submitted to Bioinformatics, 2025, 1–9
doi:

Sequence analysis

AnchoRNA: Full virus genome alignments through conserved anchor regions

Tom Eulenfeld,[1,2,3] Sandra Triebel[1,2] and Manja Marz[1-5]

[1]Bioinformatics and High-Throughput Analysis, Friedrich Schiller University Jena, 07743 Jena, Germany, [2]European Virus Bioinformatics Center, 07743 Jena, Germany, [3]Cluster of Excellence Balance of the Microverse, Friedrich Schiller University Jena, 07743 Jena, Germany, [4]German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, 04103 Leipzig, Germany and [5]Fritz Lipmann Institute-Leibniz Institute on Aging, 07745 Jena, Germany

Corresponding authors: tom.eulenfeld@uni-jena.de, manja@uni-jena.de

Abstract

Motivation: Multiple sequence alignment of full viral genomes can be challenging due to factors such as long sequences, large insertions/deletions (spanning several 100 nucleotides), high number of sequences, sequence divergence, and high computational complexity, as e.g., in the context of secondary structure prediction. Standard alignment methods often face these issues, especially when processing highly variable sequences or when specific phylogenetic analysis is required on selected subsequences. We present `AnchoRNA`, a Python-based command line tool designed to identify conserved regions, or anchors, within coding sequences. These anchors define split positions, guiding the alignment of complex viral genomes, including those with significant secondary structures. `AnchoRNA` enhances the accuracy and efficiency of full-genome alignment by focusing on these crucial conserved regions. The presented approach can be particularly useful when designing primers conserved across a virus family.

Results: `AnchoRNA` guided alignments are systematically compared to the results of 3 alignment programs. Utilizing a dataset of 55 representative *Pestivirus* genomes, `AnchoRNA` identified 56 anchors that are crucial for guiding the alignment process. The incorporation of these anchors led to significant improvements across all tested alignment tools, highlighting the effectiveness of `AnchoRNA` in enhancing alignment quality, especially in complex viral genomes.

Availability: `AnchoRNA` is available to the scientific community under the MIT license on GitHub at `https://github.com/rnajena/anchorna`, with releases archived on Zenodo. The package includes a tutorial featuring a *Pestivirus* dataset and is compatible with all platforms that support Python.

Key words: Multiple sequence alignment, Conserved regions, Viral genomes, Anchors, Potential primer design

Introduction

Multiple sequence alignment (MSA) is a crucial step in biological sequence analysis, serving as the basis for many downstream applications, including functional annotation, evolutionary studies, and structural predictions (e.g. Lamkiewicz et al., 2023). Despite its importance, MSA remains a challenging problem due to its inherent complexity and the diversity of biological sequences, particularly in the context of viral genomes (e.g. Hufsky et al., 2023, 2020; Madhugiri et al., 2014; Hölzer and Marz, 2020). The traditional approach, which focuses on local alignments, has evolved significantly, leading to advancements in alignment accuracy with tools like `ClustalW` (Thompson et al., 1994), `T-Coffee` (Notredame et al., 2000), `MUSCLE` (Edgar, 2004), `MAFFT` (Katoh, 2002), and `DIALIGN` (Morgenstern et al., 1998). These methods have improved the alignment process by integrating local alignment information into global alignment frameworks and incorporating homology data from databases. However, aligning highly variable sequences, such as full viral genomes, still poses significant challenges that require more specialized approaches. Despite these advancements, current MSA tools face limitations when handling certain complexities. Specifically, they often struggle with long sequences, large insertions

© The Author 2025. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

or deletions (as known for e.g. pestiviruses), and highly divergent sequences, which can lead to suboptimal full virus genome alignments or excessive computational demands, e.g., when simultaneously annotating RNA secondary structures with `LocARNA` (Will et al., 2012). Anchors help by focusing on conserved regions, improving alignment accuracy and reducing computational demands despite long sequences, large insertions and deletions, or high divergence. Tools with anchors have been developed for (i) very large genomes: `MUMmer` (Delcher et al., 1999), `Mugsy` (Angiuoli and Salzberg, 2010); (ii) handling duplications and genome rearrangements: `Mauve` (Darling et al., 2004), `progressiveMauve` (Darling et al., 2010), `TBA` (Blanchette et al., 2004); (iii) comparative genomics especially in the context of phylogenetics: `LAGAN`, `Multi-LAGAN` (Brudno et al., 2003b), `AVID` (Bray et al., 2002); (iv) precise alignments based on anchor points: `CHAOS/DIALIGN` (Brudno et al., 2003a), `VISTA genome pipeline` (Dubchak et al., 2009); (v) advanced algorithmic techniques, such as graph-based methods or spaced-seeds: `ACANA` (Huang et al., 2005)), `FSWM` (Leimeister et al., 2018), `OWEN` (Ogurtsov et al., 2002); (vi) extended general multiple sequence alignment: `ClustalW` (Thompson et al., 1994), `T-Coffee` (Notredame et al., 2000), `MAFFT` (Katoh, 2002); and (vii) larger insertions and deletions: `DIALIGN` (Morgenstern et al., 1998), `DIALIGN2` (Morgenstern, 1999).

Viral genomes are small, compact, and generally less structurally complex compared to the genomes of larger organisms. Therefore, the specialized tools designed for handling large genomes, complex rearrangements, or extensive duplications are not suitable. Currently, most studies on virus alignments are performed by `MAFFT`, which is often not sufficient enough for accurate full-genome alignments, especially when large insertion/deletions occurred or RNA secondary structures are of importance of the study (e.g. Madhugiri et al., 2014). For viral genomes, especially RNA viruses (up to 40000 nt), the focus is on the development of tools that can efficiently handle high mutation rates and align a large number of genomes (typically 50 to 2000) within a multiple sequence alignment. Additionally, these tools need to account for occasional RNA structural elements. As a result, many specialized tools designed for large, complex genomes or specific structural challenges are not necessary for the relatively straightforward task of full viral genome alignment.

