.8

c. MYN

Query ID	Subject ID	dN	dS	dNdS	Percent identity
lcl|NC_045512.2_cds_YP_009725295.1_2	lcl|NC_004718.3_cds_YP_009944365.1_2	0.112225	5.9311	0.0189215	79.9
lcl|NC_045512.2_cds_YP_009724390.1_3	lcl|NC_004718.3_cds_YP_009825051.1_3	0.136865	4.87033	0.0281018	74.8
lcl|NC_045512.2_cds_YP_009724389.1_1	lcl|NC_004718.3_cds_NP_828849.7_1	0.0763289	2.11051	0.036166	85.8
lcl|NC_045512.2_cds_YP_009724393.1_6	lcl|NC_004718.3_cds_YP_009825055.1_7	0.0494167	0.837613	0.058997	90.6
lcl|NC_045512.2_cds_YP_009724397.2_11	lcl|NC_004718.3_cds_YP_009825061.1_13	0.0507127	0.678607	0.0747306	91
lcl|NC_045512.2_cds_YP_009724391.1_4	lcl|NC_004718.3_cds_YP_009825052.1_4	0.166164	2.17863	0.0762701	72.5
lcl|NC_045512.2_cds_YP_009724395.1_8	lcl|NC_004718.3_cds_YP_009825057.1_9	0.0835135	0.734039	0.113772	85.4
lcl|NC_045512.2_cds_YP_009724394.1_7	lcl|NC_004718.3_cds_YP_009825056.1_8	0.170061	1.34389	0.126544	68.9
lcl|NC_045512.2_cds_YP_009724392.1_5	lcl|NC_004718.3_cds_YP_009825054.1_6	0.0270485	0.148266	0.182432	94.8

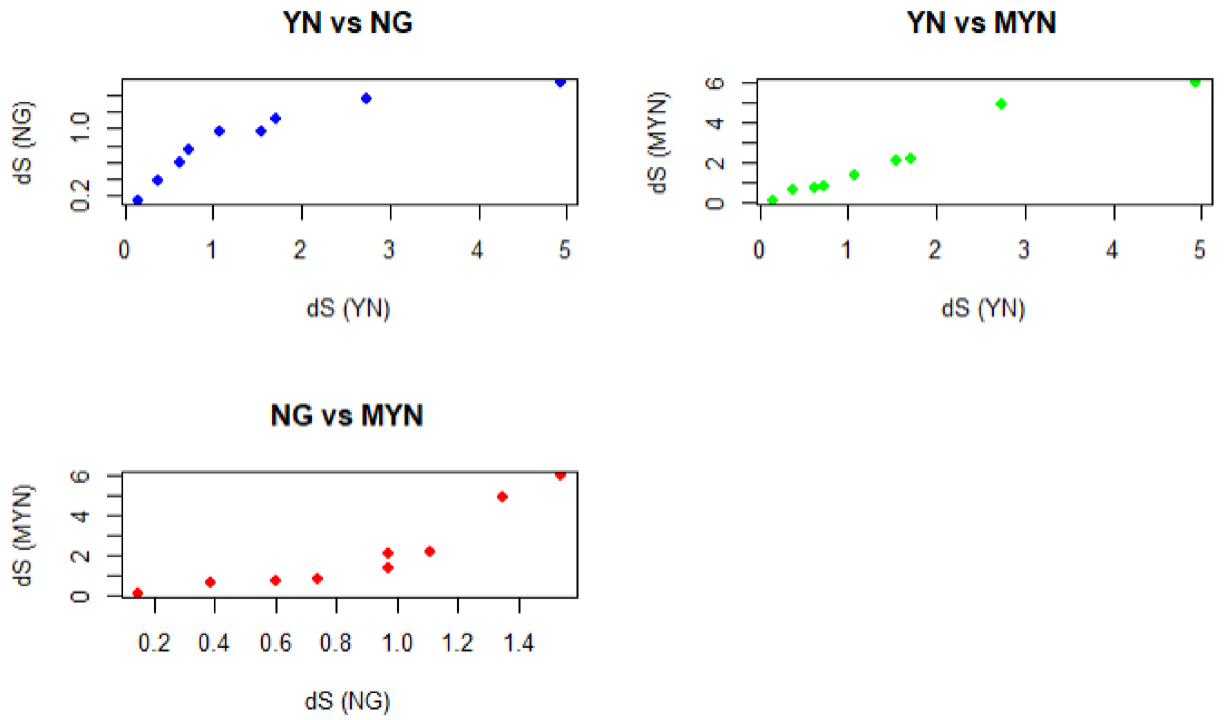
Table 2 shows the average dN, dS, and dN/dS +/- standard error.

