POZNAN UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTING AND TELECOMMUNICATIONS

Institute of Computing Science

DOCTORAL DISSERTATION

Algorithms for feature exploration and modeling of quadruplex structures

Michał Paweł Żurkowski, M.Sc.

Supervisor: **Marta Szachniuk, Prof. Dr Eng.**

Poznań, 2024

Acknowledgements

I want to thank everyone who made the creation of this dissertation possible. It would not have been possible without the support of many people

First and foremost, I wish to express my thanks and gratitude to my supervisor prof. dr hab. inż. Marta Szachiuk. Your guidance and invaluable knowledge provided through every step of my studies allowed me to come this far. I would also like to thank you for your bottomles patience and understanding, I really appreciate it.

Finally, I want to express my love and gratitude to my family and friends. Their endless understanding and support provided through all my life allowed me to come this far. I also wish to especialy thank my brothers, Tomasz and Piotr who have always inspired me. Their unwavering understanding and patience gave me the courage to push forward and become who I am today.

Michał Żurkowski.

Abstract

The subject of this dissertation is derived from bioinformatics, in which biology and computing science converge to address complex challenges of life sciences. As one of the fastest-growing branches of science, bioinformatics plays a crucial role in generating, processing, and analyzing bioscience-related data. This type of data includes quadruplex (G4) structures, the discovery and analysis of which is the focus of this work. Through the development of new computational methods and bioinformatic tools, it contributes to a better understanding of these motifs with implications for the development of bioinformatics, molecular medicine, and biotechnology. The first result of this work is *ONQUADRO*, a self-updating repository dedicated to quadruplexes. This resource retrieves data from the Protein Data Bank (PDB) and, through further analysis and processing, enables exploration of quadruplex structures, including sequences, and the secondary and tertiary structures of tetrads, G4s, and G4 helices. Complementing this database is *WebTetrado*, a web server to analyze structures that were obtained *in silico* or experimentally and have not yet been submitted to the PDB, or structures with simulated modifications. Both tools are consistent in their parametric descriptions of structures, forming a duo that provides a comprehensive analysis of quadruplexes. Another result is *DrawTetrado* that automates the creation of 2.5D layer diagrams, offering optimized visualization to facilitate understanding of G4s and the discovery of their

structural relationships. From my previous work related to improvements with visualizations of 2D structures of RNAs which led to the discovery of a new ONZ classification for quadruplexes, it was of utmost importance to provide that improvement to the most common way of visualizing quadruplexes. *DrawTetrado* is integrated into *ONQUADRO* and *WebTetrado*.

The analysis of structural features and similarities between biomolecules often requires the alignment of 3D structures. Algorithms designed for this task are also used to evaluate structure modeling. However, aligning the motifs with the unusual architecture adopted by quadruplexes presents a significant bioinformatic challenge. Therefore, as part of my doctoral thesis, I developed two proprietary algorithms for flexible alignment of nucleic acid structures, *GEOS* and *GENS*. I then created a system called *RNAhugs*, which enables these algorithms to operate independently and dependently on the sequence, with adjustable RMSD cutoff values. I validated the algorithms through extensive benchmarking against all currently available methods dedicated to 3D structure alignment. These tests confirmed that the new methods can produce longer alignments while maintaining the same or better structural similarity.

The *LinkTetrado* algorithm is the latest achievement in this dissertation. It is the world's first automatic method for identifying multimeric nucleotide assemblies based on G4 structures. It allows the discovery of nucleotides in DNA and RNA molecules that interact with tetrads so that pentads, hexads, heptads, and beyond are formed. The application of *LinkTetrado* expands the catalog of known motifs and allows the analysis of their properties. The algorithm was validated against experimental data from Nuclear Magnetic Resonance (NMR). In analyzing the NMR data, I collaborated with researchers from the Department of Biomolecular NMR, IBCH PAS, and Dr. Maja Marušič from the Slovenian NMR Center.

Streszczenie

Tematyka rozprawy wywodzi się z bioinformatyki, w której biologia i nauki komputerowe łączą się, aby stawić czoła wyzwaniom nauk przyrodniczych. Bioinformatyka, jedna z najszybciej rozwijających się gałęzi nauki, odgrywa kluczową rolę w generowaniu, przetwarzaniu i analizowaniu danych z bionauk. Do tego rodzaju danych należą struktury kwadrupleksów (G4), na których odkrywaniu i analizie skupia się niniejsza praca doktorska. Poprzez opracowanie nowych metod obliczeniowych i narzędzi bioinformatycznych, przyczynia się ona do lepszego zrozumienia tych motywów mając wpływ na rozwój bioinformatyki, medycyny molekularnej i biotechnologii.

Pierwszym wynikiem niniejszej pracy jest *ONQUADRO*, samoaktualizujące się repozytorium dedykowane kwadrupleksom. Pozyskuje ono dane z Protein Data Bank (PDB), a dzięki dalszej analizie i przetwarzaniu umożliwia eksplorację ich struktur, w tym sekwencji oraz struktur drugo- i trzeciorzędowych, tetrad, G4 oraz helis G4. Uzupełnieniem tej bazy danych jest *WebTetrado*, aplikacja do analizy struktur, które zostały otrzymane *in silico* lub eksperymentalnie i nie przesłano ich jeszcze do PDB, lub struktur z symulowanymi modyfikacjami. Oba narzędzia są spójne jeśli chodzi o parametryczny opis struktur tworząc duet zapewniający kompleksową analizę kwadrupleksów. Kolejnym rezultatem jest narzędzie *DrawTetrado*, które automatyzuje tworzenie diagramów warstwowych w grafice 2.5D, oferując wizualizację ułatwiającą zrozumienie G4 i odkrywanie ich relacji

strukturalnych. Na podstawie moich wcześniejszych prac związanych z poprawą wizualizacji struktur 2D RNA, które doprowadziły do odkrycia nowej klasyfikacji kwadrupleksów ONZ, niezwykle istotne było wprowadzenie tej poprawy do najbardziej powszechnego sposobu przedstawiania kwadrupleksów. *DrawTetrado* jest zintegrowane z *ONQUADRO* i *WebTetrado*. Analiza cech strukturalnych i podobieństw między biocząsteczkami często wymaga dopasowania ich struktur 3D. Algorytmy zaprojektowane do tego zadania są też wykorzystywane do oceny modelowania struktur. Jednak dopasowanie motywów o nietypowej architekturze, jaką przyjmują kwadrupleksy, stanowi istotne wyzwanie bioinformatyczne. W związku z tym, w ramach mojej pracy doktorskiej, opracowałem dwa autorskie algorytmy do elastycznego dopasowania struktur kwasów nukleinowych, *GEOS* i *GENS*. Następnie stworzyłem system o nazwie *RNAhugs*, który umożliwia uruchomienie tych algorytmów zarówno zależnie, jak i niezależnie od sekwencji, z możliwością dostosowania wartości odcięcia RMSD. Algorytmy te zostały zweryfikowane przez szeroko zakrojone testy porównawcze z wszystkimi dostępnymi obecnie metodami dedykowanymi dopasowaniu struktur 3D. Testy potwierdziły, że *GEOS* i *GENS* mogą zapewnić dłuższe dopasowania, przy zachowaniu tego samego lub większego podobieństwa strukturalnego. Algorytm *LinkTetrado* to ostatnie z osiągnięć niniejszej pracy. Jest to pierwsza w świecie automatyczna metoda do detekcji motywów multimerycznych opartych na strukturach G4. Pozwala ona na odkrycie w cząsteczkach DNA i RNA nukleotydów wchodzących w interakcje z tetradami w taki sposób, że tworzą się pentady, heksady, heptady, itd. Zastosowanie *LinkTetrado* pozwala rozszerzyć katalog znanych motywów i umożliwia analizę ich właściwości. Algorytm został zweryfikowany w kontekście danych eksperymentalnych z Magnetycznego Rezonansu Jądrowego (MRJ). Przy analizie tych danych współpracowałem z badaczami z Zakładu Biomolekularnego NMR, ICHB PAN oraz z dr Maja Marušič ze Słoweńskiego Centrum NMR.

List of publications

Papers to form the basis of the dissertation:

- [A1.] Tomasz Zok, Natalia Kraszewska, Joanna Miskiewicz, Paulina Pielacinska, **Michal Zurkowski**, Marta Szachniuk (2021) ONQUADRO: a database of experimentally determined quadruplex structures. *Nucleic Acids Research* 50(D1), D253–D258 (doi: 10.1093/nar/gkab1118).

- [A2.] **Michal Zurkowski**, Tomasz Zok, Marta Szachniuk (2022) DrawTetrado to create layer diagrams of G4 structures. *Bioinformatics* 38(15), 3835–3836 (doi: 10.1093/bioinformatics/btac394).

- [A3.] Bartosz Adamczyk, **Michal Zurkowski**, Marta Szachniuk, Tomasz Zok (2023) WebTetrado: a webserver to explore quadruplexes in nucleic acid 3D structures. *Nucleic Acids Research* 51(W1), W607–W612 (doi: 10.1093/nar/gkad346).

- [A4.] **Michal Zurkowski**, Maciej Antczak, Marta Szachniuk (2023) High-quality, customizable heuristics for RNA 3D structure alignment. *Bioinformatics* 39(5), btad315 (doi: 10.1093/bioinformatics/btad315).

- [A5.] **Michal Zurkowski**, Mateusz Swiercz, Filip Wozny, Maciej Antczak, Marta Szachniuk (2024) RNAhugs web server for customized 3D RNA structure alignment. *Nucleic Acids Research* 52(W1), W348–W353 (doi: 10.1093/nar/gkae259).

Table 1: Bibliometric parameters.

Article ID	PY[1]	IF (PY[1])	5-IF (2024)	MEiN[2] (PY[1])	MEiN[2] (2024)	Quartile (WoS[3])	Rank (WoS[3])
A1	2021	19.2	16.1	200	200	Q1	6/313
A2	2022	5.8	7.6	200	200	Q1	15/174
A3	2023	16.6	16.1	200	200	Q1	6/313
A4	2023	4.4	7.6	200	200	Q1	15/174
A5	2024	16.6	16.1	200	200	Q1	6/313
Total		62.6	63.5	1000	1000	n/a	n/a

Table 2: Number of citations and H-index.

Article ID	Web of Science (all citations)	Web of Science (without self-citations)	Scopus	Google Scholar
A1	16	13	17	23
A2	1	0	2	3
A3	2	2	3	4
A4	2	1	2	3
A5	1	1	0	1
Total	22	17	24	34
H-index(A)	2	2	2	3
H-index(B)	3	3	3	4

The H-index(A) was calculated based solely on publications A1-A5, whereas H-index(B) represents the actual index encompassing all publications.

All journals were qualified in the discipline of *Information and Communication Technology* by the MEiN[2]. The rank of the journal (Table 1) is given for computational biology and bioinformatics, if possible, otherwise for the multidisciplinary area.

[1] Publication Year

[2] The Ministry of National Education (Poland)

[3] Web of Science

Contribution to publications

I declare the following contributions to the articles supporting my doctoral dissertation:

A1. I designed key components of the *ONQUADRO* database system's web application, enabling the visualization of canonical and non-canonical quadruplex structures in nucleic acids. I developed and implemented functions for annotating non-canonical base pairs in tetrad structures, facilitating tetrad topology classification according to the ONZ nomenclature. I contributed to the implementation of the statistical module and conducted system testing. Finally, I contributed to the manuscript preparation and revision.

A2. I designed the world's first algorithm for automatically creating 2.5D layer diagrams of quadruplexes, optimizing their view based on the number of DNA/RNA strand crossings. I tested the algorithm on available quadruplex structures, adjusting it to visualize all loop types. I implemented it in the *DrawTetrado* application with parameterized input for user-customized diagrams. The program saves layer representations as vector graphics for high-quality scientific publications. I integrated *DrawTetrado* into the *ONQUADRO* database [A1] and *WebTetrado* web application [A3]. I wrote the initial article draft, prepared figures, and contributed to the manuscript revision.

A3. I contributed to the development of the *WebTetrado* analytical system, which automates the identification, classification, and structural analysis of DNA and RNA tetrads, quadruplexes, and G4-helices. I was responsible for the development of the *WebTetrado* backend and contributed to the development of the frontend. I collected structural data for testing and validated the system using a dataset of over 1,900 tetrads, 600 quadruplexes, and 30 helices, ensuring the accuracy of the results. I contributed to the manuscript preparation and revision. I generated the accompanying figures.

A4. I developed and implemented two innovative algorithms, *GEOS* and *GENS*, for the flexible alignment of 3D RNA and DNA structures. These algorithms optimize superimposed structure fragments based on an objective function using the RMSD metric with a focus on minimizing expected similarity. I gathered and installed all state-of-the-art applications for structural alignment, configuring them for easy use. I benchmarked *GEOS* and *GENS* against these algorithms using a dataset of 1,000 structures, both with and without quadruplexes. The algorithms were parameterized to operate in two modes, independent and dependent on a sequence. I was responsible for preparing all graphics and tables presented in the paper, including performance and comparative tests. I drafted the first version of the article. I contributed to the manuscript revision and performed further analyses based on reviewer suggestions.

A5. I designed the *RNAhugs* analytical system for 3D structure alignment of RNA and DNA molecules. I supervised two students in implementing the web server, ensuring the application's correctness and security. I prepared the test dataset for benchmarking and conducted extensive testing and validation of the final system. I contributed to the article by writing the first draft and preparing most of the figures, and I participated in the manuscript revision.