To bridge these gaps, we present `AnchoRNA`, a new Python-based command line tool designed to enhance the alignment of complex viral sequences. `AnchoRNA` optionally translates an RNA virus genome into its protein sequence, identifies conserved regions, or anchors, within (non-)coding sequences, which can be inexact and are then used to define split-up positions, simplifying the alignment process. By focusing on these key conserved regions, `AnchoRNA` addresses the challenges of long viral genomes, large insertions/deletions, and highly divergent sequences, offering a more accurate and efficient solution for aligning both nucleotide and protein sequences. The tool's ability to handle complex viral genomes and other highly variable sequences sets it apart from existing methods, making it a valuable addition to the MSA toolkit.

Algorithm and Usage

The primary objective of `AnchoRNA` is to translate the coding sequences of the nucleotide sequences into amino acid sequences and to search for anchors on the translated strings. By employing a substitution matrix such as BLOSUM62 (Henikoff and Henikoff, 1992), this approach automatically accounts for synonymous mutations

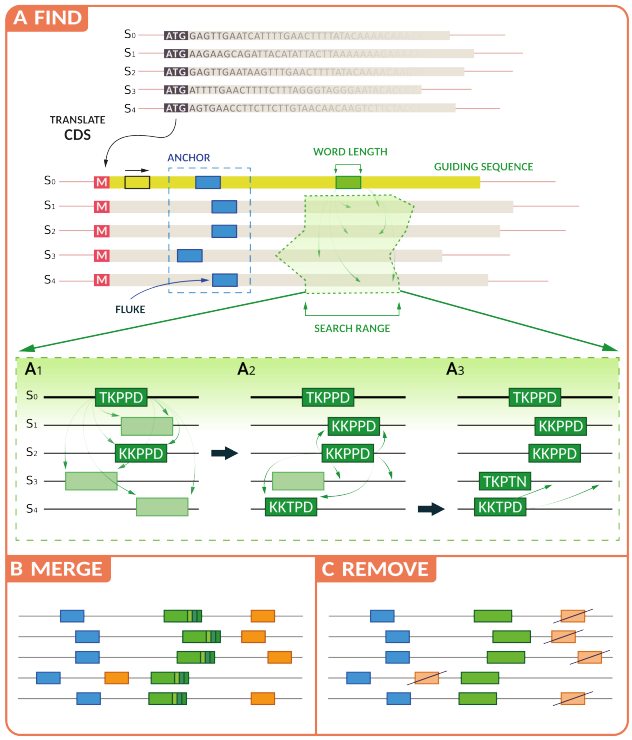

Figure 1: Algorithm for anchor detection in `AnchoRNA` used by the `anchorna go` command. It consists of three steps: **A – Find anchors of predefined word length in all sequences S_i:** The algorithm optionally translates viral RNA sequences into the amino acid sequences. The user defines one sequence as the guiding sequence and a sliding window of predefined length (default: 5) is used as initial fluke for a potential anchor. The final anchor consists of one fluke per sequence (blue). Therefore, all other virus sequences are searched for the most similar word compared to the guiding sequence (default usage `BLOSUM62`, $\mathbf{A}_1$) in a specific search range (default: ±100 characters and genome length difference; green). Highest scored word positions are used as additional flukes. When a new word is used as a fluke ($\mathbf{A}_2$), the search is repeated with this word and positions of similar words are updated with higher-scored candidates. This process is reiterated until a fluke is defined for each sequence ($\mathbf{A}_3$). A potential anchor turns into a fixed anchor if its flukes fulfill a specified condition (default: for 100 % of flukes `BLOSUM62` score ≥22). **B – Merge overlapping anchors:** Overlapping anchors are merged into longer anchors if the relative fluke positions in each sequence are the same. **C – Remove conflicting anchors:** Beginning with the highest-scoring anchor, `AnchoRNA` removes any crossing anchor. This process is then repeated for each remaining anchor in order of decreasing score.

and mutations for which the corresponding codon is translated to an amino acid with similar properties. We define the term *fluke*, which is characterized by a start and stop index on a single sequence. Consequently, an *anchor* consists of several related flukes, with one for each sequence (see Fig. 1). The use of anchors allows the original nucleotide sequences to be split into smaller subsequences that can be processed in a shorter runtime.

The command line tool `anchorna` requires two input files: the sequence file, e.g. full virus genomes, and a configuration file, see

```
### Configuration for AnchoRNA in TOML format
fname = "pesti_example.gff"# Seq file with CDS annotation
w = 5                      # Word length
gseqid = "KC533775"        # ID of the guiding sequence
search_range = 100         # Search range [#letters]
scoring = "blosum62"       # Scoring matrix
score_add_word = 22        # Score to add word to word set
            # flukes with lower score are marked as "poor"
thr_quota_add_anchor = 1   # Discard anchor if
                           # quota is not met (1=100%)
thr_score_add_anchor = 22  # Score threshold for quota
score_use_fluke = 22       # Score to use fluke in
                           # export, view, cutout
```

Figure 2: Configuration parameters for `AnchoRNA`.

Fig. 2. The sequence file is parsed by the `rnajena-sugar` package (Eulenfeld, 2024b), which currently supports either a `.gbk` file, including annotations of viral protein positions; or a `.gff` file with `fasta` directive. The configuration file (default name: `anchorna.conf`) can be created by `anchorna create` and contains all user specific features. `AnchoRNA` comes with a demonstrative example of 55 *Pestivirus* genomes. A tutorial is provided on the GitHub landing page (Eulenfeld, 2024a).