Table 2. Average dN, dS, and dN/dS

	dN	dS	dN/dS
YN	0.0986 +/- 0.0186	1.5421 +/- 0.4982	0.0998 +/- 0.0178
NG	0.1090 +/- 0.0186	0.8686 +/- 0.1489	0.1358 +/- 0.0145
MYN	0.0969 +/- 0.0174	2.0925 +/- 0.6689	0.0795 +/- 0.0177

Figure 1 shows biplots comparing the ordered dS values for each pair of the three methods.

Figure 1. Ordered dS values for each pair of dN/dS estimation methods



Discussion

The dNdS values for every ortholog pair under each estimation method suggest that all genes in the COVID-19 genome are subject to purifying selection. This is evident from all dN/dS values being less than 1. The highest value across all orthologs and all methods was only 0.2, suggesting that all genes in the covid-19 genome have a high effect on fitness and are largely conserved as a result. Some genes show high dS values over 1, indicating that these genes are experiencing mutations at a higher rate than is expected under neutral evolution. This could be due to the unstable RNA genome duplication process that results in an accumulation of mutations. Nonsynonymous mutations get filtered out, as evident by low dN/dS values, while the synonymous mutations are tolerated and able to remain. The MYN and YN estimation methods produced dS estimates significantly larger than the NG method. A possible interpretation of this is the NG method does not account for multiple mutations at the same location, while MYN and YN account for this possibility. The gene under the most selective constraint as determined by each method is YP_009724390.1_3, YP_009724389.1_1, or YP_009725295.1_2, according to YN, NG, or MYN, respectively.

The spike protein of viral genomes allows viral particles to enter cell membranes, where they replicate. This is a critical feature of viruses and is thus subject to high selective pressure. All three estimation methods have this gene (YP_009724390.1) in the top three most conserved sequences. The dN, dS, dN/dS, and precent identity for the spike protein approximated by the MYN method is 0.076, 2.11, 0.036, and 85.8%, respectively. In conclusion, many viral genes experience a higher mutation rate than neutral evolution, and all are heavily conserved.

References

1. Pormohammad A, Ghorbani S, Khatami A, Farzi R, Baradaran B, Turner DL, Turner RJ, Bahr NC, Idrovo JP. Comparison of confirmed COVID-19 with SARS and MERS cases - Clinical characteristics, laboratory findings, radiographic signs and outcomes: A systematic review and meta-analysis. Rev Med Virol. 2020 Jul;30(4):e2112. doi: 10.1002/rmv.2112. Epub 2020 Jun 5. PMID: 32502331; PMCID: PMC7300470.
2. Drost HG, Paszkowski J. **Biomartr: genomic data retrieval with R**. *Bioinformatics* (2017) 33(8): 1216-1217. doi:10.1093/bioinformatics/btw821.
3. Drost et al. 2015. **Evidence for Active Maintenance of Phylotranscriptomic Hourglass Patterns in Animal and Plant Embryogenesis**. *Mol. Biol. Evol.* 32 (5): 1221-1231. doi:10.1093/molbev/msv012
4. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990 Oct 5;215(3):403-10. doi: 10.1016/S0022-2836(05)80360-2. PMID: 2231712.
5. Buchfink B, Reuter K, Drost HG, "Sensitive protein alignments at tree-of-life scale using DIAMOND", *Nature Methods* **18**, 366–368 (2021). doi:10.1038/s41592-021-01101-x
6. Da-Peng Wang, Hao-Lei Wan, Song Zhang and Jun Yu. γ-MYN: a new algorithm for estimating Ka and Ks with consideration of variable substitution rates. Biology Direct 2009, 4:20.