Michał Żurkowski

Contents

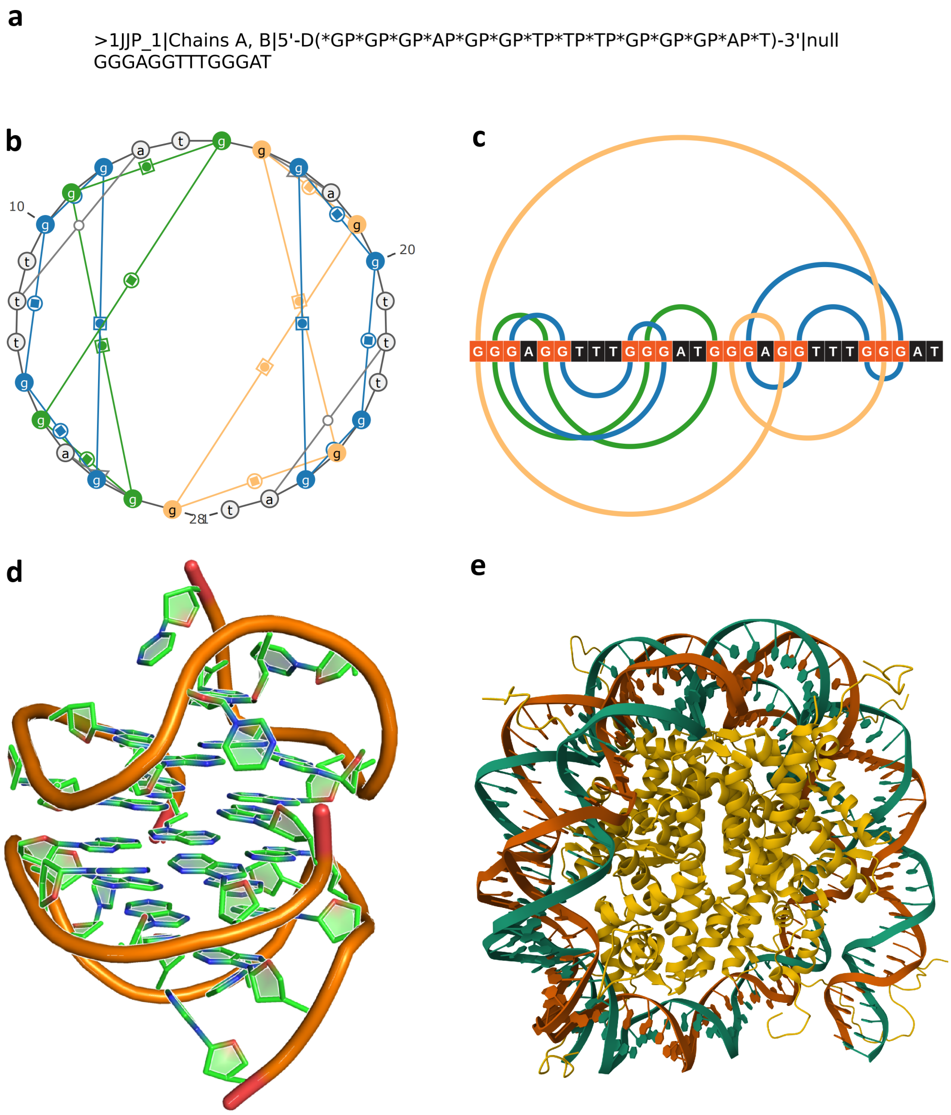
CHAPTER 1

Introduction

1.1. Nucleic acid structure

Nucleic acids play various functions that are highly dependent on their structures. RNA molecules are essential players in a wide array of biological processes, including the regulation of gene expression and catalysis of chemical reactions [Berg (2002)]. For example, RNA molecules like mRNA, miRNA, and lncRNA are integral to controlling the flow of genetic information within cells, influencing everything from development to cellular responses to external stimuli. Additionally, certain RNA molecules, known as ribosomes, can catalyze biochemical reactions, further demonstrating RNA's versatility beyond its traditional role as a messenger molecule. Understanding DNA structure is crucial, as it directly influences gene expression and genome maintenance. Insights into DNA structure aid in developing therapies like gene therapy, offering potential treatments for genetic disorders and cancer. The study of DNA remains central to genetics, molecular biology, and biotechnology, driving progress in medical research and healthcare. In the context of viruses, RNA serves as the primary genetic material for many pathogens, including those responsible for significant human diseases such as rabies [Albertini et al. (2007)], polio [Kitamura et al. (1981)], and the recent COVID-19 pandemic caused by SARS-CoV-2 [Hu et al. (2020)]. The rapid replication and mutation rates of RNA viruses contribute to their pathogenicity and the challenges in developing effective treatments and vaccines. Moreover, the abnormal accumulation of noncoding RNA repeats has been linked to the onset of neurodegenerative diseases such as amyotrophic lateral sclerosis (ALS) and Huntington's disease-like 2 (HDL-2) [Swinnen et al. (2019)]. These noncoding RNAs can form toxic aggregates that disrupt normal cellular functions, leading to disease progression. Given RNA and DNA's critical role in both health and diseases, understanding its structure is crucial. The conformation of RNA is intimately tied to its function [Doudna & Cech (2002)], and insights into these structures can drive the development of nucleic acid-based based therapies. Such advancements hold particular promise for treating complex disorders, including neurodegenerative diseases, where traditional therapeutic approaches have often fallen short.

The structure of nucleic acid, DNA or RNA, can be considered at various organizational levels, ranging from the basic building blocks, nucleotides, to the complex three-dimensional (3D) forms these molecules adopt in living organisms. Understanding this structure is crucial as it underpins the molecule's ability to store and transmit genetic information, interact with proteins, and perform various cellular functions. Biomolecule architecture is generally divided into four levels: primary, secondary, tertiary, and quaternary structure (Figure 1.1). For nucleic acids, the primary structure refers to the sequence of nucleotides linked by the sugar-phosphate backbone. This sequence is typically stored in FASTA-format files, which include a header with a description of the molecule, followed by lines of sequence data [Lipman & Pearson (1985)].

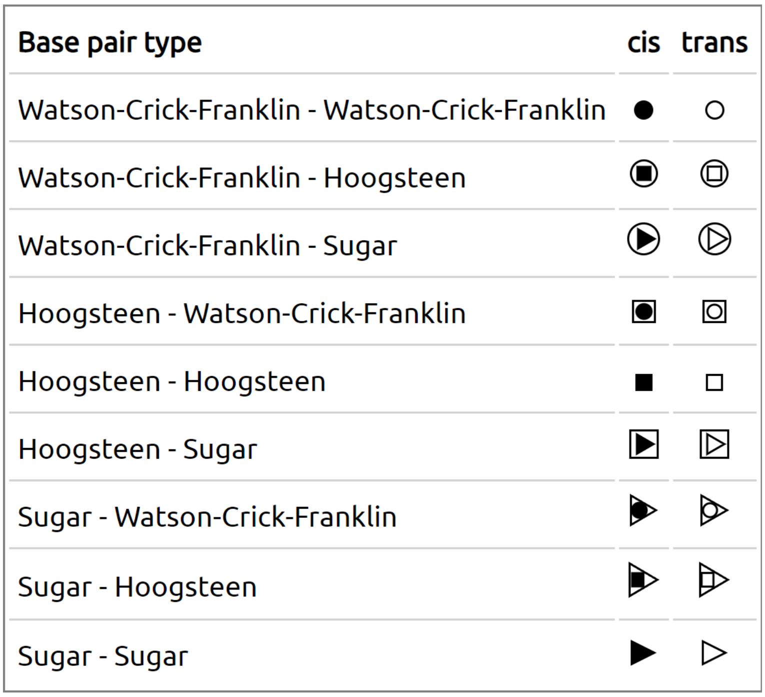


```
>1JJP_1|Chains A, B|5'-D(*GP*GP*GP*AP*GP*GP*TP*TP*TP*GP*GP*GP*AP*T)-3'|null
GGGAGGTTTGGGAT
```

Figure 1.1: Nucleic acid organization levels shown for example molecules: (a) sequence, (b) 2D structure by VARNA (classic diagram), (c) 2D structure by R-chie (arc diagram), (d) 3D structure by PyMOL (cartoon model) and (e) 4D structure by Mol* (ribbon model).

The secondary (2D) structure is defined by nucleobase pairings, including both canonical [Halder & Bhattacharyya (2013); Šponer et al. (2005)] and non-canonical [Leontis & Westhof (2001); Hoehndorf et al. (2011)] interactions. Canonical base pairs, such as A-T and G-C in DNA or A-U and G-C in RNA, form core nucleic acid with two- (A-T, A-U) or three (G-C) hydrogen bonds [Watson & Crick (1974); Halder & Bhattacharyya (2013); Šponer et al. (2005)]. The G-U wobble base pair, which forms two hydrogen bonds, is also considered canonical [Varani & McClain (2000)]. Although less common, non-canonical base pairs are crucial for the proper functioning of RNA molecules.

Many non-canonical base pairs form in RNA structures [Leontis & Westhof (2001); Hoehndorf et al. (2011)]. Leontis and Westhof classified them into twelve distinct families (sixteen if unique types are considered) of edge-to-edge interactions (Figure 1.2). This classification is based on the types of interacting edges and the orientation of the glycosidic bond (cis or trans). Each nucleotide can interact along one of three edges: Watson-Crick-Franklin (W), Hoogsteen (H), or Sugar edge (S) [Leontis & Westhof (2001)]. A detailed description of the 2D structure includes both a list of these base pairs and their classification according to this nomenclature.

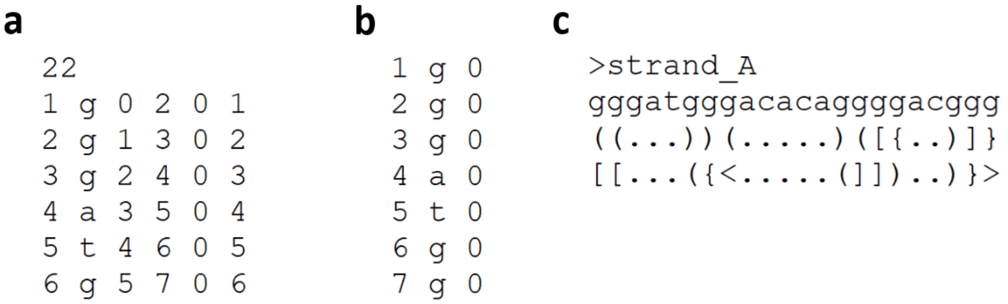


Base pair type	cis	trans
Watson-Crick-Franklin - Watson-Crick-Franklin	●	○
Watson-Crick-Franklin - Hoogsteen	■⃝	□⃝
Watson-Crick-Franklin - Sugar	▶⃝	▷⃝
Hoogsteen - Watson-Crick-Franklin	●⃞	○⃞
Hoogsteen - Hoogsteen	■	□
Hoogsteen - Sugar	▶⃞	▷⃞
Sugar - Watson-Crick-Franklin	●⃤	○⃤
Sugar - Hoogsteen	■⃤	□⃤
Sugar - Sugar	▶	▷

Figure 1.2: Base pair types according to Leontis-Westhof classification and pictograms to represent them in the secondary structure diagrams.

Secondary structure information is typically stored in one of the three formats: Connectivity Table (CT), BPSEQ (Base-Pairs & SEQuence), or Dot-Bracket (also known as parenthesis notation) [Ponty & Leclerc (2014)] (Figure 1.3). The CT format includes a header with the number of nucleotides and the name of the structure, followed by records describing each nucleotide. Each record in the CT format contains six values: the nucleobase number index (*bni*), the one-letter nucleobase code (A, U, G, C) for RNA or (A, T, G, C) for DNA, $bni - 1$, $bni + 1$, the index of the paired nucleobase (or 0 if unpaired), and the original number of a nucleotide (might be other than *bni*). In the BPSEQ format, nucleotide information is described using three columns: the first column indicates the nucleotide position in the sequence (starting from one), the second column shows the one-letter nucleobase code, and the third column displays the index of the paired nucleobase (or 0 if the residue is unpaired). The dot-bracket notation file consists of three lines: a header line, the sequence line, and the secondary structure line. The secondary structure, encoded in dot-bracket notation, is represented as a linear string of dots and brackets (and sometimes lowercase and uppercase letters corresponding to higher levels of brackets if necessary, such as for high-order pseudoknots). The length of this string matches the RNA sequence length, where dots indicate unpaired residues and various brackets denote paired residues, including pseudoknots. A single string of dots and parentheses is sufficient to encode a secondary structure in dot-bracket notation when each nucleotide forms at most one pairing. However, this notation can be inadequate for encoding the 2D structure of certain motifs, such as quadruplexes or tetrads, which are formed by multiples — nucleotides with more than one pairing partner. In a tetrad, each nucleotide pairs with two others. To address this, a two-line dot-bracket notation was defined specifically for these types of secondary structure motifs [Popenda et al. (2019)]. In the two-line dot-bracket notation, each line represents interactions that do not share nucleobases.

The secondary structure of nucleic acids can be visualized in various ways, with the classic diagram and arc diagram being among the most popular.



```
22
1 g 0 2 0 1
2 g 1 3 0 2
3 g 2 4 0 3
4 a 3 5 0 4
5 t 4 6 0 5
6 g 5 7 0 6
```

```
1 g 0
2 g 0
3 g 0
4 a 0
5 t 0
6 g 0
7 g 0
```

```
>strand_A
gggatgggacacaggggacggg
((...))(.....)([{..)]}
[[...({<.....(]])..)}>
```

Figure 1.3: Example formats to store the secondary structure: (a) CT, (b) BPSEQ, and (c) two-line dot-bracket notation.

The classic diagram is often generated using tools such as VARNA [Darty et al. (2009)] or PseudoViewer [Byun & Han (2009)]. Alternatively, the arc diagram provides a different view where arcs correspond to the dot-bracket encoding, and the backbone is depicted as a straight line. Both diagrams are effective for analyzing pseudoknots and non-canonical base pairs.

The tertiary (3D) structure of a nucleic acid molecule represents its three-dimensional spatial arrangement, stabilized by interactions such as ion binding and hydrogen bonds. This 3D conformation is influenced by factors including the nucleotide sequence and environmental conditions [Eric & Pascal (2006); Hoehndorf et al. (2011); Zemora & Waldsich (2010)]. Both RNA and DNA tertiary structures feature specific motifs and structural elements that are critical for the molecule's overall 3D folding. These motifs are often conserved across different RNA species and play essential roles in the stability, function, and molecular interactions. Quadruplexes, for instance, are motifs of significant functional importance.

The quaternary (4D) structure represents the highest level of structure organization and involves interactions between multiple molecules, leading to the formation of complexes such as dimers, trimers, or larger assemblies composed of structures of the same or different types. These interactions encompass RNA-RNA interactions as well as RNA-protein and RNA-ligand interactions, which are crucial for assembling and functioning larger molecular machines, such as the ribosome or spliceosome. In the context of DNA, quaternary structure refers to the binding of DNA to histones to form nucleosomes, which are further organized into higher-order chromatin fibers. DNA quaternary structure is dynamic, varying over time as regions of DNA become condensed or exposed for transcription. Understanding nucleic acid quaternary structure is essential for comprehending DNA, and RNA, as it can significantly impact molecular function.

Both 3D and 4D structures are primarily stored in mmCIF (Macromolecular Crystallographic Information File) format [Bourne et al. (1997)]. The mmCIF format, which succeeded the older PDB (Protein Data Bank) format [Bernstein et al. (1977)], is more versatile and capable of handling the complex data associated with large molecular assemblies. The mmCIF files not only store 3D atomic coordinates of the molecules (algebraic representation of the structure) but also include extensive metadata.

1.2. Quadruplex motifs

Quadruplexes are four-stranded structures found in both DNA and RNA molecules, playing critical roles in essential genetic processes such as transcription, replication, and epigenetic regulation [Rhodes & Lipps (2015); Varshney et al. (2020)]. These structures are typically formed in guanine-rich sequences by the stacking of tetrads, i.e., nucleotide quartets in planar arrangements (Figure 1.4), stabilized by Hoogsteen hydrogen bonds and the presence of monovalent cations like potassium or sodium, which reside at the center of each quartet. Because they are primarily composed of guanine (G), these structures are commonly called G4s. However, it’s important to note that quadruplexes can also be composed of other nucleotides.

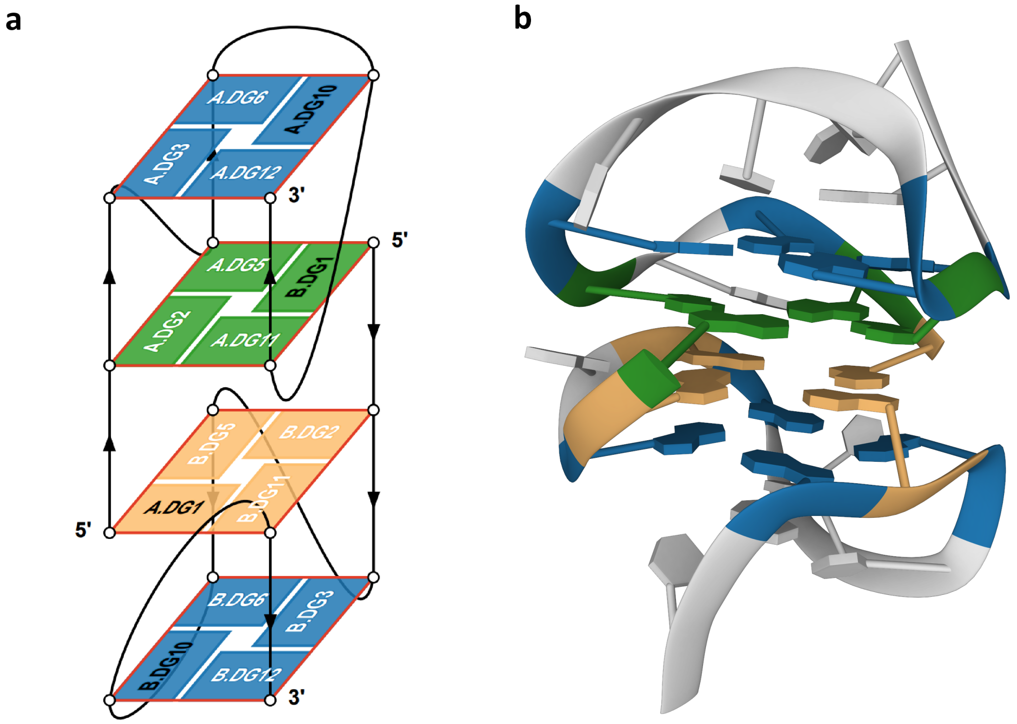


Figure 1.4: DNA structure (PDB ID: 1JJP) with color-coded quadruplex shown as (a) layer diagram created by *DrawTetrado* and (b) cartoon model generated in Mol*.