The `AnchoRNA` algorithm, see Fig. 1, is run by the `anchorna go` command. An anchor is defined as a set of equal-length ranges, called flukes, one for each sequence describing the conserved region. The coding sequences of the provided viral RNA genomes are translated into amino acid sequences. Alternatively, the algorithm can be directly applied to nucleotide or amino acid sequences (`--no-cds`). In this case, the input sequence can also be provided as `.fa` or `.stk` file. The sliding window approach of `AnchoRNA` searches for anchors by setting an initial fluke of fixed length at each position in the user-selected guiding sequence. The guiding sequence selected by the user may be a reference sequence or any other suitable sequence. However, we generally recommend to avoid short sequences as guiding sequences for a higher number of possible anchors. The default length for the sliding window and initial anchors is for protein sequences 5 amino acids. However, the optimal window size may vary depending on the level of conservation among the sequences (e.g. for highly conserved Hepatitis C Viruses a length of 8 is recommended). When using directly with nucleotide sequences, e.g., for non-coding sequences such as 5' UTR and 3' UTR the length should be in the range 10–15 nt. The search range (option `search_range`) by default is ±100 characters from the fluke position in the guiding sequence. When the searched sequence is shorter or longer than the guiding sequence, the search range is extended to the left or to the right by the difference in lengths, respectively. All virus sequences except the guiding sequence are searched for the most similar word in the search range (default `BLOSUM62` matrix), Fig. 1A$_1$, and highest scored word positions are used as additional flukes. When a newly added fluke comprises a new word which does exceed a predefined score threshold (option `score_add_word`), Fig. 1A$_2$, the search is repeated with this word and positions of similar words are updated with higher-scored candidates. Additionally, the new word is added to a reference word set, which also incorporates the initial word from the guiding sequence. This process is reiterated until one fluke is defined for each sequence, Fig. 1A$_3$. A potential anchor becomes a fixed anchor if per default all flukes score above or equal 22 (options `thr_quota_add_anchor` and `thr_score_add_anchor`). A lower threshold can be applied to find anchors in regions where some sequences contain deletions. The anchor search can be parallelized across different CPU cores (using the command line flag `--njobs`), with each core handling different positions in the guiding sequence, leading to a linear reduction in runtime as the number of cores increases. The fluke score and median fluke score are defined as the maximum and median scores of a detected word when compared to all words in the reference word set, respectively. The anchor score is the minimum of the median scores of the flukes associated with the anchor. Flukes that do not contribute a word to the reference word set are labeled as "poor". In the next step of the algorithm, overlapping anchors are merged (Fig. 1B) if the relative fluke positions in each sequence are the same, resulting in updated anchors that are either equal to or longer than the initial word length. Fluke and anchor scores of merged anchors are recalculated. Finally, starting with the highest-scoring anchor, `AnchoRNA` removes any crossing anchor (Fig. 1C). This process is then repeated for each remaining anchor in order of decreasing score. Two crossing anchors contain flukes whose start positions appear in a different order across different sequences. Additionally, overlapping anchors are removed unless otherwise specified (option `aggressive_remove`). The process of removing crossing anchors is particularly applicable to RNA viruses that rarely undergo genomic rearrangements. The resulting anchors of the `anchorna go` command are written to a file with a special format based on GFF.

`AnchoRNA` comes with several additional functionalities, see Fig. 3 and Fig. S1 in the appendix:

- `anchorna print` gives an overview of detected anchors directly to the terminal.
- `anchorna create` initializes the example configuration file. Using the `--tutorial` flag, the demonstrative example of pestiviruses can be loaded.
- `anchorna view` is loading the sequences or an alignment as well as the anchors into `Jalview` (Waterhouse et al., 2009) with special labeling and distinct colors for nearby anchors (see Fig. 3B) for a manual quality control. In the background, `anchorna export` is called.
- `anchorna cutout` creates for each virus a subsequence with anchors on the 5' and 3' end. Start and stop codons as well as start and end of sequences can also be used as anchors in this step. The cutout option can directly retrotranscribe the amino acid region and return the sequence on nucleotide level.
- `anchorna combine` can be used to manually select or delete anchors. Further, it can combine two anchor files by adjusting positions to each other.
- `anchorna load` is loading the anchors into an `IPython` session for further inspection.
- `anchorna export` creates `.gff` files with positions in the nucleotide or amino acid sequences for each virus genome. Alternatively, a `Jalview` feature file can be exported, as used by `anchorna view`. The command can also be used to export anchors to files as used by `Dialign` and `LocARNA`.

Furthermore, `AnchoRNA` can be used in an iterative manner to search for anchors in "empty" regions of the sequence set where no anchors were identified in the first run, see Fig. S1 in the appendix.

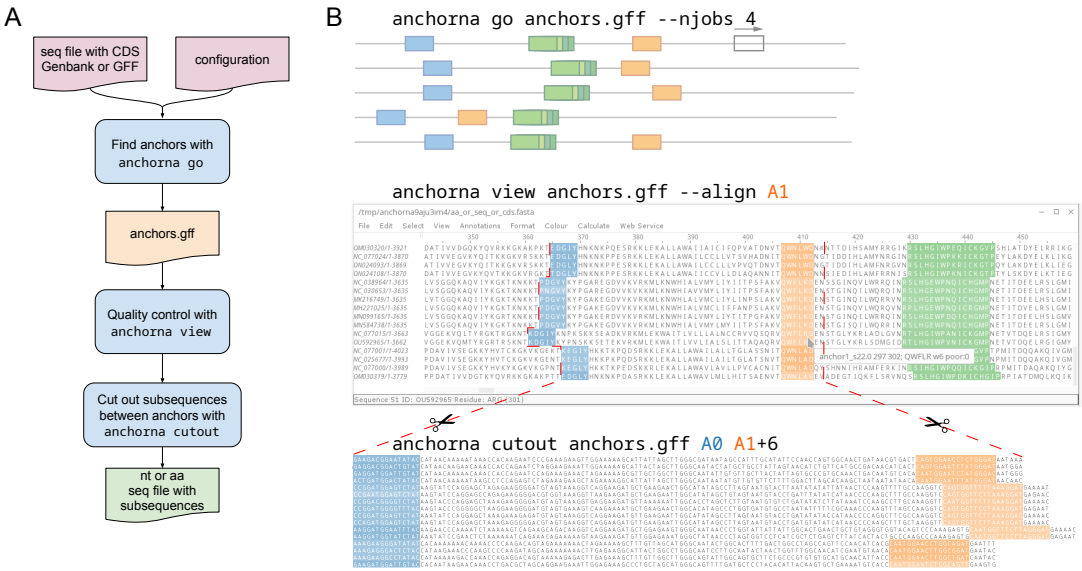

Figure 3: Basic `AnchoRNA` workflow. **A** The `anchorna go` command takes a sequence file and a configuration file and outputs anchors in GFF format. Anchor GFF files can be viewed with `Jalview` (Waterhouse et al., 2009) using the `anchorna view` command. The original or translated sequences can be cut at positions relative to the anchors with the command `anchorna cutout`. A more advanced workflow is shown in Fig. S1 in the appendix. **B** Example usage of `AnchoRNA` commands based on the tutorial dataset. `anchorna go` is used with 4 CPUs; `anchorna view` shows anchors A0 (blue), A1 (orange, aligned), A2 (green); `anchorna cutout` cuts the nucleotide sequences from anchor A0 to anchor A1 and adds 6 nucleotides to the right.