Population Genetics and Evolutionary Biology Final Project: Jacob Halle

2024-12-18

```
library(biomartr)
library(orthologr)
```

```
## Loading required package: data.table
```

```
## Warning: package 'data.table' was built under R version 4.4.2
```

```
# download all coding sequences for SARS
SARS_file <- biomartr::getCDS(db = "refseq", organism = "GCF_000864885.1", path = getwd())
```

```
## -> Starting CDS retrieval of 'GCF_000864885.1' from refseq ...
```

```
##
```

```
## It seems that this is the first time you run this command for refseq .
```

```
## Thus, 'assembly_summary.txt' files for all kingdoms will be retrieved from refseq.
```

```
## Don't worry this has to be done only once if you don't restart your R session.
```

```
##
```

```
## Due to its extended dataset size (GenBank: >700 MB, RefSeq: >150 MB) Kingdom 'bacteria' will
not be downloaded by default anymore. To also include 'bacteria' please specify the argument 'sk
ip_bacteria = FALSE'
##
## -> Starting download for: archaea
## --------> Skipping bacteria download .....
##
## -> Starting download for: fungi
## -> Starting download for: invertebrate
## -> Starting download for: plant
## -> Starting download for: protozoa
## -> Starting download for: vertebrate_mammalian
## -> Starting download for: vertebrate_other
## -> Starting download for: viral
```

```
##
```

```
## Completed!
```

```
## Now continue with species download ...
```

```
## File C:/Users/jacob/OneDrive/Documents/Masters/Population Genomics/Final_project/GCF_00086488
5.1_cds_from_genomic_refseq.fna.gz exists already. Thus, download has been skipped.
```

```
## -> CDS download of GCF_000864885.1 is completed!
```

```
## -> Checking md5 hash of file: C:/Users/jacob/OneDrive/Documents/Masters/Population Genomics/F
inal_project/GCF_000864885.1_cds_from_genomic_refseq.fna.gz (md5: 30a60961f2968e15ed6b8c1d408536
6d) ...
```

```
## -> The md5 hash of file 'C:/Users/jacob/OneDrive/Documents/Masters/Population Genomics/Final_
project/GCF_000864885.1_md5checksums.txt' matches!
```

```
## -> The CDS has been downloaded to 'C:/Users/jacob/OneDrive/Documents/Masters/Population Genom
ics/Final_project' and has been named 'GCF_000864885.1_cds_from_genomic_refseq.fna.gz'.
```

```
##
##
## Please cite: Drost HG, Paszkowski J. Biomartr: genomic data retrieval with R. Bioinformatics
(2017) 33(8): 1216-1217. doi:10.1093/bioinformatics/btw821.
```

```
# download all coding sequences for Covid-19
Covid19_file <- biomartr::getCDS(db = "refseq", organism = "GCF_009858895.2", path = getwd())
```

```
## -> Starting CDS retrieval of 'GCF_009858895.2' from refseq ...
```

```
##
```

```
## File C:/Users/jacob/OneDrive/Documents/Masters/Population Genomics/Final_project/GCF_00985889
5.2_cds_from_genomic_refseq.fna.gz exists already. Thus, download has been skipped.
```

```
## -> CDS download of GCF_009858895.2 is completed!
```

```
## -> Checking md5 hash of file: C:/Users/jacob/OneDrive/Documents/Masters/Population Genomics/F
inal_project/GCF_009858895.2_cds_from_genomic_refseq.fna.gz (md5: bf495657f0be6901ad9ae8a2a5e4f3
51) ...
```

```
## -> The md5 hash of file 'C:/Users/jacob/OneDrive/Documents/Masters/Population Genomics/Final_
project/GCF_009858895.2_md5checksums.txt' matches!
```

```
## -> The CDS has been downloaded to 'C:/Users/jacob/OneDrive/Documents/Masters/Population Genom
ics/Final_project' and has been named 'GCF_009858895.2_cds_from_genomic_refseq.fna.gz'.
```

```
##
##
## Please cite: Drost HG, Paszkowski J. Biomartr: genomic data retrieval with R. Bioinformatics
(2017) 33(8): 1216-1217. doi:10.1093/bioinformatics/btw821.
```

```
# compute dN/dS values using NG
dNdS_NG <-
  dNdS(query_file      = Covid19_file,
       subject_file    = SARS_file,
       delete_corrupt_cds = FALSE,
       ortho_detection = "RBH",
       aa_aln_type     = "pairwise",
       aa_aln_tool     = "NW",
       codon_aln_tool  = "pal2nal",
       dnds_est.method = "NG",
       comp_cores      = 1 )
```

```
##
```

```
## Starting orthology inference (RBH) and dNdS estimation (NG) using the follwing parameters:
```

```
## query = 'GCF_009858895.2_cds_from_genomic_refseq.fna.gz'
```

```
## subject = 'GCF_000864885.1_cds_from_genomic_refseq.fna.gz'
```

```
## aligner = 'diamond'
```

```
## sensitivity_mode = 'fast'
```

```
## seq_type = 'cds'
```

```
## e-value: 1E-5
```

```
## aa_aln_type = 'pairwise'
```

```
## aa_aln_tool = 'NW'
## comp_cores = '1'
```

```
##
```

```
## Starting Orthology Inference ...
```

```
## Running diamond version 2.1.10 ...
```

```
## sensitivity mode: fast
```

```
## creating a diamond database
```

```
## Running diamond version 2.1.10 ...
```

```
## sensitivity mode: fast
```

```
## creating a diamond database
```

```
## Orthology Inference Completed.
```

```
## Starting dN/dS Estimation ...
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## dN/dS Estimation Completed.
```

```
##
```

```
## Please cite the following paper when using orthologr for your own research:
```

```
## Drost et al. Evidence for Active Maintenance of Phylotranscriptomic Hourglass Patterns in Ani
mal and Plant Embryogenesis. 2015. Mol. Biol. Evol. 32 (5): 1221-1231.
```

```
##
```

```
##
```