G4s have been implicated in the development of cancer and neurodegenerative diseases [Plavec (2020); Spiegel et al. (2020)]. Research suggests that these motifs are surprisingly common in the human genome, with potentially hundreds of thousands of occurrences. They are frequently located in telomeric regions and promoter areas of specific genes, where they contribute to telomere extension and gene expression. In telomeres, DNA quadruplexes help maintain chromosome stability and are considered potential targets for cancer therapeutics, as stabilizing these structures can inhibit telomerase activity. In promoter regions, quadruplexes can regulate gene expression by either inhibiting or enhancing the binding of transcription factors. In RNA, quadruplexes often appear in regulatory regions of mRNA, particularly the $5'$ untranslated region ($5'$ UTR), affecting mRNA stability and translation. Despite their prevalence and significant roles, the detailed properties and functions of quadruplexes remain elusive, largely due to the intricate complexity of their structures. This complexity underscores the importance of G4s in genetic regulation and stability, highlighting the need for continued research to understand their biological implications and potential therapeutic uses. The global scientific community is actively searching for quadruplexes in nucleic acid molecules and developing nomenclatures to support classifying quadruplexes by their features.

Several key features distinguish G4s and contribute to their biological functions. One of the fundamental aspects is the number of stacked tetrads that form the quadruplex, with the structure's stability and function often depending on this number. The planarity of each tetrad can be influenced by the nucleotide sequence and the presence of stabilizing cations, such as potassium or sodium ions, which are crucial for maintaining the structural integrity of the quadruplex. Quadruplexes can also be categorized based on the glycosidic bond angles of the intervening nucleotides, which create either anti or syn conformation [Webba da Silva (2007); Webba da Silva et al. (2009)]. This angle, commonly referred to as the Chi angle, influences the overall geometry of the quadruplex and can affect how the structure interacts with other molecules. Another critical feature of quadruplexes is the twist parameter, which is the angular rotation between two stacked tetrads. On the other hand, the rise parameter determines the distance between neighboring tetrads. These two parameters are defined for every pair of neighboring tetrads in a quadruplex. They influence the helical structure, compactness, and stability of the whole G4.

The interaction within each tetrad can be further classified based on the base-pairing patterns formed between nucleotides and the strand directionality (from the $5'$- to $3'$-end). This is known as the ONZ classification (Figure 1.5) and was named for the shapes of the formed connections [Popenda et al. (2019)]. The ONZ classification is defined for tetrads, while the corresponding ONZM classification is for quadruplexes. The latter introduces the "M" (mixed) type for quadruplexes containing mixed stacks of tetrads. The classes have an added attribute, based on the polarity of interactions within the tetrad. ONZ and ONZM nomenclature resulted indirectly from my former research on improving the secondary structure diagrams generated in the *RNApdbee* system [P1].

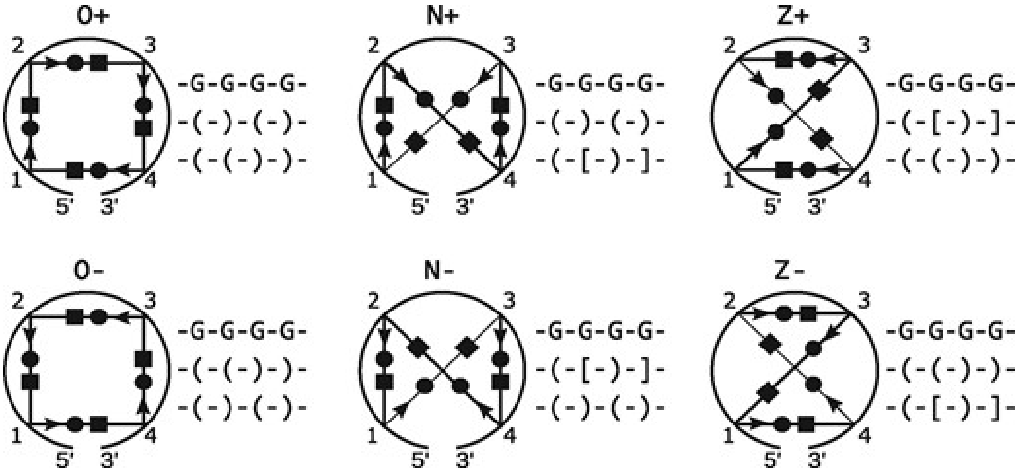


Figure 1.5: ONZ classes defined for tetrads: O+, O-, N+, N-, Z+, and Z-. Black circles denote Watson-Crick-Franklin (WCF) edges and squares represent Hoogsteen (H) edges. Arrows indicate the WCF-to-H direction.

A quadruplex can be classified as parallel, antiparallel, or hybrid, depending on the orientation of its G-tracts. A G-tract refers to a continuous sequence of guanines within a DNA or RNA strand that participates in the formation of a quadruplex. If all G-tracts in a quadruplex have the same orientation, the quadruplex is classified as parallel. If two G-tracts run in one direction and the other two in the opposite direction, the quadruplex is considered antiparallel. And finally, a hybrid quadruplex has one G-tract running in the opposite direction of the other three [Esposito et al. (2007)] (Figure 1.6). Another significant feature of a quadruplex is the number of strands involved in its formation: it can be unimolecular (single strand), bimolecular (two strands), or tetramolecular (four strands). The strands often fold into loops that occur either between the tetrads or externally. Various types of loops can form: lateral (side) loops connect two adjacent, antiparallel

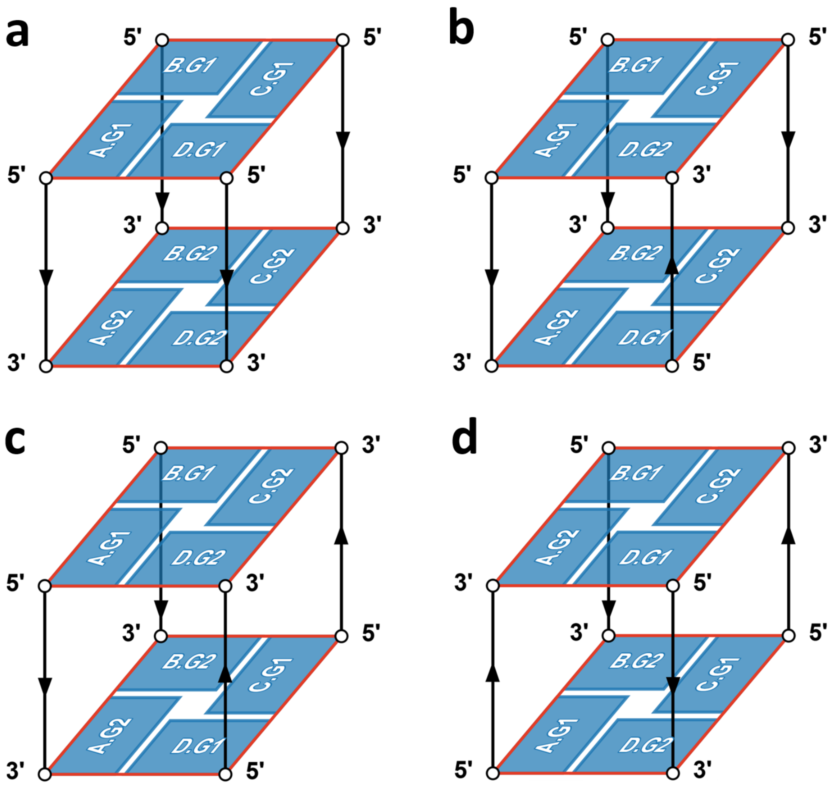


Figure 1.6: Quadruplex classification by relative orientation of G-tracts: (a) parallel, (b) hybrid, (c) antiparallel (top-top-down-down), (d) antiparallel (top-down-top-down).

strands; diagonal loops connect two opposite, antiparallel strands; propeller (reversed) loops connect two adjacent, parallel strands; and V-shaped loops connect two vertices of adjacent tetrads where one guanosine residue lacks phosphodiester chain support [Dvorkin et al. (2018)]. The length and sequence of these loops can significantly influence the stability and flexibility of the quadruplex, with shorter loops generally contributing to greater stability and longer loops providing more flexibility but potentially reducing stability. The topology of loops, strands, and types of tetrads contributes to the classification of quadruplex structures proposed by Webba da Silva [Webba da Silva (2007)].

Understanding quadruplex structures is crucial for therapeutic interventions, particularly in cancer and neurodegenerative diseases. Advanced computational tools and algorithms that efficiently identify, analyze, and visualize quadruplexes while describing their features are essential for expanding our knowledge of these structures. The arsenal of available tools for quadruplex analysis is relatively small but continually expanding. It includes tools like ElTetrado [Zok et al. (2020)] and DSSR [Lu et al. (2015)] that identify G4s in the all-atom 3D structures; visualizers such as DSSR-PyMOL [Lu (2020)] for 3D visualization, and *RNApdbee* [P1] and ElTetrado [Zok et al. (2020)] for 2D visualization; and tools for predicting G4 locations within sequences. The latter subgroup is the largest, including tools like G4Hunter [Bedrat et al. (2016); Brázda et al. (2019)], G4RNA Screener [Garant et al. (2017, 2018)], and QGRS Mapper [Kikin et al. (2006)], among others. There are also databases and data aggregators dedicated to quadruplex 3D structures, such as DSSR-G4DB [Lu et al. (2015); Lu (2020)], and sequence databases like G4RNA [Garant et al. (2015)]. Again, the group of resources collecting experimentally confirmed or putative quadruplex sequences is the largest. My work has further expanded the set of available tools with *ONQUADRO* [A1], a comprehensive database focused on quadruplex structures; *WebTetrado* [A3], a web server dedicated to analyzing structures containing quadruplexes; and *DrawTetrado* [A2], a visualizer for creating 2.5D layer diagrams of quadruplex structures.

1.3. Modeling of nucleic acid structure

Computational prediction (modeling) of RNA/DNA tertiary structures based on nucleotide sequences is one of the major challenges in modern structural bioinformatics [Parisien & Major (2008); Leontis & Westhof (2012); Miao et al. (2020); Townshend et al. (2021)]. While we have extensive knowledge of nucleotide sequences — thanks in part to increasingly widespread sequencing technologies — experimental methods for determining 3D structures are too expensive and have too many limitations to keep pace with sequence acquisition. Therefore, only computational modeling can bridge the gap between our understanding of sequences and our knowledge of structures. The latter is crucial, as the specific fold of DNA or RNA determines how these molecules interact with proteins, small molecules, and other nucleic acids, which is key to understanding their function [Westhof et al. (2011)].

According to Anfinsen's dogma, sequences with high similarity to known structures should fold into similar 3D shapes. This principle, originally formulated based on studies of protein structures, initiated efforts to develop fast, inexpensive, and accurate computational methods to predict protein 3D folds and, later, nucleic acid 3D structures. However, for nucleic acids (and likely some proteins), the relationship between sequence and structure is not always straightforward, as seen particularly in structures containing quadruplexes. Moreover, in many cases, even a single nucleotide mutation can significantly alter the 3D structure [Wiedemann & Miłostan (2017)], while in other cases, the structure remains preserved despite sequence changes [Hoehndorf et al. (2011)]. This variability, combined with the inherent uncertainty in experimentally determined structures, increases the complexity of accurately predicting the 3D structures of nucleic acids. In order to break the long-standing deadlock in RNA 3D structure prediction, the RNA-Puzzles competition was launched in 2011 [Cruz et al. (2012); Miao et al. (2017, 2020)]. The goal of this collaborative initiative is to motivate and inspire the RNA community to improve computational methods for predicting RNA 3D structures. This initiative is analogous to the CASP (Critical Assessment of Structure Prediction) project, which has been ongoing since the 1990s and initially focused on protein structure prediction [Kryshtafovych et al. (2019)], but recently introduced an

RNA prediction category. Both CASP and RNA-Puzzles provide valuable platforms for researchers to refine their predictive algorithms through structured challenges. Over time, the models submitted to these competitions have increasingly resembled native structures, demonstrating that structural bioinformatics can effectively address complex structural questions. In recent years, 3D structure prediction has also attracted the interest of technology companies, such as Google, which launched the AlphaFold system based on deep learning [Jumper et al. (2021); Abramson et al. (2024)] and used it to predict protein targets in CASP. Tools like AlphaFold underscore the importance of developing accurate and reliable 3D structure prediction methods.

As new RNA structure prediction approaches continue to emerge, there is an increasing need for reliable and effective metrics to assess the quality of these predictions. In both CASP and RNA-Puzzles, the evaluation of predicted models is conducted by comparing them to experimentally determined target structures. RNA-Puzzles employs several evaluation measures, including Root-Mean-Square Deviation (RMSD) [Kabsch (1978); Maiorov & Crippen (1994)], Interaction Network Fidelity (INF), Deformation Index (DI) [Parisien et al. (2009)], Mean of Circular Quantities (MCQ) [Zok et al. (2013)], TM-score [Gong et al. (2019)], and Clash score [Davis et al. (2007)]. Of these, the Clash score is the only measure that does not require a reference structure. RMSD and TM-score necessitate structural alignment, whereas the other measures are independent of superimposition. Superimposition [Kabsch (1978); Kneller (1991); Horn (1987); Coutsias et al. (2004)] refers to the process of aligning one object with another to highlight their similarities or differences (Figure 1.7). In the context of molecular 3D structures, which are represented as sets of atoms in 3D space, the Kabsch algorithm [Kabsch (1978)] is commonly used to achieve this alignment. This algorithm calculates the optimal translation and rotation

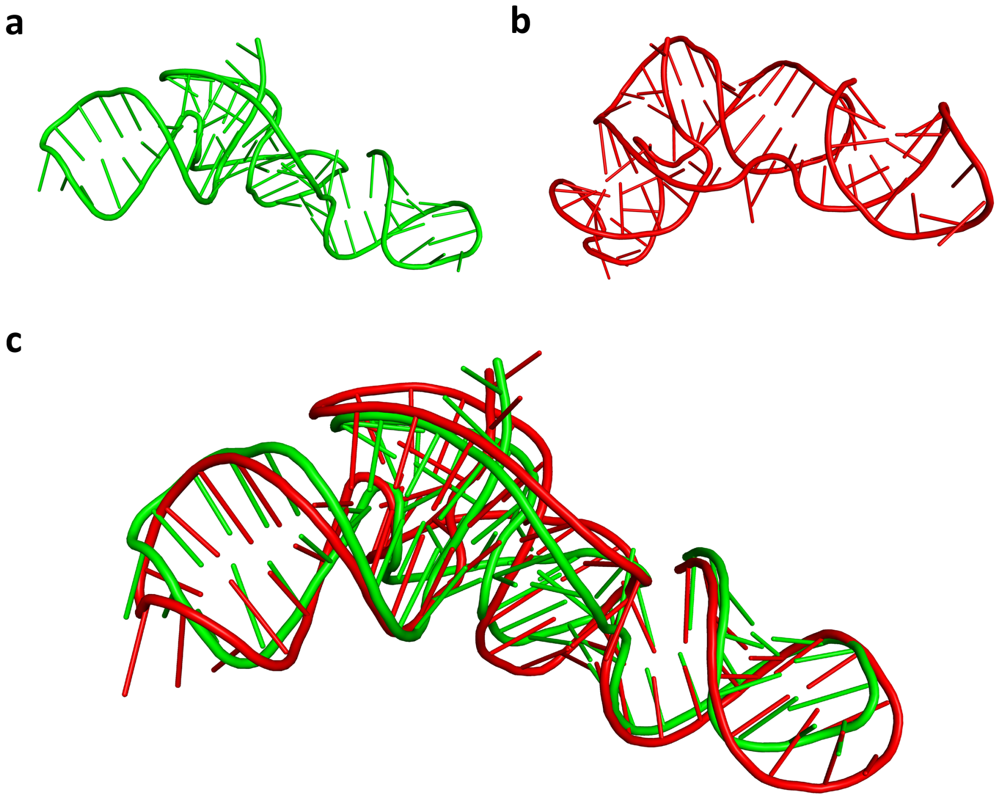


Figure 1.7: (a) Target structure (PDB ID: 5TPY) and (b) its prediction from RNA-Puzzles (Puzzle 18); (c) same structures aligned onto each other.