Figure 4: *Pestivirus* genome of around 12 kb length adapted from Viralzone (Hulo et al., 2010) with anchors indicated by black and colored vertical bars. 56 anchors were found using the tutorial dataset of 55 *Pestivirus* sequences. Different alignments of the genome region marked with the horizontal bar and light gray box are shown in Fig. 6. The first four colored anchors in this region are displayed in Fig. 5. The distance between neighboring anchors ranges from 1 aa to around 1100 aa depending on position in the genome and sequence.

Results

Full-genome alignment of pestiviruses.

To demonstrate the performance of `AnchoRNA`, we downloaded all 1,685 *Pestivirus* genomes (as of October 2023) from the BV-BRC and NCBI databases (Olson et al., 2022; Sayers et al., 2021). Using the `ViralClust` pipeline (Lamkiewicz et al., 2025), as described in Triebel et al. (2024, 2025), the dataset was reduced to 55 representative genomes of approx. 12 kb, which are included in the `AnchoRNA` distribution as demonstrative example. These genomes were selected to ensure a well-distributed and balanced representation across the phylogenetic tree of the genus *Pestivirus*. For further applications, we aim to predict RNA secondary structures across the entire RNA genome alignment. The prediction is performed using `LocARNA` ($\mathcal{O}(n^4)$), a heuristic adaptation of the Sankoff algorithm ($\mathcal{O}(n^6)$) (Sankoff, 1985).

Using `AnchoRNA`, we translated all protein-coding regions into amino acids and identified 56 anchors across the entire genome. Each anchor is 5–22 amino acids in length, spaced between 1 and 1100 amino acids apart, see Fig. 4. For a detailed view, we selected 24 sequences (at least one from each genotype) and highlighted anchor no. 11–13, which are easy to align (Fig. 5A), and anchor 14, which only `AnchoRNA` successfully aligned (Fig. 5B), see below. Triebel et al. (2025) used these anchors to split up the *Pestivirus* genome sequences into several subsequences which they fed into the `LocARNA` algorithm. We use their resulting hand-curated *Pestivirus* alignment as ground truth.

Impact of anchors on different alignment tools

We compared the impact of `AnchoRNA` anchors for several tools. We chose `MAFFT` for its popularity and widespread use, and `MAFFT` with `AnchoRNA` to assess the effect of anchor-based refinement. `Dialign` was included because it is well-suited for handling insertions and deletions, while `Dialign` with `CHAOS` has been described to utilizes pairwise anchors to enhance alignment quality. We evaluated `Dialign` with `AnchoRNA` to integrate our anchor-based approach into the `Dialign` workflow. Finally, we also employed `LocARNA` in its standard form, alongside `LocARNA` with `AnchoRNA`, and `LocARNA` applied to subsequences.

The various tools necessitate different adaptations for alignment. Due to the limitations of the `Dialign` web service, which cannot process more than 50 sequences, we reduced the dataset

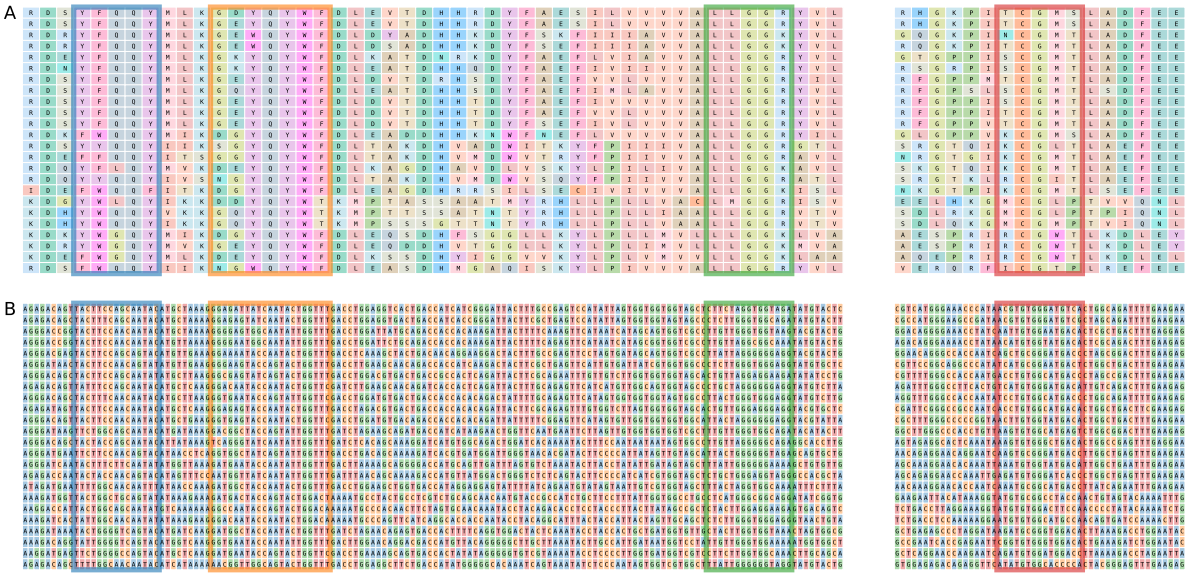

Figure 5: Sequences with four exemplary anchors found with `AnchoRNA` for the *Pestivirus* dataset. For visualization purposes only 24 sequences (each genotype represented) are displayed, with 1,000-1,600 nt omitted (white space). The four anchors correspond to anchor 11-14, which are displayed with the same colors in Fig. 4 and Fig. 6. `AnchoRNA` works on the amino acid level by default (**A**) and the anchors are transferred to the nucleotide level only afterwards (**B**).