YN

```
dNdS_YN <-
  dNdS(query_file      = Covid19_file,
       subject_file    = SARS_file,
       delete_corrupt_cds = FALSE,
       ortho_detection = "RBH",
       aa_aln_type     = "pairwise",
       aa_aln_tool     = "NW",
       codon_aln_tool  = "pal2nal",
       dnds_est.method = "YN",
       comp_cores      = 1 )
```

```
##
```

```
## Starting orthology inference (RBH) and dNdS estimation (YN) using the follwing parameters:
```

```
## query = 'GCF_009858895.2_cds_from_genomic_refseq.fna.gz'
```

```
## subject = 'GCF_000864885.1_cds_from_genomic_refseq.fna.gz'
```

```
## aligner = 'diamond'
## sensitivity_mode = 'fast'
```

```
## seq_type = 'cds'
```

```
## e-value: 1E-5
```

```
## aa_aln_type = 'pairwise'
```

```
## aa_aln_tool = 'NW'
```

```
## comp_cores = '1'
```

```
##
```

```
## Starting Orthology Inference ...
```

```
## Running diamond version 2.1.10 ...
```

```
## sensitivity mode: fast
```

```
## creating a diamond database
```

```
## Running diamond version 2.1.10 ...
```

```
## sensitivity mode: fast
```

```
## creating a diamond database
```

```
## Orthology Inference Completed.
```

```
## Starting dN/dS Estimation ...
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## dN/dS Estimation Completed.
```

```
##
```

```
## Please cite the following paper when using orthologr for your own research:
```

```
## Drost et al. Evidence for Active Maintenance of Phylotranscriptomic Hourglass Patterns in Ani
mal and Plant Embryogenesis. 2015. Mol. Biol. Evol. 32 (5): 1221-1231.
```

```
##
```

```
##
```