matrix that minimizes the root mean square deviation (RMSD) between paired sets of points (atoms) in these structures. RMSD measures the distance between points in multi-dimensional space and serves as a distance metric in structural bioinformatics. It quantifies the average structural deviation observed when comparing the 3D structures of two molecules. Lower RMSD values indicate a better match between the structures, while higher values suggest greater discrepancies. Although RMSD is influenced by the number of atoms involved, it remains a valuable tool for assessing the similarity of compared structures. RMSD is computed as follows

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\delta_i^2} \tag{1.1}$$

where δ_i is the distance between the i-th pair of atoms in both structures computed usually as Euclidean distance (Equation 1.2):

$$\begin{aligned}\delta\left(v,w\right) &= \|v_i - w_i\| \\ &= \sqrt{(v_{ix} - w_{ix})^2 + (v_{iy} - w_{iy})^2 + (v_{iz} - w_{iz})^2}\end{aligned} \tag{1.2}$$

1.4. Fundamental concepts in algorithm theory

Solving biological problems often requires complex calculations, detailed analyses, or the processing of large data volumes. Modern computational biology and bioinformatics successfully address these challenges in most cases [Gauthier et al. (2018); Hagen (2000); Ouzounis & Valencia (2003)]. The primary objective of these fields is to model and solve complex biologically inspired problems by developing and adapting techniques from computer science for biological applications. Numerous outstanding algorithms have advanced the biological sciences. For example, one of the earliest and most well-known bioinformatics algorithms, developed by Saul Needleman and Christian Wunsch in 1970, employs dynamic programming to align biological sequences [Needleman & Wunsch (1970)]. In contrast, one of the most recent breakthroughs in bioinformatics is AlphaFold, which has revolutionized the study of protein structures by accurately predicting the three-dimensional shape of any protein [Jumper et al. (2021); Abramson et al. (2024)]; its newest version also deals with macromolecular complexes. However, the development of an effective computational method must be preceded by a thorough analysis of the problem and a clear understanding of the type of problem being addressed. This is crucial as it determines the choice of methodology, which is then used in algorithm design.

From the computing science perspective, a combinatorial problem is a finite collection of objects associated with integer-type parameters. Combinatorial problems are categorized into the following classes:

- **Decision problems**: A problem of this type determines whether a specific condition is met, with only two possible outcomes (binary solution): a positive answer (YES) or a negative answer (NO). An example of a decision problem is determining whether a given graph is Hamiltonian.
- **Search problems**: The goal in such a problem is to find a specific solution, which is not a binary answer, for example locating a Hamiltonian path in a given graph.
- **Optimization problems**: Problems of this type fall under search problems, where the goal is to find an optimal solution that maximizes or minimizes the objective function. For example, finding a Hamiltonian path that minimizes the total travel cost is an optimization problem.

Combinatorial problems vary in difficulty, which influences the choice of algorithmic techniques used to solve them. The difficulty level is determined by the problem's complexity class. Without going into details, problems from computationally easy class can be solved by algorithms with polynomial complexity. Conversely, computationally hard problems belong to classes for which no exact algorithms exist solving them in polynomial time. The running time, or computational complexity, of an algorithm is determined by the number of basic operations it performs to solve a problem. This execution time highly depends on the input, making the algorithm's computational complexity typically a function of the input data size.

Algorithms designed to solve combinatorial problems can be applied to a wide range of challenges and are generally classified into two major categories: exact and heuristic algorithms. Exact algorithms always produce an optimal solution, guaranteeing accuracy, but they can be computationally expensive and complex to design. Heuristic algorithms, on the other hand, are designed to tackle problems that exact methods cannot solve within a reasonable timeframe or due to other constraints. They provide an approximate solution when finding the optimal one might not be feasible. The goal of heuristics is to deliver a sufficiently good solution more quickly or with fewer resources by sacrificing factors such as optimality, completeness, accuracy, and precision. When using a heuristic approach, trade-offs must be considered to determine if the solution is adequate for the application:

- **Optimality**: Is the optimal solution necessary? Heuristics cannot guarantee finding the best solution, especially for computationally complex problems with multiple possible solutions.

- **Completeness**: Do we need all possible solutions? Heuristics typically identify only one solution, even if other solutions of similar quality exist.

- **Execution time**: Is the lower quality of the solution worth the faster execution time? Some heuristics are significantly faster than others, but the trade-off may result in only a marginal speed improvement compared to methods that provide a much better solution.

- **Accuracy and precision**: Is the solution of acceptable quality compared to the optimal one? Since heuristics do not always yield the best possible outcome, it is crucial to assess whether the quality of the obtained solution is satisfactory.

There are many heuristic techniques, with the greedy approach being one of the simplest. Greedy algorithms make locally optimal choices at each step with the goal of finding a global optimum. While they typically do not guarantee an optimal solution, they can be effective when applied strategically, often yielding good results and simplifying the problem. However, in some cases, such as the traveling salesman problem, greedy algorithms can produce suboptimal solutions that are not satisfactory. Despite this limitation, they remain useful due to their ability to provide good approximations of a global optimum with ease and speed compared to exact algorithms.

Metaheuristic algorithms, or metaheuristics, are another type of heuristic designed to provide sufficiently good solutions to optimization problems. While they do not guarantee a globally optimal solution, metaheuristics often achieve acceptable results with significantly less computational effort than exact methods. Their primary goal is to explore the solution space and find an adequate solution for the problem at hand. However, their applicability depends on the specific problem, as some combinatorial problems may not suit this approach. Metaheuristics are generally approximate and can yield non-deterministic results, meaning outcomes may vary between runs. The quality of the solution is typically linked to the execution time of the algorithm, with improvements in solution quality diminishing over time. A notable advantage of metaheuristics is their ease of parallelization, which can range from running multiple independent instances to more complex approaches where instances exchange information to enhance the overall solution. Parallelization can be achieved not only on a single machine but also across multiple systems, significantly increasing computational power. There are various types of metaheuristic approaches, such as simulated annealing, genetic algorithms, and tabu search. Each method has its own advantages and disadvantages, making some better suited to specific types of problems or data. However, these approaches generally share certain characteristics, such as non-deterministic behavior and the risk of getting trapped in local minima, which can limit their ability to fully explore the solution space.

Not to be overlooked here is Artificial Intelligence (AI), and in particular deep learning (DL), which is currently breaking into all areas of life and science, including life sciences. DL models are applied to many biological problems, especially where large datasets are available for training. However, the issues addressed in this paper do not fall into this category. We currently lack extensive information about quadruplex structures, so studies on these structures are primarily conducted using traditional methods. Such an approach is typical: problems are often first addressed with traditional methods that are interpretable, and only after accumulating sufficient knowledge and solving a substantial number of instances does the field become ready to leverage deep learning models.

1.5. Aim of the thesis

The purpose of my doctoral research was to develop new computational methods and bioinformatic tools to enhance the exploration and modeling of quadruplex structures. Specifically, I aimed to:

- Develop efficient algorithms to identify the features of quadruplexes.
- Develop algorithms to facilitate the prediction of quadruplex 3D structures from nucleotide sequences.
- Create programs to allow effective and user-friendly use of these algorithms.
- Ensure high quality and reliability of the results through rigorous computational testing.
- Make the source codes of the developed algorithms publicly available for unrestricted use, modification, and distribution.

- Disseminate the obtained results through conference presentations and publications in leading biology and bioinformatics journals.

The long-term goal of this doctoral dissertation was to contribute to a broader understanding of the relationship between the structure and function of quadruplexes by creating a robust ecosystem of reusable tools and algorithms that facilitate further discoveries in the field of nucleic acid research involving these specific motifs.

CHAPTER 2

Results

The research underlying this doctoral dissertation led to the development of several innovative methods and bioinformatic tools. They address issues related to the exploration of quadruplex features, visualization of quadruplex structures, searching for structural similarities, evaluation of structure predictions, and detection of multimeric nucleotide assemblies. All but one of the findings from this thesis were published in scientific journals. Complete texts of the published articles are included in Chapter *Publication Reprints*. The following tools are the outcome of the studies presented:

- **ONQUADRO** (https://onquadro.cs.put.poznan.pl) [A1] p. 62
- **DrawTetrado** (https://github.com/RNApolis/drawtetrado) [A2] p. 68
- **WebTetrado** (https://webtetrado.cs.put.poznan.pl) [A3] p. 70
- **GEOS, GENS** (https://github.com/RNApolis/rnahugs) [A4] p. 76
- **RNAhugs** (https://rnahugs.cs.put.poznan.pl) [A5] p. 96
- **LinkTetrado** (https://github.com/michal-zurkowski/linktetrado) unpublished (manuscript in preparation)

2.1. Exploration of quadruplex features

Quadruplex is a complex motif formed by 1 to 4 nucleic acid strands and consists of at least two stacked tetrads. Due to its intricate architecture, the quadruplex can be analyzed both globally as a complete structure and in terms of its constituent elements. This approach allows for a comprehensive examination of various parameters and features, some of which were analyzed as part of this doctoral research:

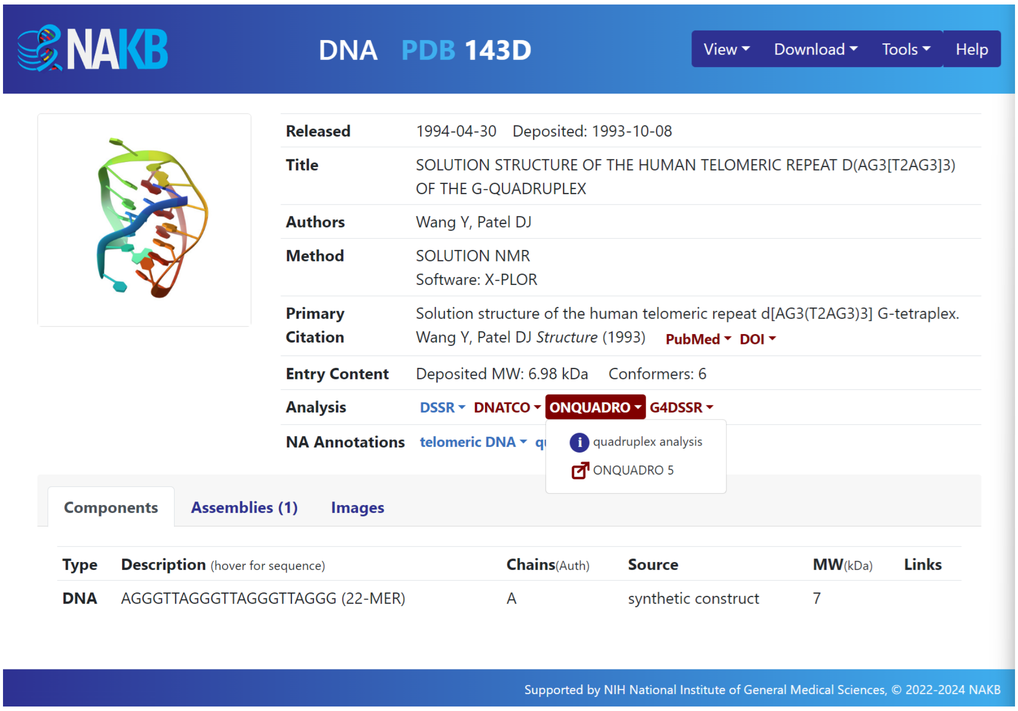
- **Quadruplex level:** PDB ID, PDB deposition date, molecule, experimental method, sequence of tetrads, ions, ionic charge, type by number of strands (unimolecular, bimolecular, tetramolecular), type by ONZM (regular, irregular), ONZM class (O, N, Z, M), number of tetrads, loop topologies by Webba da Silva, loop types (lateral, diagonal, propeller), loop lengths, tetrad combination by Webba da Silva [Webba da Silva (2007); Webba da Silva et al. (2009); Dvorkin et al. (2018)], secondary structure, tertiary structure, twist parameters, rise parameters, strand polarity (parallel, antiparallel),
- **Tetrad level:** sequence, secondary structure, ONZ type (O, N, Z) with polarity, planarity, syn/anti conformation of nucleotides, Chi angles.

Efficient analysis of these parameters and features is now possible thanks to two bioinformatic resources developed during my doctoral research. The first is *ONQUADRO* [A1], a comprehensive database system that facilitates the study of quadruplexes in experimentally resolved nucleic acid structures. The second is *WebTetrado* [A3], an agile online platform designed for on-demand analytics of new and modified quadruplexes that have not been yet deposited in the Protein Data Bank (PDB) [Bernstein et al. (1977)].

ONQUADRO compiles data on nucleic acid structures obtained from the Protein Data Bank, specifically those containing tetrads, quadruplexes, and G4-helices. It stores detailed information about their sequences, secondary, and tertiary structures. The database allows users to search for specific structures, visualize secondary and tertiary models using various representations (such as classic diagrams, arc diagrams, layer diagrams, ball-and-stick models, and surface models), and access detailed information on structural features. Additionally, *ONQUADRO* supports quantitative data analysis through statistical summaries available in both tabular and graphical formats and allows users to download data on tetrads and quadruplexes provided by the database. The database is updated weekly with new structures added to the PDB, and it modifies existing entries when necessary, ensuring that the records remain up-to-date and consistent with the Protein Data Bank. To complement the auto-update feature, an automated newsletter informs subscribers about new quadruplex structures added to the database.

As of August 2024, *ONQUADRO* has contained 1,946 tetrads, 615 quadruplexes, 36 G4-helices, and their 530 parent nucleic acid structures. In the middle of 2022, *ONQUADRO* has been integrated into the Nucleic Acid Knowledge Base (NAKB) [Lawson et al. (2023)], the world's largest repository of nucleic acid data, as a key resource on quadruplexes (Figure 2.1).