accordingly. On this reduced dataset of 50 sequences, `AnchoRNA` identified the same 56 anchors as in the full dataset of 55 sequences. For `AnchoRNA+Dialign`, we used the `anchorna export` command to export the anchors A14 and A41, which were not aligned by vanilla `Dialign`, along with the surrounding anchors, into the `Dialign` anchor file, which serves as a constraint for the `Dialign` alignment process. For `AnchoRNA+MAFFT`, we utilized the `anchorna cutout` command to split the genome sequences into two subsets at anchor A14, as `MAFFT` does not provide an option to directly incorporate anchors. This allowed us to force `MAFFT` to align the anchor, after which we concatenated the resulting alignments. For `AnchoRNA+LocARNA`, we exported all anchors except one erroneous anchor, A8, to the `LocARNA` anchor format using the `anchorna export` command. These anchors were then utilized as constraints for `LocARNA`, enabling more accurate alignment by guiding the alignment process based on the identified anchor points. In another approach, we split the genome at the anchors and fed each subsequence set into `LocARNA`, subsequently concatenating the resulting alignments.

Fig. 6 displays parts of the resulting alignments for the various alignment approaches for *Pestivirus* genomes between `AnchoRNA` anchor no. 11 and 17. The most compact alignment-part, produced by `AnchoRNA` in conjunction with `LocARNA`, measures ≈3,000 nt in length, while the widest alignment, optimized for longer insertions using `vanilla Dialign`, extends to ≈6,980 nt. Notably, while this example highlights a single anchor region at the nucleotide level that is not aligned by standard multiple sequence alignment methods, in more complex real-world scenarios, this issue typically affects numerous such regions. To measure a precise alignment score, we used Todd Lowe's `NUC4.4` scoring system[1], which assigns +5 for identical letters, -4 for different letters, additionally we set -11 for a new gap, and -1 for gap extensions. We calculate the alignment score for each sequence pair, compute the mean of these scores, and divide by the constant median length of the 50 sequences (12 296.5). This results in a mean alignment score per letter, with a perfect score of 5. Tab. 1 compares various alignment methods for the 50 *Pestivirus* genomes, both with and without anchor guidance. Without the use of anchors, `MAFFT` produced a reasonable alignment with a length of 16.1 kb and a mean score of 0.41. In contrast, `Dialign` generated a much longer alignment being gap-rich of 24.9 kb length, resulting in a low mean score of -1.00. `LocARNA` produced the alignment with the highest score of 0.67 and an alignment length of ≈14.3 kb. When anchor guidance by `CHAOS` was introduced, `Dialign` benefits substantially, achieving a score of 0.29 with a more manageable alignment length of 16.9 kb. This

Table 1. Mean alignment scores and alignment lengths (rounded to 100 bases) of the curated *Pestivirus* sequences using different alignment programs with and without guidance by anchors. The scores displayed are mean scores per letter using the `NUC4.4` matrix. A perfect mean alignment score is 5. `CHAOS` calculates anchors for pairs of sequences. It is therefore difficult to combine CHAOS anchors with `MAFFT` and `LocARNA`. NA – not applicable.

	vanilla	anchors with `CHAOS`	anchors with `AnchoRNA`
`MAFFT`	0.41/16100	NA	0.53/15500
`Dialign`	-1.00/24900	0.29/16900	-0.62/22300
`LocARNA`	0.67/14300	NA	0.68/14300
			0.66/14300
hand-curated			0.63/14800
runtime	≈ 2 min	≈ 7 h	≈ 12 h	≈ 6 d

[1] `ftp://ftp.ncbi.nih.gov/blast/matrices/`, last accessed on 21 Dec 2024, also distributed with the `sugar` package