MYN

```
dNdS_MYN <-
  dNdS(query_file      = Covid19_file,
       subject_file    = SARS_file,
       delete_corrupt_cds = FALSE,
       ortho_detection = "RBH",
       aa_aln_type     = "pairwise",
       aa_aln_tool     = "NW",
       codon_aln_tool  = "pal2nal",
       dnds_est.method = "MYN",
       comp_cores      = 1 )
```

```
##
```

```
## Starting orthology inference (RBH) and dNdS estimation (MYN) using the follwing parameters:
```

```
## query = 'GCF_009858895.2_cds_from_genomic_refseq.fna.gz'
```

```
## subject = 'GCF_000864885.1_cds_from_genomic_refseq.fna.gz'
```

```
## aligner = 'diamond'
```

```
## sensitivity_mode = 'fast'
```

```
## seq_type = 'cds'
```

```
## e-value: 1E-5
```

```
## aa_aln_type = 'pairwise'
```

```
## aa_aln_tool = 'NW'
```

```
## comp_cores = '1'
```

```
##
```

```
## Starting Orthology Inference ...
```

```
## Running diamond version 2.1.10 ...
```

```
## sensitivity mode: fast
## creating a diamond database
```

```
## Running diamond version 2.1.10 ...
```

```
## sensitivity mode: fast
```

```
## creating a diamond database
```

```
## Orthology Inference Completed.
```

```
## Starting dN/dS Estimation ...
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("pairwiseAlignment", ...): pairwiseAlignment() has moved to t
he pwalign package. Please call
##   pwalign::pairwiseAlignment() to get rid of this warning.
```

```
## Warning in .call_fun_in_pwalign("writePairwiseAlignments", ...): writePairwiseAlignments() ha
s moved to the pwalign package. Please call
##   pwalign::writePairwiseAlignments() to get rid of this warning.
```

```
## dN/dS Estimation Completed.
```

```
##
```

```
## Please cite the following paper when using orthologr for your own research:
```

```
## Drost et al. Evidence for Active Maintenance of Phylotranscriptomic Hourglass Patterns in Ani
mal and Plant Embryogenesis. 2015. Mol. Biol. Evol. 32 (5): 1221-1231.
```

```
##
```

```
##
```

```
# Extract the most relevant columns
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.4.2
```

```
YN_table = dNdS_YN[c("query_id","subject_id","dN","dS","dNdS","perc_identity")]
YN_table
```

```
## # A tibble: 9 × 6
##   query_id                          subject_id     dN    dS   dNdS perc_identity
##   <chr>                             <chr>       <dbl> <dbl>  <dbl>         <dbl>
## 1 lcl|NC_045512.2_cds_YP_009724389… lcl|NC_00… 0.0773 1.71  0.0451          85.8
## 2 lcl|NC_045512.2_cds_YP_009724390… lcl|NC_00… 0.139  4.92  0.0283          74.8
## 3 lcl|NC_045512.2_cds_YP_009724391… lcl|NC_00… 0.171  1.54  0.111           72.5
## 4 lcl|NC_045512.2_cds_YP_009724392… lcl|NC_00… 0.0264 0.153 0.173           94.8
## 5 lcl|NC_045512.2_cds_YP_009724393… lcl|NC_00… 0.0522 0.629 0.0830          90.6
## 6 lcl|NC_045512.2_cds_YP_009724394… lcl|NC_00… 0.171  1.08  0.158           68.9
## 7 lcl|NC_045512.2_cds_YP_009724395… lcl|NC_00… 0.0837 0.724 0.116           85.4
## 8 lcl|NC_045512.2_cds_YP_009724397… lcl|NC_00… 0.0545 0.380 0.143           91  
## 9 lcl|NC_045512.2_cds_YP_009725295… lcl|NC_00… 0.113  2.73  0.0414          79.9
```

```
NG_table = dNdS_NG[c("query_id","subject_id","dN","dS","dNdS","perc_identity")]