What if you have a quadruplex structure that is not stored in *ONQUADRO* and you want to analyze it? This situation might arise, for example, if a new structure containing a quadruplex has been computationally modeled or if experimental data on a new structure has been obtained but not yet submitted to the Protein Data Bank. To address such cases, the *WebTetrado* web server was developed. It provides users with a streamlined and



DNA PDB 143D

Released	1994-04-30 Deposited: 1993-10-08
Title	SOLUTION STRUCTURE OF THE HUMAN TELOMERIC REPEAT D(AG3[T2AG3]3) OF THE G-QUADRUPLEX
Authors	Wang Y, Patel DJ
Method	SOLUTION NMR Software: X-PLOR
Primary Citation	Solution structure of the human telomeric repeat d[AG3(T2AG3)3] G-tetraplex. Wang Y, Patel DJ *Structure* (1993)
Entry Content	Deposited MW: 6.98 kDa Conformers: 6
Analysis	DSSR DNATCO ONQUADRO G4DSSR
NA Annotations	telomeric DNA

Type	Description (hover for sequence)	Chains(Auth)	Source	MW(kDa)	Links
DNA	AGGGTTAGGGTTAGGGTTAGGG (22-MER)	A	synthetic construct	7	

Supported by NIH National Institute of General Medical Sciences, © 2022-2024 NAKB

Figure 2.1: Screencapture from NAKB with the information on the quadruplex structure (PDB ID: 143D) with the link to *ONQUADRO*.

efficient way to analyze structures for potential quadruplex formations and explore their features. *WebTetrado* offers comprehensive information similar to what is available in *ONQUADRO*, but with the added convenience of a web application, eliminating the need for the structure to be submitted to the PDB. All results generated by the tool are stored for two weeks with a unique identifier, allowing for easy and seamless sharing. Users can download all generated information for further analysis or record-keeping. This makes *WebTetrado* a versatile and accessible tool for researchers working with quadruplex structures.

2.2. Visualization of G4 structure

Molecular visualizations play a crucial role in analyzing biological structures. Accurate visual models provide insight into complex molecular conformations, which can drive further discoveries and research. Various representations of RNA/DNA structures offer unique perspectives and context. My journey with visualizing molecular structures began even before my doctoral studies, when I contributed to enhancing the functionality of VARNA [Darty et al. (2009)], a widely used tool to visualize 2D RNA structures. While VARNA already supported the representation of canonical base pairs, it did not accommodate non-canonical ones. I expanded its capabilities by incorporating the visualization of non-canonical base pairs. This enhancement involved adapting distinct iconography for each class of these interactions, based on the Leontis-Westhof nomenclature [Leontis & Westhof (2001)], and modifying the visualization modules to support additional non-standard connections between nucleotides.

The modified version of VARNA was subsequently integrated into *RNApdbee 2.0* [P1] as the default tool for displaying RNA 2D structure. Since its integration, it has been widely utilized for structural analysis. Notably, members of Marta Szachniuk's research team, using the enhanced visualization features, identified specific topological patterns in secondary quadruplex structures. This discovery led to the development of new classification systems for tetrads (ONZ) and quadruplexes (ONZM) [Popenda et al. (2019)]. Figure 2.2 illustrates the difference between visualizations with and without the non-canonical interactions drawn by VARNA. Getting into the subject of quadruplexes, I observed that many researchers prefer to represent these structures using so-called layer diagrams. However, no tool was available that could generate such diagrams automatically from atomic coordinates. To address this gap, I set out to develop a tool capable of this task. The result was the creation of an algorithm for generating layer diagrams based on 3D structures, implemented in the *DrawTetrado* program [A2]. A layer diagram is a 2.5D visualization that translates complex spatial information into a more easily interpretable for-

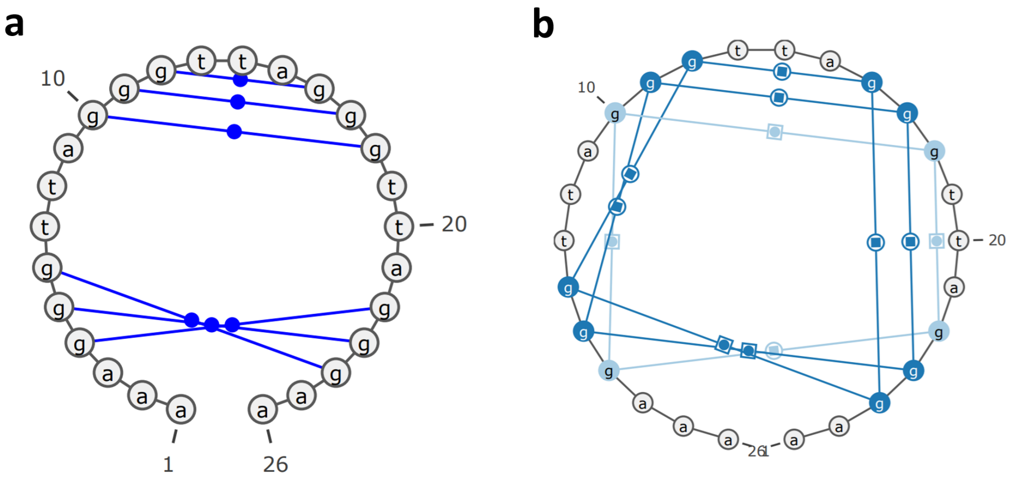


Figure 2.2: Example 2D structure (PDB ID: 2HY9) drawn in VARNA (a) before and (b) after introducing my modifications.

mat. My objective was to encapsulate as many features, aspects, and parameters of the quadruplex structure as possible within the visualization alone (Figure 2.3). Thus, the diagram is colored by default according to the ONZ nomenclature. In this scheme, tetrads assigned to the O class are colored blue, N-type tetrads are green, and Z-type tetrads are orange. Additionally, we distinguish between the + (clockwise) and − (anticlockwise) subclass concerning the interaction scheme, with the former depicted in dark shades and the latter in lighter ones. These classifications are further represented in the nucleotide shapes and their arrangement within the tetrad. The algorithm also accounts for the glycosidic bond, which adopts either anti or syn conformation; this feature is depicted using different font colors for nucleotides. The termini of each strand are clearly marked with 5′ and 3′ indicators, and the polarity of the strand is illustrated with arrows connecting the layers. The tool is optimized for readability, minimizing the complexity of the connections displayed in the visualization. It prioritizes clear, straight-line connections between adjacent tetrads and avoids crossing lines where possible. This optimization uses tract information of the quadruplex and tetrad formations, and potential rotational adjustments are made by rotating the whole tract. When tract information is unavailable, the tool adjusts the perspective by rotating some tetrads to reduce non-ideal connection types that might compromise readability. To accommodate extensive customization of the visualizations, ranging from color adjustments to changes in nucleotide representation size and spacing, I developed a robust system for drawing connections that maintains clarity regardless of the customization options. I employed Bézier curves, a parametric curve used in computer graphics, adjusting them according to the type and length of each connection. This approach ensures that even complex cases, such as V-loops and G4-helices (dimers), are accurately represented, with proper inflow and outflow of connections calculated based on the strand's directionality. This automation, coupled with optimization for readability and extensive customization options, was prioritized to facilitate the discovery and analysis of new, previously unknown structural characteristics, similar to the advancements made in VARNA visualization and the ONZ classification. *DrawTetrado* [A2] has been successfully integrated into both the *ONQUADRO* database [A1] and the *WebTetrado* system [A3], offering an automated solution for generating 2.5D layer diagram visualizations of quadruplexes.

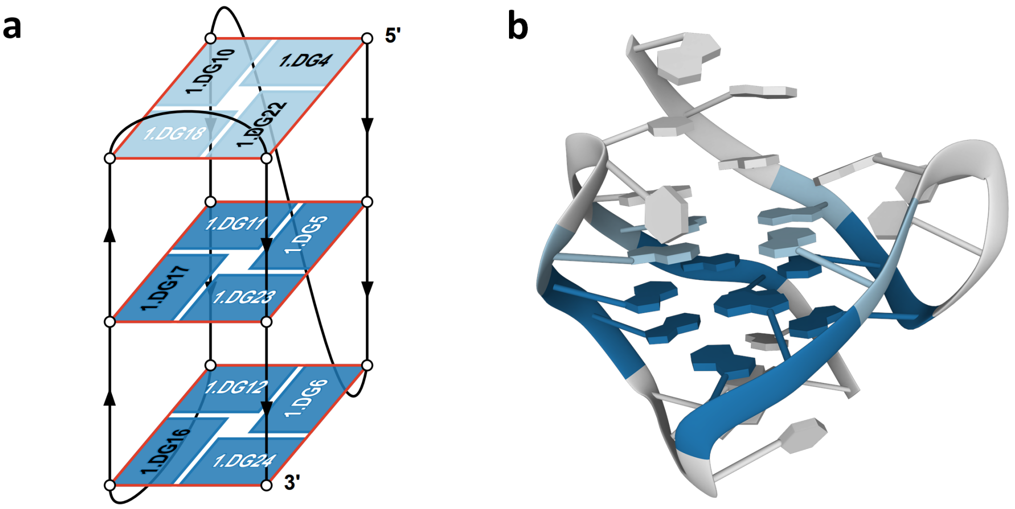


Figure 2.3: Example quadruplex structure (PDB ID: 2HY9) visualized as (a) 2.5D layer diagram created by *DrawTetrado* and (b) cartoon models of the 3D structure generated by Mol*.

2.3. 3D structure alignment

Structural alignment aims to compare the 3D structures of biological molecules, such as proteins, RNA, or DNA. Unlike sequence alignment, which compares the linear sequence of nucleotides or amino acids (i.e. the primary structure), structural alignment focuses on the spatial arrangement of atoms within the molecules. This approach is particularly valuable for identifying similarities in shape and structural motifs between molecules that may not share obvious sequence similarity but have functional or evolutionary relationships. Identifying similar fragments between two structures is crucial, as certain structural motifs can appear in molecules of different sizes and origins yet still exhibit structural similarity in specific regions (Figure 2.4). Structural alignment is also essential in molecular structure prediction, serving as the basis for calculating many prediction evaluation measures, with RMSD (Root Mean Square Deviation) being one of the most prominent. Structures can be aligned either rigidly or flexibly.

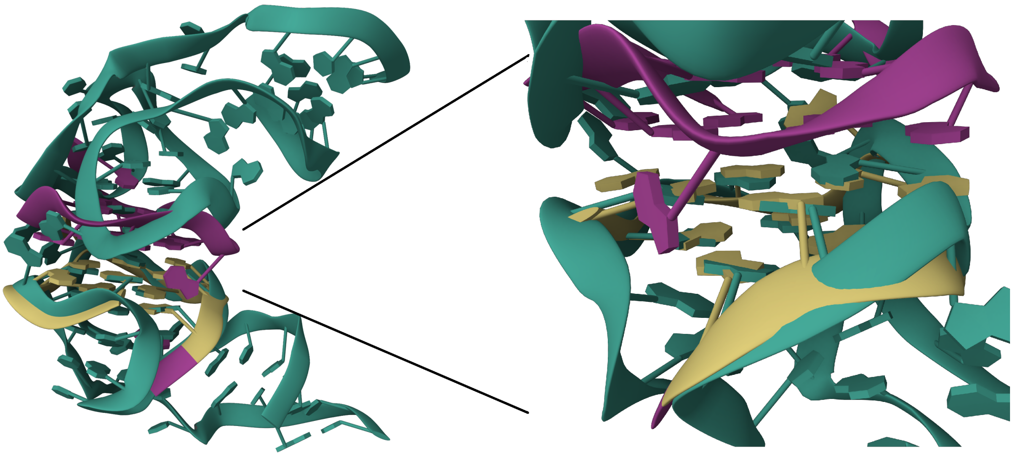


Figure 2.4: Alignment of two quadruplexes, 6E84 in teal and 2RQJ in yellow (aligned fragment) and purple (unaligned fragment), generated by *GEOS* with 2.5Å RMSD cutoff.

Rigid alignment involves applying a rigid transformation, such as rotations and translations, to superimpose one entire structure onto another. This approach is useful for comparing closely related structures. However, it can have the disadvantage of leaving entire regions unaligned, even if those regions could locally superimpose well. To address this limitation, modern alignment techniques often use flexible alignment. This approach constructs an alignment through a series of local transformations focused on specific fragments of the structures. When the goal is to identify local similarities and solve problems related to maximum common substructures, the result should include information on the location and length of matched fragments, as well as their RMSD. Flexible alignment is particularly important for comparing proteins or nucleic acids, as it allows for a more accurate assessment of alignment quality by accommodating local structural variations. These aligners can also be used to assess the global similarity of structures.

Various algorithms have been developed for structural alignment, including DALI [Holm (2020)], TM-align [Zhang & Skolnick (2005)], and CE [Shindyalov & Bourne (1998)] for proteins, and Rclick [Nguyen et al. (2016)], RMalign [Zheng et al. (2019)], and LaJolla [Bauer et al. (2009)] for RNA. These algorithms differ in their alignment strategies and the features they consider. They often simplify structures into coarse-grained models with pseudoatoms, using either a single atom for simplicity or multiple atoms for greater accuracy. The alignments are based on various parameters, including 3D coordinates, 2D units, torsion angles, and 3D geometry. Additionally, these algorithms may address other structural aspects, such as whether the alignments should be sequential or allow for different strand directionalities. The choice of alignment algorithm should therefore be tailored to the specific problem being addressed—whether the chains being compared are of the same length, whether their sequences are known and similar, whether the alignment should be sequence-dependent or independent, and whether a rigid or flexible alignment is required.

In my doctoral research, I ran several state-of-the-art alignment algorithms available publicly to compare quadruplex structures. This experiment highlighted a significant gap in current approaches: most tools provide users with a final alignment without allowing adjustments for the RMSD. The inability to fine-tune this aspect can lead to significant variations in alignment results between different structures, making them difficult to compare. To address this issue, I designed an alignment algorithm that allows users to adjust the RMSD threshold. This flexibility enables the alignment of structures with a lower RMSD value, which, although potentially shorter, would be much more spatially similar. I implemented this concept in two new algorithms, *GEOS* and *GENS*, which I developed [A4].

GEOS (Geometric Search) is a heuristics based on geometric principles detailed in Theorems 2.3.1-2.3.2. It starts by identifying a small kernel solution (Figure 2.5), aligning three nucleotides within set parameters. The algorithm then employs a greedy approach to extend the alignment until it reaches the maximum allowed RMSD value. *GEOS* is particularly effective for larger structures and those with less pronounced structural similarities.

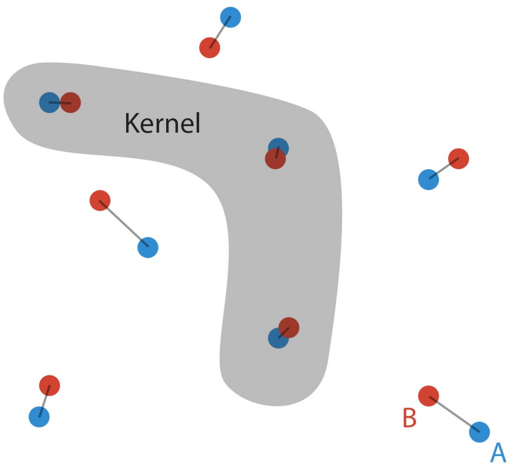


Figure 2.5: Representation of a kernel shown in a simplified 2D example, where each point corresponds to a nucleotide, approximating three key atoms in coarse-grained structural modeling.