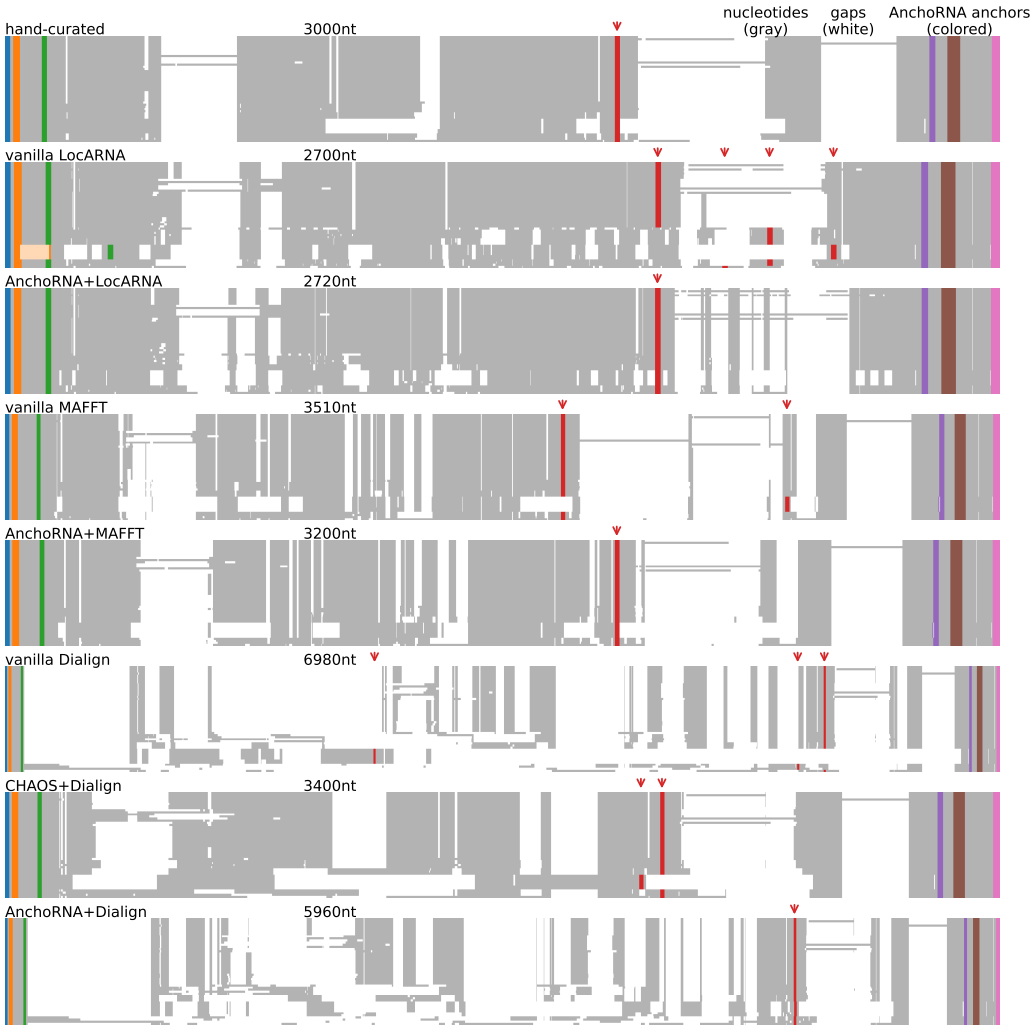

Figure 6: Part of the alignments produced by different tools with and without guidance by anchors. Anchors calculated by `AnchoRNA` are displayed over each alignment with colored bars. The positions of anchor 14 are marked with red arrows. Length of the displayed parts of alignments rounded to 10 nt are indicated. Each panel corresponds to one cell in Tab. 1. The guidance by anchors decreases alignment length for each tool.

result can be attributed to the fact that `CHAOS` calculates anchors specifically for pairs of sequences. As a result, it is challenging to integrate `CHAOS` anchors with alignment tools like `MAFFT` and `LocARNA`.

All alignment methods could be improved by the guidance of `AnchoRNA` anchors. `MAFFT` could be improved to a mean score of 0.53 and an alignment length of 15.5 kb. `Dialign` in combination with `AnchoRNA` led to an improvement with a score of -0.62 and an alignment length of 22.3 kb, but this was not as efficient as the results obtained with `CHAOS` anchors, which use a larger number of pairwise anchors. Finally, we input `AnchoRNA` anchors into `LocARNA`, which resulted in a slightly improved alignment score of 0.68, however, it still had an extensive runtime of several days. Therefore, we also ran `LocARNA` on subsequences based on the calculated anchors, which yielded a slightly lower alignment score of 0.66 and an alignment length of 14.3 kb, but with a more reasonable runtime of 7 h. This result outperformed score-wise the hand-curated 'ground truth' alignment by Triebel et al. (2025), which achieved an alignment score of 0.63 and an alignment length of 14.8 kb. However, the hand-curated alignments are partially based on pure secondary structure conservation, which leads to less optimal sequence alignments. Consequently, while the alignment score is lower, it is considered to more accurately reflect the evolution of the virus family.

Conclusion

The alignment of full viral genomes presents significant challenges due to the complexities associated with long sequences, extensive insertions and deletions, and high sequence divergence. Our study demonstrates that `AnchoRNA` effectively addresses these challenges by identifying conserved regions, or anchors, within the coding sequences, demonstrated at the example of pestiviruses. Utilizing

a dataset of 55 representative genomes, `AnchoRNA` identified 56 anchors that are crucial for guiding the alignment process. The incorporation of these anchors led to significant improvements across all tested alignment tools, highlighting the effectiveness of `AnchoRNA` in enhancing alignment quality, especially in complex viral genomes where standard methods often fall short.

While not the primary focus of our tool, `AnchoRNA` is capable of identifying anchors in both amino acid and non-coding sequences without requiring translation. The runtime complexity of the `AnchoRNA` algorithm is $\mathcal{O}(LSN^2)$, where L represents the length of the reference sequence, S corresponds to the `search_range` parameter plus the maximum of the absolute length differences between the reference sequence and all other sequences, and N denotes the number of sequences analyzed. In practical terms, we found that `AnchoRNA` operates efficiently on a standard PC with four cores, processing the *Pestivirus* dataset of 55 sequences – each approximately 12 kb in length – in about one minute. This rapid processing time makes `AnchoRNA` a viable option for researchers requiring timely results without sacrificing the accuracy of alignment.

Furthermore, the abundance and distribution of anchors identified by `AnchoRNA` represent a significant breakthrough for primer design across entire virus families. The conserved nature of these anchors ensures effective amplification of genetic material, which is crucial for diagnostics, epidemiology, and virology research. Therefore, integrating `AnchoRNA` into alignment and primer design processes not only enhances alignment quality but also makes a meaningful contribution to the broader field of viral research and management.

Acknowledgments

We thank Daria Kubrak for drawing the method scheme. We happily developed the manuscript *in pitu* and therefore thank Niklas Basler from the ITN for his remarks. This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy – EXC 2051 – Project-ID 390713860; NFDI4Microbiota (NFDI 28/1); and the Ministry for Economics, Sciences and Digital Society of Thuringia (TMWWDG), under the framework of the Landesprogramm ProDigital (DigLeben-5575/10-9).

References

Angiuoli SV and Salzberg SL (2010), Mugsy: Fast multiple alignment of closely related whole genomes, *Bioinformatics*, 27(3):334–342, doi:10.1093/bioinformatics/btq665.

Blanchette M, Kent WJ, Riemer C, Elnitski L, Smit AFA, Roskin KM, Baertsch R, Rosenbloom K, Clawson H, Green ED, Haussler D, and Miller W (2004), Aligning multiple genomic sequences with the threaded blockset aligner, *Genome Research*, 14(4):708–715, doi:10.1101/gr.1933104.

Bray N, Dubchak I, and Pachter L (2002), AVID: A global alignment program, *Genome Research*, 13(1):97–102, doi:10.1101/gr.789803.