NG_table
```

```
## # A tibble: 9 × 6
##   query_id                          subject_id     dN    dS   dNdS perc_identity
##   <chr>                             <chr>       <dbl> <dbl>  <dbl>         <dbl>
## 1 lcl|NC_045512.2_cds_YP_009724389… lcl|NC_00… 0.0949 1.11  0.0857          85.8
## 2 lcl|NC_045512.2_cds_YP_009724390… lcl|NC_00… 0.165  1.54  0.107           74.8
## 3 lcl|NC_045512.2_cds_YP_009724391… lcl|NC_00… 0.178  0.971 0.183           72.5
## 4 lcl|NC_045512.2_cds_YP_009724392… lcl|NC_00… 0.0307 0.150 0.205           94.8
## 5 lcl|NC_045512.2_cds_YP_009724393… lcl|NC_00… 0.0633 0.601 0.105           90.6
## 6 lcl|NC_045512.2_cds_YP_009724394… lcl|NC_00… 0.176  0.972 0.181           68.9
## 7 lcl|NC_045512.2_cds_YP_009724395… lcl|NC_00… 0.0827 0.739 0.112           85.4
## 8 lcl|NC_045512.2_cds_YP_009724397… lcl|NC_00… 0.0555 0.387 0.144           91  
## 9 lcl|NC_045512.2_cds_YP_009725295… lcl|NC_00… 0.135  1.35  0.100           79.9
```

```
MYN_table = dNdS_MYN[c("query_id","subject_id","dN","dS","dNdS","perc_identity")]
MYN_table
```

```
## # A tibble: 9 × 6
##   query_id                          subject_id     dN    dS   dNdS perc_identity
##   <chr>                             <chr>       <dbl> <dbl>  <dbl>         <dbl>
## 1 lcl|NC_045512.2_cds_YP_009724389… lcl|NC_00… 0.0763 2.11  0.0362          85.8
## 2 lcl|NC_045512.2_cds_YP_009724390… lcl|NC_00… 0.137  4.87  0.0281          74.8
## 3 lcl|NC_045512.2_cds_YP_009724391… lcl|NC_00… 0.166  2.18  0.0763          72.5
## 4 lcl|NC_045512.2_cds_YP_009724392… lcl|NC_00… 0.0270 0.148 0.182           94.8
## 5 lcl|NC_045512.2_cds_YP_009724393… lcl|NC_00… 0.0494 0.838 0.0590          90.6
## 6 lcl|NC_045512.2_cds_YP_009724394… lcl|NC_00… 0.170  1.34  0.127           68.9
## 7 lcl|NC_045512.2_cds_YP_009724395… lcl|NC_00… 0.0835 0.734 0.114           85.4
## 8 lcl|NC_045512.2_cds_YP_009724397… lcl|NC_00… 0.0507 0.679 0.0747          91  
## 9 lcl|NC_045512.2_cds_YP_009725295… lcl|NC_00… 0.112  5.93  0.0189          79.9
```

```
# Find the mean and standard error for dN, dS, and dN/dS for each method
mean(dNdS_YN$dN)
```

```
## [1] 0.09862757
```

```
sd(dNdS_YN$dN) / sqrt(9)
```

```
## [1] 0.01755183
```

```
mean(dNdS_YN$dS)
```

```
## [1] 1.542104
```

```
sd(dNdS_YN$dS) / sqrt(9)
```

```
## [1] 0.4982259
```

```
mean(dNdS_YN$dNdS)
```

```
## [1] 0.09980028
```

```
sd(dNdS_YN$dNdS) / sqrt(9)
```

```
## [1] 0.0177929
```

```
mean(dNdS_NG$dN)
```

```
## [1] 0.1089779
```

```
sd(dNdS_NG$dN) / sqrt(9)
```

```
## [1] 0.01862832
```

```
mean(dNdS_NG$dS)
```

```
## [1] 0.8686318
```

```
sd(dNdS_NG$dS) / sqrt(9)
```

```
## [1] 0.1488903
```

```
mean(dNdS_NG$dNdS)
```

```
## [1] 0.1358108
```

```
sd(dNdS_NG$dNdS) / sqrt(9)
```

```
## [1] 0.01453919
```

```
mean(dNdS_MYN$dN)
```

```
## [1] 0.09692614
```

```
sd(dNdS_MYN$dN) / sqrt(9)
```

```
## [1] 0.0174211
```

```
mean(dNdS_MYN$dS)
```

```
## [1] 2.092554
```

```
sd(dNdS_MYN$dS) / sqrt(9)
```

```
## [1] 0.6689404
```

```
mean(dNdS_MYN$dNdS)
```

```
## [1] 0.07954833
```

```
sd(dNdS_MYN$dNdS) / sqrt(9)
```

```
## [1] 0.01772694
```