Theorem 2.3.1. *For a given solution S with $RMSD = R$, if we remove a pair of points furthest apart from each other after the superposition, the RMSD of such S' will be $R' \leqslant R$.*

Proof. Assume, without losing the generalization, that we have N point pairs in the solution, and denote the distances between each of those pairs as $\Delta_1 ... \Delta_n$ and that for each $0 \leqslant i < n, \Delta_n > \Delta_i$. Then:
Using the fact that $\Delta_n > \Delta_i$, we get $\Delta_n^2 > \Delta_i^2$. This means the following is true as well:

$$\begin{aligned}
(n-1) \cdot \Delta_n^2 &\geqslant \Delta_1^2 + \Delta_2^2 + \cdots + \Delta_{n-1}^2 \\
(n-1) \cdot (\Delta_1^2 + \Delta_2^2 + \cdots + \Delta_n^2) &\geqslant n \cdot (\Delta_1^2 + \Delta_2^2 + \cdots + \Delta_{n-1}^2) \\
\frac{\Delta_1^2 + \Delta_2^2 + \cdots + \Delta_n^2}{n} &\geqslant \frac{\Delta_1^2 + \Delta_2^2 + \cdots + \Delta_{n-1}^2}{n-1} \\
\sqrt{\frac{\Delta_1^2 + \Delta_2^2 + \cdots + \Delta_n^2}{n}} &\geqslant \sqrt{\frac{\Delta_1^2 + \Delta_2^2 + \cdots + \Delta_{n-1}^2}{n-1}} \\
RMSD(S) &\geqslant RMSD(S')
\end{aligned}$$

Which proves the Theorem 2.3.1 □

Theorem 2.3.1 leads to the following:

Theorem 2.3.2. *An optimal solution for given N (i.e., that aims to find the best subset of N pairs), bounds the RMSD from the bottom for the optimal solution for any $M \geqslant N$.*

Proof. Let's assume the above is not true, i.e., there is an optimal solution $S(N)$ with RMSD R, and $S'(M)$ with $R' < R$. Using the Theorem 2.3.1, We can remove the worst set of points from $S'(M)$ and always end up with RMSD smaller or equal to the previous one. This means that we can repeat this process $M - N$ times until we reach $S'(N)$, and $R'' \leqslant R' < R$. This would imply that $S(N)$ is not optimal, which contradicts the initial assumption. □

The last observation pertains to a heuristic approximation that assumes a sensible solution exists. For a given Solution $S(N)$, the optimal superposition T of the molecules is very similar to the optimal superposition T' for $S(N-1)$, assuming the pair with the largest distance has been removed. Moreover, the difference between T and T' decreases as N increases. This observation is explained by the fact that RMSD for clouds of points in 3D is highly sensitive to changes in superposition. A slight rotation in the transformation can significantly increase the distance between two distant structures. As more points are added to the solution, the transformation becomes more rigid and eventually converges to a final transform.

GENS (Genetic Search) is another algorithm developed during this study, utilizing a metaheuristic genetic algorithm to find the optimal alignment. Due to the exponential increase in search space with larger structures, *GENS* is best suited for smaller or more structurally similar molecules. The non-deterministic nature of metaheuristics allows for the discovery of varied solutions with each run, offering different alignments that may be similar in length but located in distinct regions of the structures. To address some of the inherent uncertainty in metaheuristic approaches, I incorporated functionality to initialize the population of the *GENS* algorithm with solutions from *GEOS*. This approach mitigates some of the randomness associated with metaheuristics, enabling a more focused exploration of potential solutions and resulting in more consistent outcomes. By starting with robust initial solutions from *GEOS*, the metaheuristic can further refine and optimize the alignment, potentially leading to superior results compared to the greedy algorithm alone.

GEOS and *GENS* were benchmarked against all available algorithms for 3D structure alignment. I started by running the competing algorithms and calculating the RMSD of their resulting alignments. I then adjusted the RMSD parameter in my algorithms to match those of the competing methods, allowing for a fair comparison of alignment lengths. Using the RNA-Puzzle dataset for testing, *GEOS* and *GENS* consistently produced longer alignments than the other methods [Table 3, Page 6, Article 4].

Both algorithms are available as standalone applications, which is particularly advantageous when multiple alignments are needed, such as during testing or benchmarking. The standalone version also facilitates easier integration with other tools or workflows, providing faster feedback on the similarity between reference structures and modeled solutions. However, I recognized that this approach might not be ideal for researchers who need only a few alignments. To improve accessibility, I developed the *RNAhugs* web application [A5]. It is available online, allowing users to run alignments directly via a web browser. This platform offers a comprehensive solution for aligning RNA structures, supporting multiple models and reference structures, and aggregating and presenting detailed information about the alignment results. It operates in two modes: sequence-dependent and sequence-independent. Sequence-dependent alignment is the more conventional approach, aligning closely with the biological functions of the fragments being compared. In contrast, the sequence-independent approach does not consider sequence information, making it suitable for identifying structurally similar fragments from a purely 3D perspective, especially when dealing with significantly varied sequences. Other key features of *RNAhugs* include color-coded on-site 3D visualizations that provide clear insights into aligned fragments between the model and reference structures, residue alignment within a sequence context, and residue mapping for individual fragments, including matched fragments and sequence mismatches. Users can also customize and adjust search functions to suit their needs. All results are stored for two weeks with a unique identifier, enabling easy and seamless sharing without the need for recomputation.

Identifying common features and differences in the 3D structures of biomolecules is a complex task that requires advanced computational methods, many of which involve structural alignment procedures. The developed algorithms offer robust tools for finding alignments between structures and are designed for integration with other systems and for facilitating on-site alignments. Building on this, the *RNAhugs* web application enhances usability with an intuitive interface for aligning structures, complemented by additional features such as on-site 3D visualization and clearly presented alignment results. Both the algorithms and the web application focus on improving the critical aspect of structural comparison.

2.4. Multimeric motif detection

Multimeric motifs were observed in nucleic acid structures relatively recently [Zhang et al. (2001)]. They are closely related to tetrads and quadruplexes, as they form when additional nucleotides align within the plane of a tetrad and pair with its nucleotides, effectively extending the tetrad. For instance, a pentad consists of five interacting nucleotides arranged in the same plane, where the fifth nucleotide pairs with one nucleotide of the tetrad. Similarly, a hexad consists of six nucleotides, a heptad of seven, and so forth (Figure 2.6 and Figure 2.7). A stack of tetrads forms a quadruplex. In the remaining part of this chapter, I will refer to stacks composed of pentads, hexads, and other similar motifs as multiplexes or multimeric motifs.

Like quadruplexes, multimeric motifs are stabilized by hydrogen bonds and metal ions. While their functions are not yet fully understood, they are believed to contribute to regions of the genome that require sophisticated regulatory mechanisms.

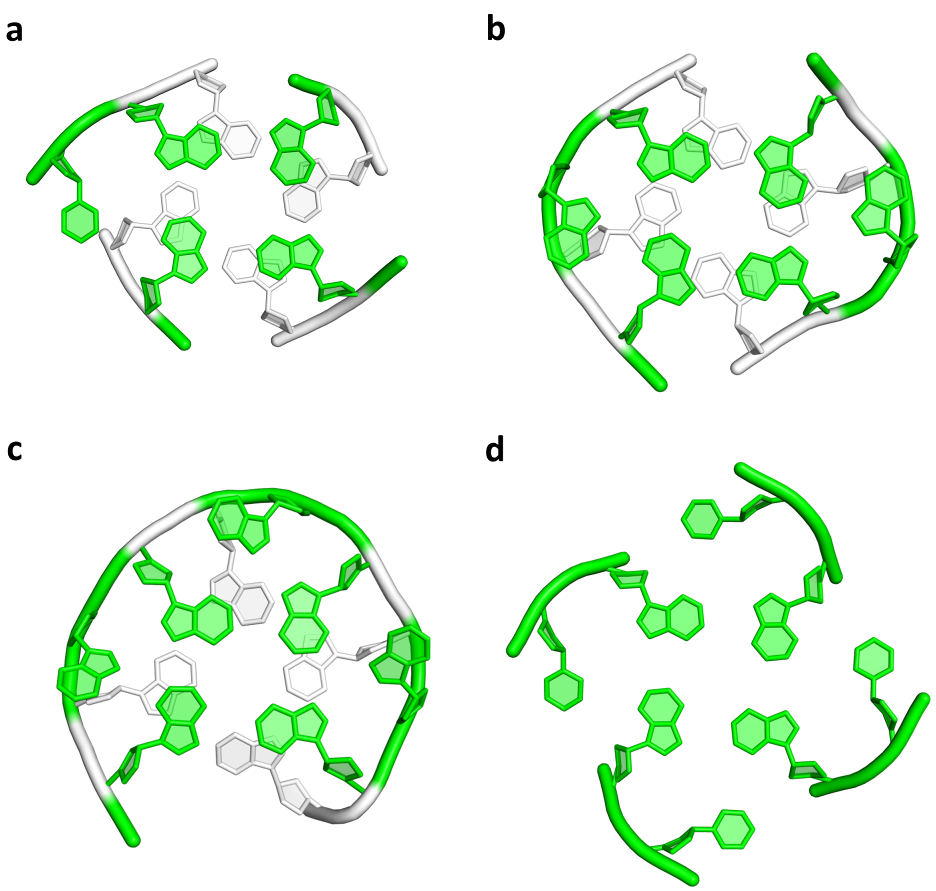


Figure 2.6: Example structures containing (a) a pentad (PDB ID: 2GRB), (b) a hexad (PDB ID: 2RQJ), (c) a heptad (PDB ID: 1OZ8), and (d) an octad (PDB ID: 1J6S). Respective multimeric motifs are colored green.

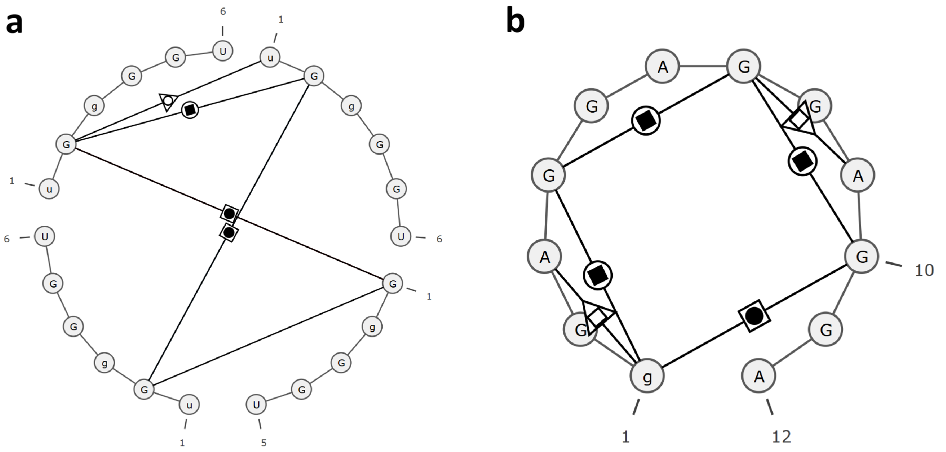


Figure 2.7: Interactions in (a) a pentad (PDB ID: 2GRB) and (b) a hexad (PDB ID: 2RQJ) drawn in the 2D structure diagram prepared by a modified version of VARNA.

The complexity of multimeric motifs poses significant challenges for researchers, particularly in their identification and characterization. Advanced computational tools are therefore essential for exploring these motifs, potentially uncovering new insights into their roles in genetic regulation and cellular function. To address this challenge, I developed the *LinkTetrado* algorithm, which is capable of identifying and describing multimeric motifs in nucleic acid structures. Although the results of this project have not yet been published, a paper is currently in progress.

LinkTetrado analyzes nucleic acid 3D structures, accepting both PDB and mmCIF file formats. It leverages the *WebTetrado* engine to extract detailed structural information about base pairs and tetrads in an input structure. Next, it searches the space in the vicinity of tetrads for possible nucleotides that could interact with the tetrads. This process utilizes structural information, such as base pairs and connection types between nucleotides, as well as raw geometric spatial data, which provide a wealth of information. Several parameters and geometric relationships between a potential nucleotide N and tetrad T (including nucleotides N') are considered to determine whether nucleotide N can be added to the motif. First, it is crucial to ensure that the considered nucleotide N does have some connections to tetrad T. This is done by checking if N has any base pair connections to N' or if it is the next or previous nucleotide in the sequence, thus, providing a connection by maintaining the continuity of the strand. Next, the algorithm verifies that N does not affect the planarity of the final multimeric motif. To do this, the algorithm calculates the tilt of the nucleobase of N relative to all the N' within the tetrad by computing the arccosine of the dot product of the base vector of N and N' (Equation 2.1). This gives us the tilt of the N nucleobase relative to each N'. Further experiments showed that the average tilt should not exceed 40° with a maximum of 50°.

$$Tilt = \arccos{(\vec{N} \cdot \vec{N'})} \tag{2.1}$$

Next, we ensure that N is not too far from the tetrad T and does not overlap with any N'. For this, the algorithm calculates the Euclidean distance from the center of T and from each N'. The distance from the center should be < 13 Å, and the difference between the two closest N' and N should be < 2 Å, to maintain a symmetrical shape of the potential multimeric motif formation (Equation 2.2).

$$Distance = \sqrt{(N_x - N'_x)^2 + (N_y - N'_y)^2 + (N_z - N'_z)^2} \tag{2.2}$$

The final geometric property to validate is whether N lies within the same plane as T. This is done by calculating the geometric center of the nucleobase of N (GC_N) and the geometric center of T (GC_T), and then computing the dot product of the vector $GC_T - GC_N$ relative to the base vector of each N'. The plane height difference should be < 3.0 Å for each N' (Equation 2.3).

$$HeightDiff = (\vec{GC_T} - \vec{GC_N}) \cdot \vec{N'} \tag{2.3}$$

If all parameters fall within the established thresholds, N is considered a candidate for building a multimeric motif with T. After analyzing all tetrads and nucleotides, the algorithm ensures that no nucleotide is associated with multiple tetrads. In such cases, the nucleotide is removed from the tetrad with worse overall parameters. The algorithm also checks that the number of potential nucleotides for tetrads is consistent, based on the assumption that no different levels of multimeric motifs can exist within the same stacked tetrad formation — a conclusion derived from analyzing all documented multimeric structures. Finally, the algorithm compiles a list of nucleotides likely associated with each tetrad, which are then classified together as a multimeric motif. A generalized flowchart showcasing all these steps is presented in Figure 2.8.

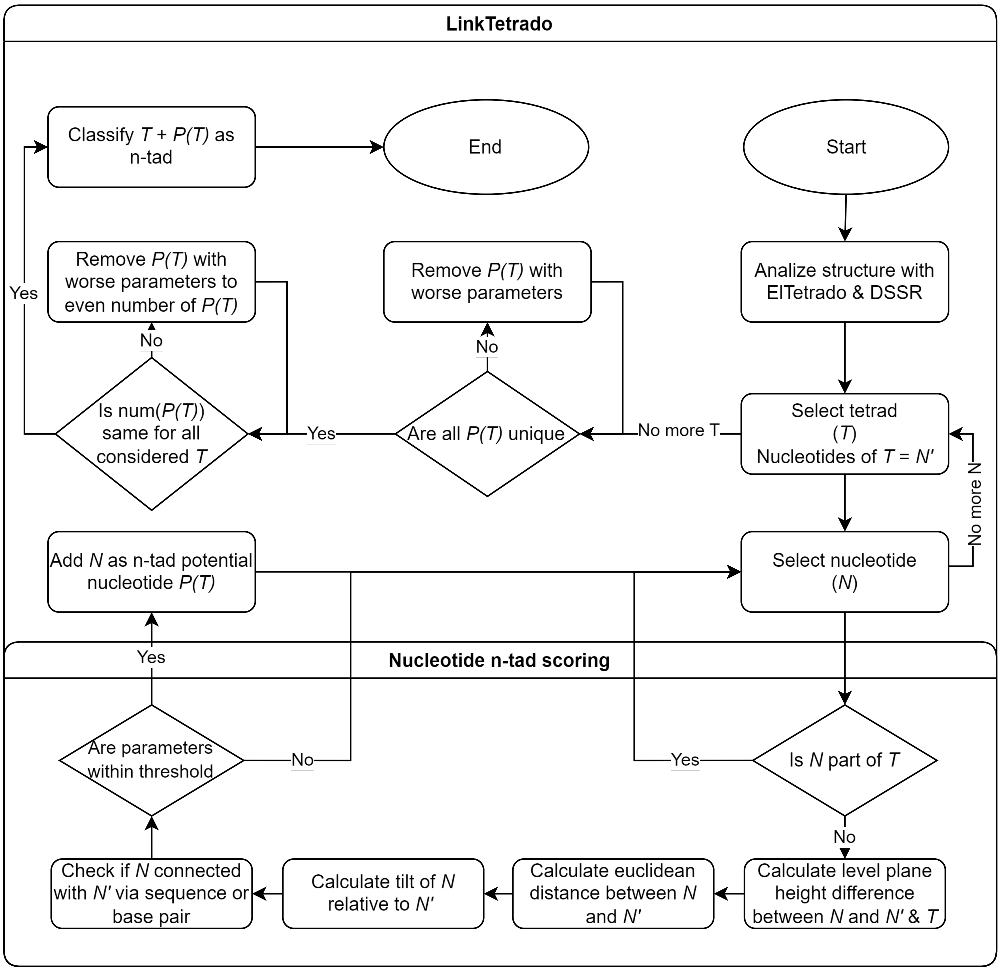

LinkTetrado
Start
Analize structure with ElTetrado & DSSR
Select tetrad (T) Nucleotides of T = N′
Select nucleotide (N)
No more N
Nucleotide n-tad scoring
Is N part of T
Yes
No
Calculate level plane height difference between N and N′ & T
Calculate euclidean distance between N and N′
Calculate tilt of N relative to N′
Check if N connected with N′ via sequence or base pair
Are parameters within threshold
No
Yes
Add N as n-tad potential nucleotide P(T)
No more T
Are all P(T) unique
No
Remove P(T) with worse parameters
Yes
Is num(P(T)) same for all considered T
No
Remove P(T) with worse parameters to even number of P(T)
Yes
Classify T + P(T) as n-tad
End

Figure 2.8: *LinkTetrado* flowchart.