Brudno M, Chapman M, Göttgens B, Batzoglou S, and Morgenstern B (2003a), Fast and sensitive multiple alignment of large genomic sequences, *BMC Bioinformatics*, 4(1), doi:10.1186/1471-2105-4-66.

Brudno M, Do CB, Cooper GM, Kim MF, Davydov E, Program NISCCS, Green ED, Sidow A, and Batzoglou S (2003b), LAGAN and Multi-LAGAN: Efficient tools for large-scale multiple alignment of genomic DNA, *Genome Research*, 13(4):721–731, doi:10.1101/gr.926603.

Darling ACE, Mau B, Blattner FR, and Perna NT (2004), Mauve: Multiple alignment of conserved genomic sequence with rearrangements, *Genome Research*, 14(7):1394–1403, doi:10.1101/gr.2289704.

Darling AE, Mau B, and Perna NT (2010), progressiveMauve: Multiple genome alignment with gene gain, loss and rearrangement, *PLOS One*, 5(6):e11147, doi:10.1371/journal.pone.0011147.

Delcher AL, Kasif S, Fleischmann RD, Peterson J, White O, and Salzberg SL (1999), Alignment of whole genomes, *Nucleic Acids Research*, 27(11):2369–2376, doi:10.1093/nar/27.11.2369.

Dubchak I, Poliakov A, Kislyuk A, and Brudno M (2009), Multiple whole-genome alignments without a reference organism, *Genome Research*, 19(4):682–689, doi:10.1101/gr.081778.108.

Edgar RC (2004), MUSCLE: Multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Research*, 32(5):1792–1797, doi:10.1093/nar/gkh340.

Eulenfeld T (2024a), AnchoRNA, `https://github.com/rnajena/anchorna`, doi:10.5281/zenodo.12190267.

Eulenfeld T (2024b), Sugar: A Python framework for bioinformatics, `https://github.com/rnajena/sugar`, doi:10.5281/zenodo.11388074.

Henikoff S and Henikoff JG (1992), Amino acid substitution matrices from protein blocks., *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, doi:10.1073/pnas.89.22.10915.

Huang W, Umbach DM, and Li L (2005), Accurate anchoring alignment of divergent sequences, *Bioinformatics*, 22(1):29–34, doi:10.1093/bioinformatics/bti772.

Hufsky F, Abecasis AB, Babaian A, Beck S, Brierley L, Dellicour S, Eggeling C, Elena SF, Gieraths U, Ha AD, Harvey W, Jones TC, Lamkiewicz K, Lovate GL, Lücking D, Machyna M, Nishimura L, Nocke MK, Renard BY, Sakaguchi S, Sakellaridi L, Spangenberg J, Tarradas-Alemany M, Triebel S, Vakulenko Y, Wijesekara RY, González-Candelas F, Krautwurst S, Pérez-Cataluña A, Randazzo W, Sánchez G, and Marz M (2023), The International Virus Bioinformatics Meeting 2023, *Viruses*, 15(10):2031, doi:10.3390/v15102031.

Hufsky F, Lamkiewicz K, Almeida A, Aouacheria A, Arighi C, Bateman A, Baumbach J, Beerenwinkel N, Brandt C, Cacciabue M, Chuguransky S, Drechsel O, Finn RD, Fritz A, Fuchs S, Hattab G, Hauschild AC, Heider D, Hoffmann M, Hölzer M, Hoops S, Kaderali L, Kalvari I, von Kleist M, Kmiecinski R, Kühnert D, Lasso G, Libin P, List M, Löchel HF, Martin MJ, Martin R, Matschinske J, McHardy AC, Mendes P, Mistry J, Navratil V, Nawrocki EP, O'Toole AN, Ontiveros-Palacios N, Petrov AI, Rangel-Pineros G, Redaschi N, Reimering S, Reinert K, Reyes A, Richardson L, Robertson DL, Sadegh S, Singer JB, Theys K, Upton C, Welzel M, Williams L, and Marz M (2020), Computational strategies to combat COVID-19: Useful tools to accelerate SARS-CoV-2 and Coronavirus research, *Briefings in Bioinformatics*, 22(2):642–663, doi:10.1093/bib/bbaa232.

Hulo C, de Castro E, Masson P, Bougueleret L, Bairoch A, Xenarios I, and Le Mercier P (2010), ViralZone: a knowledge resource to understand virus diversity, *Nucleic Acids Research*, 39(suppl_1):D576–D582, doi:10.1093/nar/gkq901.

Hölzer M and Marz M (2020), PoSeiDon: a Nextflow pipeline for the detection of evolutionary recombination events and positive selection, *Bioinformatics*, 37(7):1018–1020, doi:10.1093/bioinformatics/btaa695.

Katoh K (2002), MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform, *Nucleic Acids Research*, 30(14):3059–3066, doi:10.1093/nar/gkf436.

Lamkiewicz K, Esquivel Gomez LR, Kühnert D, and Marz M (2023), Genome structure, life cycle, and taxonomy of Coronaviruses and the evolution of SARS-CoV-2, pages 305–339, Springer International Publishing, doi:10.1007/978-3-031-15640-3_9.

Lamkiewicz K, Triebel S, and Marz M (2025), ViralClust: A Nextflow pipeline to cluster viral sequences, `https://github.`

com/rnajena/viralclust, doi:10.5281/zenodo.14586897.

Leimeister CA, Dencker T, and Morgenstern B (2018), Accurate multiple alignment of distantly related genome sequences using filtered spaced word matches as anchor points, *Bioinformatics*, 35(2):211–218, doi:10.1093/bioinformatics/bty592.

Madhugiri R, Fricke M, Marz M, and Ziebuhr J (2014), RNA structure analysis of alphacoronavirus terminal genome regions, *Virus Research*, 194:76–89, doi:10.1016/j.virusres.2014.10.001.

Morgenstern B (1999), DIALIGN 2: Improvement of the segment-to-segment approach to multiple sequence alignment., *Bioinformatics*, 15(3):211–218, doi:10.1093/bioinformatics/15.3.211.