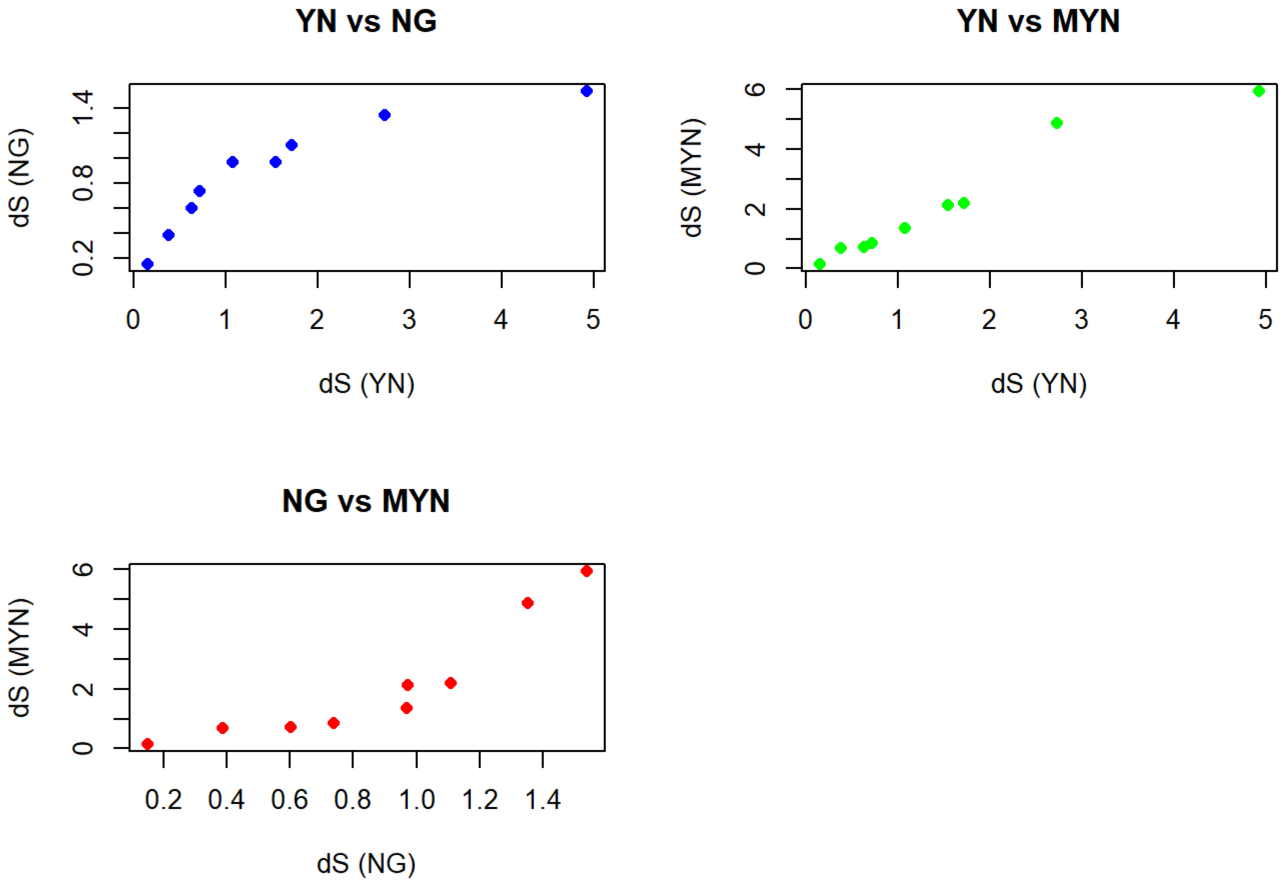
```
# Create biplots
dNdS_MYN$dS = dNdS_MYN$dS[order(dNdS_MYN$dS)]
dNdS_YN$dS = dNdS_YN$dS[order(dNdS_YN$dS)]
dNdS_NG$dS = dNdS_NG$dS[order(dNdS_NG$dS)]

# Set up plot
par(mfrow = c(2, 2))

# YN vs NG
plot(dNdS_YN$dS, dNdS_NG$dS,
     xlab = "dS (YN)", ylab = "dS (NG)",
     main = "YN vs NG",
     pch = 16, col = "blue")

# YN vs MYN
plot(dNdS_YN$dS, dNdS_MYN$dS,
     xlab = "dS (YN)", ylab = "dS (MYN)",
     main = "YN vs MYN",
     pch = 16, col = "green")

# NG vs MYN
plot(dNdS_NG$dS, dNdS_MYN$dS,
     xlab = "dS (NG)", ylab = "dS (MYN)",
     main = "NG vs MYN",
     pch = 16, col = "red")
```


YN vs NG
dS (NG)
dS (YN)
YN vs MYN
dS (MYN)
dS (YN)
NG vs MYN
dS (MYN)
dS (NG)