The algorithm correctly identified 25 structures with multimeric motifs (excluding different assemblies of the same structure), 8 of which had these motifs described in associated papers. Within this subset of 25 structures, 6 contained pentads, 13 contained hexads, 1 contained heptads, and 5 contained octads. Some structures featured multiple multimeric motifs, though these motifs were always of the same type within the stacked region containing tetrads. The distribution and frequency of these motifs are detailed in Figure 2.9.

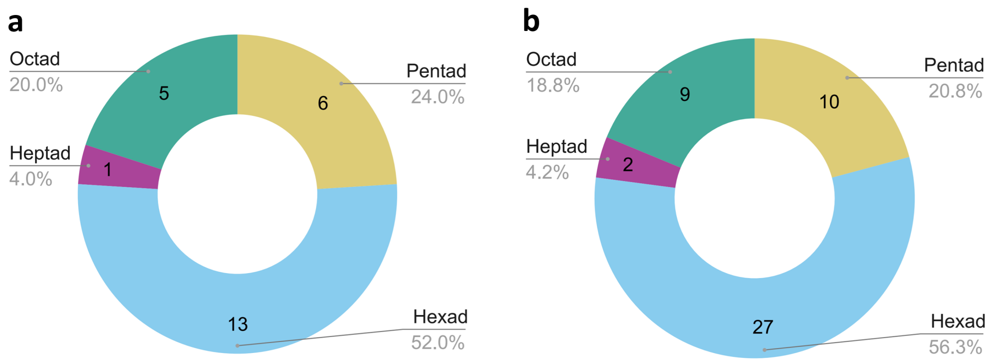


Figure 2.9: Pie charts showing (a) the number of nucleic acid structures with multimeric motifs and (b) the number of multimeric motifs found in all experimental structures.

I observed that motifs with an even number of nucleotides, such as tetrads, hexads, and octads, tend to maintain symmetrical properties, which enhances their structural stability. Hexads are more common than pentads, likely due to the added structural rigidity provided by their symmetrical shape. Although octads share similar symmetrical characteristics with hexads, they are less frequent, possibly due to their higher overall complexity. Additionally, I identified some helices containing two quadruplexes where the terminal ends were composed of multimeric motifs (Figure 2.10).

In the multimeric subset, I identified 9 DNA structures, 15 RNA structures, and 1 DNA/RNA hybrid (PDB ID: 1N7A). All structures were experimentally determined, with 14 solved by solution NMR and 11 by X-ray diffraction. Octads (5) were only found in structures obtained via X-ray.

Additionally, *LinkTetrado* produced 6 false positives, where structures were incorrectly identified as containing pentads (these are not included in the above results). These errors were detected during manual validation of the results, as described in the next section. I am currently developing a modified classification metric to address this issue. Initial tests with a simple machine learning model, based on the current geometrical metrics, have shown promising results, though further validation is required to ensure the quality and reliability of the algorithm's outputs.

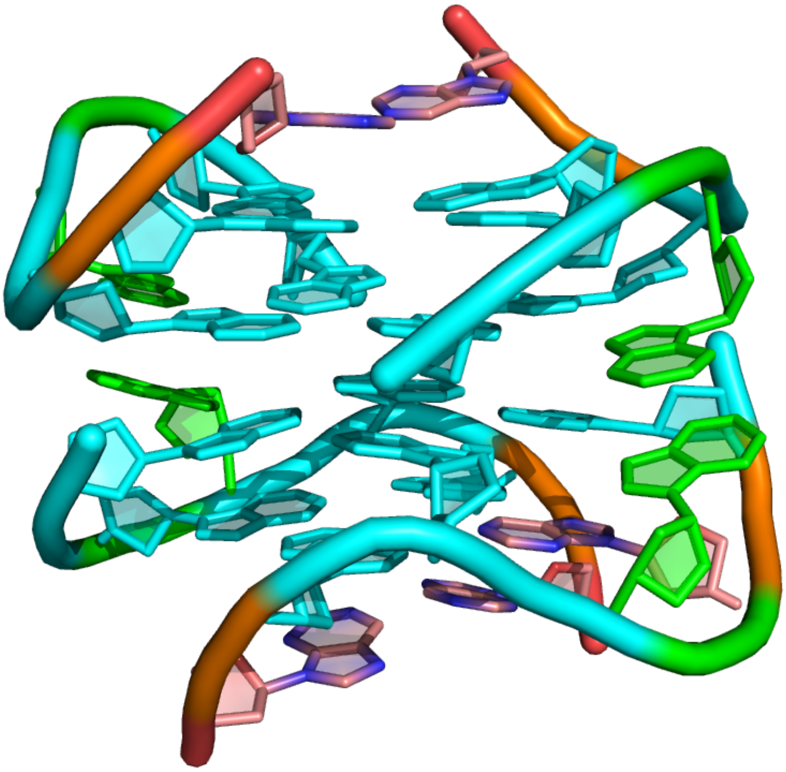
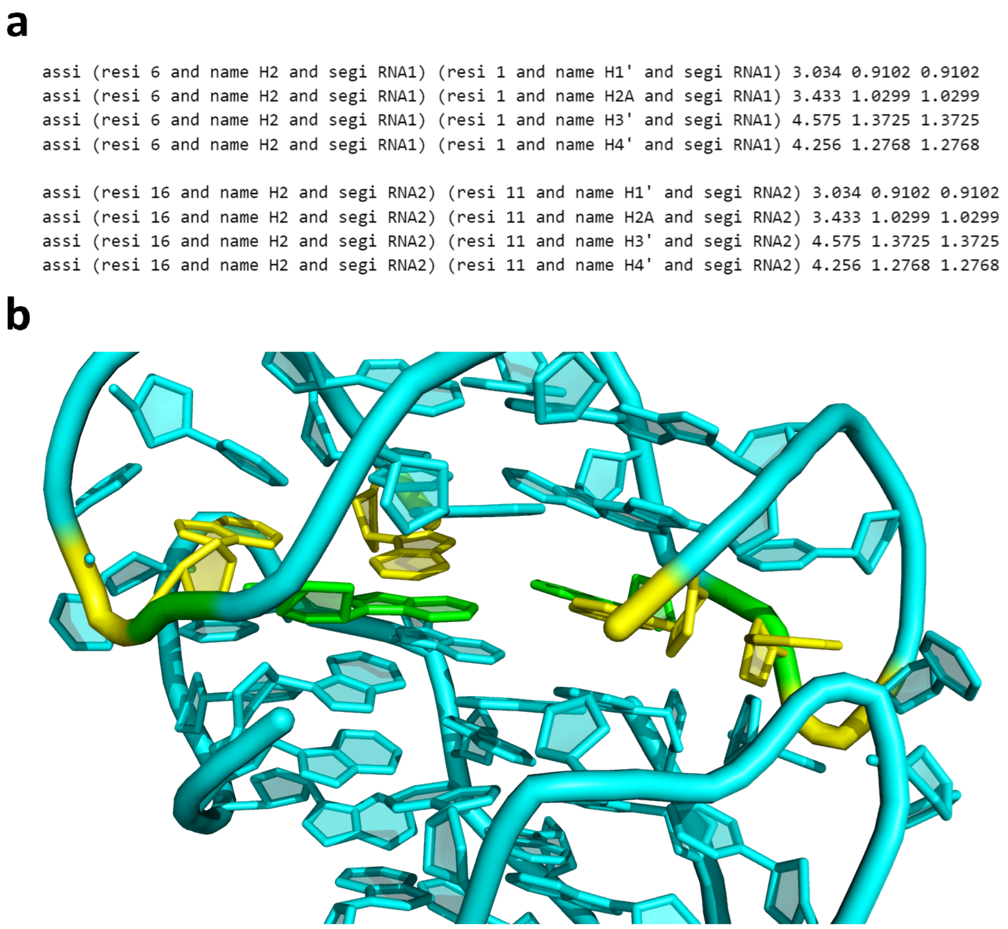


Figure 2.10: Structure (PDB ID: 1EEG) containing two stacked quadruplexes (colored teal) with hexads in between (expanded nucleotides colored green) visualized in PyMOL.

Research on multimeric structures was particularly challenging due to the limited number of known structures containing these high-order motifs and the scarcity of related studies. The absence of an existing method capable of identifying multimeric motifs further complicated the process and hindered the ability to cross-validate the findings from *LinkTetrado*. As a result, I had to meticulously gather information about structures with potential multiplexes. Initially, I identified and validated structures containing multimeric motifs by relying on existing research related to these structures. However, given my previous experience with quadruplexes and motif identification, I anticipated the discovery of new structures containing multiplexes that were not yet documented in the literature. To ensure robust validation of the results, I established close collaboration with several researchers who are experts in this field. We worked closely with national researchers Prof. Dr. Zofia Gdaniec, Dr. Dorota Gudanis-Sobocińska, and Dr. Witold Andrałojć from the Department of Biomolecular NMR at the Institute of Bioorganic Chemistry PAS, as well as with Dr. Maja Marušič from the Slovenian NMR Center. Their support was invaluable in confirming the validity of the identified structures containing multiplexes. This collaboration also deepened my understanding of NMR experiments, enabling me to further validate some structures based on NMR restraints (Figure 2.11).



```
assi (resi 6 and name H2 and segi RNA1) (resi 1 and name H1' and segi RNA1) 3.034 0.9102 0.9102
assi (resi 6 and name H2 and segi RNA1) (resi 1 and name H2A and segi RNA1) 3.433 1.0299 1.0299
assi (resi 6 and name H2 and segi RNA1) (resi 1 and name H3' and segi RNA1) 4.575 1.3725 1.3725
assi (resi 6 and name H2 and segi RNA1) (resi 1 and name H4' and segi RNA1) 4.256 1.2768 1.2768

assi (resi 16 and name H2 and segi RNA2) (resi 11 and name H1' and segi RNA2) 3.034 0.9102 0.9102
assi (resi 16 and name H2 and segi RNA2) (resi 11 and name H2A and segi RNA2) 3.433 1.0299 1.0299
assi (resi 16 and name H2 and segi RNA2) (resi 11 and name H3' and segi RNA2) 4.575 1.3725 1.3725
assi (resi 16 and name H2 and segi RNA2) (resi 11 and name H4' and segi RNA2) 4.256 1.2768 1.2768
```

Figure 2.11: (a) A fragment of the NMR restraints file obtained for structure PDB ID: 2M18 containing a hexad. The fragment includes information on the interactions between nucleotides 6 and 1, as well as 16 and 11. (b) The corresponding structure with the tetrad (made of residues 1, 7, 11, 17) colored yellow and the remaining nucleotides forming a hexad (6, 16) colored green.

Currently, this knowledge is applied to manually validate results when the structure was obtained via solution NMR. However, I plan to implement an automated method into *LinkTetrado* to further enhance the accuracy of the algorithm for these cases.

Looking ahead, the goal is to integrate *LinkTetrado* and the data on identified multimeric motifs into existing quadruplex analytical tools and databases, such as *WebTetrado* and *ONQUADRO*. This integration will expand the available information in the field and enhance the accessibility of these resources to the research community.

CHAPTER 3

Conclusions

The goals set in my doctoral research were achieved as follows.

Goal 1: Develop efficient algorithms to identify the features of quadruplexes. I designed several components of the *ONQUADRO* database to facilitate the exploration of currently known experimental structures of quadruplexes. These include (i) an algorithm for annotating non-canonical base pairs in tetrad structures, which aids in the classification of tetrad topology according to the ONZ nomenclature and quadruplex classification according to ONZM, (ii) functions that aggregate quadruplex data for statistical analysis, and (iii) the *DrawTetrado* algorithm, which generates layer diagrams of quadruplexes. The annotation and drawing algorithms have the potential to explore new quadruplex structures and have, therefore, been incorporated into the *WebTetrado* system. I developed two algorithms for 3D structure alignment, *GEOS* and *GENS*, which enable the detection of similar substructures in compared molecules. These algorithms have proven useful for aligning and analyzing quadruplex motifs. Additionally, I created *LinkTetrado*, the first algorithm designed for the identification and classification of multimeric nucleotide assemblies in nucleic acid structures. *LinkTetrado* automatically identifies nucleotides interacting with tetrads in a planar arrangement, allowing for the detection of pentads, hexads, heptads, octads, and beyond.

Goal 2: Develop algorithms to facilitate the prediction of quadruplex 3D structures from nucleotide sequences. The *GEOS* and *GENS* algorithms developed during my doctoral research are designed to identify and analyze structural similarities in compared molecules. Additionally, they can be applied to evaluate the prediction of 3D structures. To facilitate this usage, I implemented a parameterized input, allowing users to set a minimum similarity threshold, and included functionality to compute the length of alignment. This value serves as a distance measure, enabling the evaluation and ranking of predicted 3D models by comparing them to a reference structure. However, the primary objective here was to develop algorithms allowing for the automated prediction of quadruplex 3D structures based on nucleic acid sequences. Unfortunately, due to the limited amount of data, insufficient representation across various conformational classes, and the significant polymorphism of quadruplex structures, I was unable to create such an algorithm. The current understanding of G4 structures remains quite limited, comparable to our knowledge of RNAs 30-40 years ago. Further analysis revealed that a predictive system would currently require extensive user input for nearly every aspect of the desired quadruplex structure, making it more of a manual process than a true prediction. Given that a similar system already exists, I concluded that duplicating this effort would not be beneficial to the community.

Goal 3: Create programs to allow effective and user-friendly use of these algorithms. To facilitate the easy use of the designed algorithms, *WebTetrado*, *ONQUADRO*, and *RNAhugs* were developed during my doctoral research. These systems operate via a user-friendly, multimodal web interface compatible with any modern web browser, whether on mobile or desktop devices. They are freely and openly accessible to everyone. Comprehensive tutorials and help sections have been created for all three resources, providing users with the necessary information for using the services and understanding the generated results. All data provided by the database or computed in *WebTetrado* and *RNAhugs* can be downloaded for further use and processing. Each computation task in *WebTetrado* and *RNAhugs* is assigned a unique identifier, allowing users to revisit and share the results for up to two weeks.