Morgenstern B, Frech K, Dress A, and Werner T (1998), DIALIGN: Finding local similarities by multiple sequence alignment., *Bioinformatics*, 14(3):290–294, doi:10.1093/bioinformatics/14.3.290.

Notredame C, Higgins DG, and Heringa J (2000), T-Coffee: A novel method for fast and accurate multiple sequence alignment, *Journal of Molecular Biology*, 302(1):205–217, doi:10.1006/jmbi.2000.4042.

Ogurtsov AY, Roytberg MA, Shabalina SA, and Kondrashov AS (2002), OWEN: Aligning long collinear regions of genomes, *Bioinformatics*, 18(12):1703–1704, doi:10.1093/bioinformatics/18.12.1703.

Olson RD, Assaf R, Brettin T, Conrad N, Cucinell C, Davis J, Dempsey D, Dickerman A, Dietrich E, Kenyon R, Kuscuoglu M, Lefkowitz E, Lu J, Machi D, Macken C, Mao C, Niewiadomska A, Nguyen M, Olsen G, Overbeek J, Parrello B, Parrello V, Porter J, Pusch G, Shukla M, Singh I, Stewart L, Tan G, Thomas C, VanOeffelen M, Vonstein V, Wallace Z, Warren A, Wattam A, Xia F, Yoo H, Zhang Y, Zmasek C, Scheuermann R, and Stevens R (2022), Introducing the Bacterial and Viral Bioinformatics Resource Center (BV-BRC): a resource combining PATRIC, IRD and ViPR, *Nucleic Acids Research*, 51(D1):D678–D689, doi:10.1093/nar/gkac1003.

Sankoff D (1985), Simultaneous solution of the RNA folding, alignment and protosequence problems, *SIAM Journal on Applied Mathematics*, 45(5):810–825, doi:10.1137/0145048.

Sayers EW, Bolton EE, Brister JR, Canese K, Chan J, Comeau D, Connor R, Funk K, Kelly C, Kim S, Madej T, Marchler-Bauer A, Lanczycki C, Lathrop S, Lu Z, Thibaud-Nissen F, Murphy T, Phan L, Skripchenko Y, Tse T, Wang J, Williams R, Trawick B, Pruitt K, and Sherry S (2021), Database resources of the national center for biotechnology information, *Nucleic Acids Research*, 50(D1):D20–D26, doi:10.1093/nar/gkab1112.

Thompson JD, Higgins DG, and Gibson TJ (1994), CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucleic Acids Research*, 22(22):4673–4680, doi:10.1093/nar/22.22.4673.

Triebel S, Eulenfeld T, Ontiveros N, Sweeney B, Tautz N, and Marz M (2025), First full-genome alignment representative for the genus Pestivirus, *in preparation*.

Triebel S, Lamkiewicz K, Ontiveros N, Sweeney B, Stadler PF, Petrov AI, Niepmann M, and Marz M (2024), Comprehensive survey of conserved RNA secondary structures in full-genome alignment of Hepatitis C virus, *Scientific Reports*, 14(1), doi:10.1038/s41598-024-62897-0.

Waterhouse AM, Procter JB, Martin DMA, Clamp M, and Barton GJ (2009), Jalview version 2–a multiple sequence alignment editor and analysis workbench, *Bioinformatics*, 25(9):1189–1191, doi:10.1093/bioinformatics/btp033.

Will S, Joshi T, Hofacker IL, Stadler PF, and Backofen R (2012), LocARNA-P: Accurate boundary prediction and improved detection of structural RNAs, *RNA*, 18(5):900–914, doi:10.1261/rna.029041.111.

Appendix

Extended workflow

Fig. S1 shows an extension of the workflow chart in Fig. 3. It includes commands to refine the anchor search in specif regions of the virus genome.

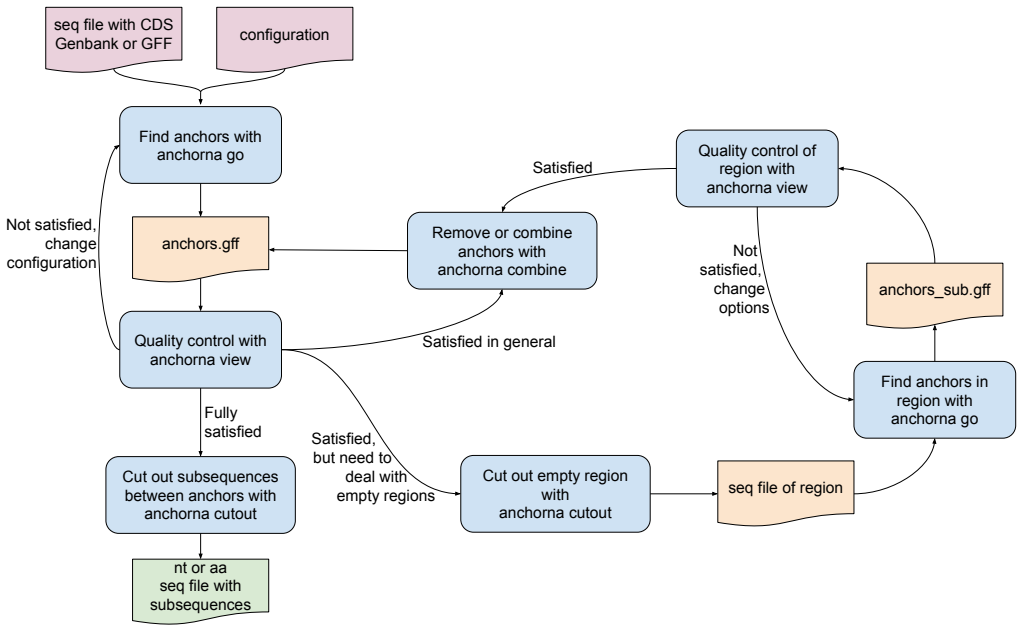

Figure S1: The place of the different `AnchoRNA` commands in the general workflow. The output of several commands, represented by orange data boxes, may be directly passed to other commands using a pipe.