Goal 4: Ensure high quality and reliability of the results through rigorous computational testing. All developed tools underwent rigorous evaluation before publication. *DrawTetrado* visualizations were validated by inspecting the readability of computer-generated models to simulate unrealistic features and connections within quadruplexes. Additionally, it was tested by generating visualizations for all structures within the *ONQUADRO* database. *ONQUADRO* was equipped with a self-updating module to ensure that all available data remains up to date. *RNAhugs*, along with its component algorithms *GEOS* and *GENS*, was extensively benchmarked against other state-of-the-art alternatives and evaluated on a dataset of over 1,000 structures. The *RNAhugs* web server underwent a similar rigorous examination on a large dataset to ensure validity and usability, including torture tests to verify the overall stability of the service. For *WebTetrado*, I prepared a verification dataset comprising over 1,900 tetrads, 600 quadruplexes, and 30 helices to guarantee the accuracy and correctness of the results. Performance tests were conducted for *ONQUADRO*, *WebTetrado*, and *RNAhugs* to ensure optimal performance in multi-user scenarios. The results from *LinkTetrado*, obtained by analyzing over 500 structures, were validated through close collaboration with researchers from the Department of Biomolecular NMR at IBCH PAS and Dr. Maja Marušič from the Slovenian NMR Center.

Goal 5: Make the source codes of the developed algorithms publicly available for unrestricted use, modification, and distribution. The algorithms and tools I have developed are publicly available and free of charge. Specifically, *DrawTetrado*, *GEOS*, *GENS*, and *LinkTetrado* are hosted on GitHub under the MIT license, allowing for unrestricted use, modification, and distribution of the code in both its original and modified forms. Additionally, *ONQUADRO*, *WebTetrado*, and *RNAhugs*, developed as web applications, are hosted by the Institute of Computing Science at Poznan University of Technology and are freely accessible without login requirements. One more evidence of the successful realization of this goal is the integration of *ONQUADRO* with the Nucleic Acid Knowledgebase (NAKB). NAKB provides direct links to *ONQUADRO* whenever a deposited nucleic acid is identified as a quadruplex, enabling users to seamlessly access detailed information and utilize the *ONQUADRO* database in their research.

Goal 6: Disseminate the obtained results through conference presentations and publications in leading biology and bioinformatics journals. During the 4 years of my PhD, I presented my results at a total of 12 national and international seminars and conferences. All findings, except for those related to *LinkTetrado*, were published in 5 scientific articles in leading journals in the fields of bioinformatics and nucleic acids.

Looking ahead, I plan to enhance the search functionality in *ONQUADRO*, enabling more refined searches based on specific features. I also intend to publish the findings related to *LinkTetrado*, with the manuscript currently in progress. Additionally, I want to integrate *LinkTetrado* into both *ONQUADRO* and *WebTetrado* to enrich their functionalities and expand the database contents. I will continue to closely monitor developments in the quadruplex field to advance fully automated sequence-based prediction of quadruplex 3D structures. Moreover, I aim to apply the knowledge gained from my research on quadruplexes to investigate other complex structures, beginning with intercalated motifs (i-motifs), which is the focus of my recent grant application submitted to the National Science Centre, Poland.

CHAPTER 4

Scientific achievements

4.1.

Other publications

P1. Tomasz Zok, Maciej Antczak, **Michal Zurkowski**, Mariusz Popenda, Jacek Blazewicz, Ryszard W. Adamiak, Marta Szachniuk (2018) RNApdbee 2.0: multifunctional tool for RNA structure annotation. *Nucleic Acids Research* 46(W1), W30-W35 (doi:10.1093/nar/gky314).
5-IF(2024): 16.1; MEiN(2024): 200; citations: 71 (Web of Science), 75 (Scopus), 109 (Google Scholar)
My contribution: I improved the VARNA visualization software by incorporating the ability to handle dot-bracket notation with multiple levels of brackets, enabling more complex pseudoknotted RNA structures to be accurately represented. Additionally, I introduced functionality to visualize non-canonical interactions between nucleotides with their graphical annotation according to the Leontis-Westhof classification. I reimplemented established algorithms for pseudoknot classification into Java programming language, optimizing them for

compatibility with the *RNApdbee* application. Moreover, I designed and implemented a novel dynamic programming algorithm to annotate pseudoknots that significantly improved both the accuracy and speed of pseudoknot classification, elevating the overall performance of RNA structural analysis in the *RNApdbee* system.

P2. Maciej Antczak, Mariusz Popenda, Tomasz Zok, **Michal Zurkowski**, Ryszard W. Adamiak, Marta Szachniuk (2018) New algorithms to represent complex pseudoknotted RNA structures in dot-bracket notation. *Bioinformatics* 34(8), 1304-1312 (doi:10.1093/bioinformatics/btx783).
5-IF(2024): 7.6; MEiN(2024): 200; citations: 29 (Web of Science), 30 (Scopus), 42 (Google Scholar)
My contribution: I designed and implemented a novel algorithm leveraging dynamic programming for pseudoknot identification and classification, which led to substantial improvements in both the accuracy and speed of the classification process compared to previous solutions. The newly introduced pseudoknot classification scoring functions served as a fundamental element for the development of the novel Hybrid Algorithm presented in this article, further enhancing the robustness of RNA structural analysis.

4.2. Participation in research projects

- Subject: Feature exploration and modelling of quadruplex structures
  Grant number: 2019/35/B/ST6/03074
  Granting body: National Science Centre (Poland)

Participation period: 14.07.2020 - 13.07.2023

Principal investigator: prof. dr hab. inż. Marta Szachniuk

- Subject: Effective algorithms for RNA 3D structure comparison

  Grant number: 0311/SBAD/0730 (Młoda Kadra)

  Granting body: Poznan University of Technology (Poland)

  Participation period: 14.04.2022 – 30.11.2023

  Principal investigator: mgr inż. Michał Żurkowski

- Subject: New optimization algorithms in the ONQUADRO system

  Grant number: 0311/SBAD/0759 (Młoda Kadra)

  Granting body: Poznan University of Technology (Poland)

  Participation period: 28.03.2024 – 31.12.2024

  Principal investigator: mgr inż. Michał Żurkowski

4.3. Conference presentations

During my Ph.D. study, I gave 12 presentations (talks and posters) at national and international scientific conferences and seminars:

1. *Novel methods for RNA 3D structure alignment*, ICOLE2021: International Colloqium Lessach, September 2021, Lessach, Austria (talk).
2. *DrawTetrado - Simplified Visualization of Quadruplexes*, 5th SICIM-Workshop Bioinformatic meets Machine Learning, December 2021, online event (talk).
3. *Novel methods for RNA 3D structure alignment*, PUT seminar, January 2022, Poznan, Poland (talk).

4. *Genetically or geometrically? How to optimally superimpose RNA structures*, ECCO XXXV - CO 2022 Joint Conference: European Chapter on Combinatorial Optimization, June 2022, online event (talk).

5. *Genetically or geometrically? How to optimally superimpose RNA structures*, PTBI 2022: - Symposium of Polish Bioinformatics Society, September 2022, Warsaw, Poland (talk).

6. *WebTetrado your assistant in the G4 space*, Eutopia Workshop: Structure and Topology of RNA in Living Systems, January 2023, Trento, Italy (talk).

7. *DrawTetrado - Simplified Visualization of Quadruplexes*, PUT seminar, March 2023, Poznan, Poland (talk).

8. *ONQUADRO: a database of experimentally determined 3D quadruplex structures*, Reporting session of the Institute of Bioorganic Chemistry, PAS, May 2023, Poznan, Poland (poster).

9. *ONQUADRO: a database of experimentally determined 3D quadruplex structures*, BIT23: Bioinformatics in Torun, June 2023, Torun, Poland (poster).

10. *WebTetrado your assistant in the G4 space*, ICOLE2023: International Colloqium Lessach, September 2023, Lessach, Austria (talk).

11. *Interaction graphs as a way to discover multiplexes in nucleic acid structures*, ECCO XXXVII: European Chapter on Combinatorial Optimization, June 2024, Ghent, Belgium (talk).

12. *Interaction graphs as a way to discover multiplexes in nucleic acid structures*, G4 webinar, June 2024, online event (talk).

4.4. Awards and distinctions

- 2018: Scholarship for outstanding academic achievements awarded by the Minister of Science and Higher Education, Poland for the academic year 2018/2019.
- 2019: Award for the bachelor thesis titled *A new method for pseudoknot identification* (supervisor: prof. M. Szachniuk) in the B.Sc. Thesis Competition under the Patronage of IEEE (Institute of Electrical and Electronics Engineers).
- 2020: The status of Arctic Code Vault Contributor achieved from GitHub, with code contributions preserved in several repositories as part of the 2020 GitHub Archive Program (some of my codes have been stored in GitHub's Arctic vault located in an abandoned mine on Spitsbergen. This storage is expected to last for 500-1000 years).
- 2023: Best Paper Award for the top doctoral student publication on RNA, given at the RNA Salon Poznan 2023 competition.

  Awarded paper: [A4] Michal Zurkowski, Maciej Antczak, Marta Szachniuk (2023) High-quality, customizable heuristics for RNA 3D structure alignment. *Bioinformatics* 39(5), btad315 (doi: 10.1093/bioinformatics/btad315).

4.5. Didactics

- Construction of cloud systems (Konstrukcja systemów chmurowych), laboratories, winter semester 2020/2021
- System and concurrent programming (Programowanie systemowe i współbieżne), laboratories, winter semesters: 2020/2021, 2022/2023, 2023/2024
- Distributed computing (Przetwarzanie rozproszone), laboratories, summer semesters: 2020/2021, 2021/2022, 2022/2023, 2023/2024
- Operating systems (Systemy operacyjne), laboratories, summer semesters: 2020/2021, 2021/2022, 2022/2023, 2023/2024, winter semester 2023/2024
- Computer networks (Sieci komputerowe), laboratories, summer semester 2020/2021, winter semester 2022/2023

Bibliography

Abramson, J., Adler, J., Dunger, J. et al. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016), 493–500.

Albertini, A. A. V., Schoehn, G., Weissenhorn, W. and Ruigrok, R. W. H. (2007). Structural aspects of rabies virus replication. *Cell Mol Life Sci*, 65(2), 282–294.

Bauer, R. A., Rother, K., Moor, P. et al. (2009). Fast Structural Alignment of Biomolecules Using a Hash Table, N-Grams and String Descriptors. *Algorithms*, 2(2), 692–709.

Bedrat, A., Lacroix, L., and Mergny, J.-L. (2016). Re-evaluation of G-quadruplex propensity with G4Hunter. *Nucleic Acids Res*, 44(4), 1746–1759.

Berg, J. (2002). *Biochemistry.* New York: W.H. Freeman.

Bernstein, F. C., Koetzle, T. F., Williams, G. J. et al. (1977). The protein data bank: A computer-based archival file for macromolecular structures. *J Mol Biol*, 112(3), 535–542.

Bourne, P. E., Berman, H. M., McMahon, B. et al. (1997). [30] Macromolecular crystallographic information file. In *Macromolecular Crystallog-*

raphy Part B, volume 277 of *Methods in Enzymology* (pp. 571–590). Academic Press.

Brázda, V., Kolomazník, J., Lýsek, J. et al. (2019). G4Hunter web application: a web server for G-quadruplex prediction. *Bioinformatics*, 35(18), 3493–3495.

Byun, Y. and Han, K. (2009). PseudoViewer3: generating planar drawings of large-scale RNA structures with pseudoknots. *Bioinformatics*, 25(11), 1435–1437.

Coutsias, E. A., Seok, C., and Dill, K. A. (2004). Using quaternions to calculate RMSD. *J Comput Chem*, 25(15), 1849–1857.

Cruz, J. A., Blanchet, M.-F., Boniecki, M. et al. (2012). RNA-Puzzles: a CASP-like evaluation of RNA three-dimensional structure prediction. *RNA*, 18(4), 610–625.

Darty, K., Denise, A., and Ponty, Y. (2009). VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, 25(15), 1974–1975.

Davis, I. W., Leaver-Fay, A., Chen, V. B. et al. (2007). MolProbity: all-atom contacts and structure validation for proteins and nucleic acids. *Nucleic Acids Res*, 35(Web Server), W375–W383.

Doudna, J. A. and Cech, T. R. (2002). The chemical repertoire of natural ribozymes. *Nature*, 418(6894), 222–228.

Dvorkin, S. A., Karsisiotis, A. I., and da Silva, M. W. (2018). Encoding canonical DNA quadruplex structure. *Sci Adv*, 4(8), eaat3007.

Eric, W. and Pascal, A. (2006). *RNA Tertiary Structure*, chapter Nucleic Acids Structure and Mapping. John Wiley & Sons, Ltd.

Esposito, V., Galeone, A., Mayol, L. et al. (2007). A topological classification of G-quadruplex structures. *Nucleos Nucleot Nucl*, 26(8-9), 1155–1159.

Garant, J.-M., Luce, M. J., Scott, M. S. and Perreault, J.-P. (2015). G4RNA: an RNA G-quadruplex database. *Database*, 2015, bav059.

Garant, J.-M., Perreault, J.-P., and Scott, M. S. (2017). Motif independent identification of potential RNA G-quadruplexes by G4RNA screener. *Bioinformatics*, 33(22), 3532–3537.

Garant, J.-M., Perreault, J.-P., and Scott, M. S. (2018). G4RNA screener web server: User focused interface for RNA G-quadruplex prediction. *Biochimie*, 151, 115–118.

Gauthier, J., Vincent, A. T., Charette, S. J. and Derome, N. (2018). A brief history of bioinformatics. *Brief Bioinform*, 20(6), 1981–1996.

Gong, S., Zhang, C., and Zhang, Y. (2019). RNA-align: quick and accurate alignment of RNA 3D structures based on size-independent TM-scoreRNA. *Bioinformatics*, 35(21), 4459–4461.

Hagen, J. B. (2000). The origins of bioinformatics. *Nat Rev Genet*, 1(3), 231–236.

Halder, S. and Bhattacharyya, D. (2013). RNA structure and dynamics: A base pairing perspective. *Prog Biophys Mol Bio*, 113(2), 264–283.

Hoehndorf, R., Batchelor, C., Bittner, T. et al. (2011). The RNA Ontology (RNAO): An ontology for integrating RNA sequence and structure data. *Appl Ontol*, 6(1), 53–89.

Holm, L. (2020). *Using Dali for Protein Structure Comparison*, (pp. 29–42). Springer US: New York, NY.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *J Opt Soc Am A*, 4(4), 629–642.

Hu, B., Guo, H., Zhou, P. and Shi, Z.-L. (2020). Characteristics of SARS-CoV-2 and COVID-19. *Nat Rev Microbiol*, 19(3), 141–154.

Jumper, J., Evans, R., Pritzel, A. et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873), 583–589.

Kabsch, W. (1978). A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr A*, 34(5), 827–828.

Kikin, O., D'Antonio, L., and Bagga, P. S. (2006). QGRS Mapper: a web-based server for predicting G-quadruplexes in nucleotide sequences. *Nucleic Acids Res*, 34(suppl_2), W676–W682.

Kitamura, N., Semler, B. L., Rothberg, P. G. et al. (1981). Primary structure, gene organization and polypeptide expression of poliovirus RNA. *Nature*, 291(5816), 547–553.

Kneller, G. R. (1991). Superposition of Molecular Structures using Quaternions. *Mol Simulat*, 7(1-2), 113–119.

Kryshtafovych, A., Schwede, T., Topf, M. et al. (2019). Critical assessment of methods of protein structure prediction (CASP)-Round XIII. *Proteins*, 87(12), 1011–1020.

Lawson, C. L., Berman, H. M., Chen, L. et al. (2023). The Nucleic Acid Knowledgebase: a new portal for 3D structural information about nucleic acids. *Nucleic Acids Res*, 52(D1), D245–D254.

Leontis, N. and Westhof, E. (2012). Modeling RNA Molecules. In *Nucleic Acids and Molecular Biology* (pp. 5–17). Springer Berlin Heidelberg.

Leontis, N. B. and Westhof, E. (2001). Geometric nomenclature and classification of RNA base pairs. *RNA*, 7(4), 499–512.

Lipman, D. J. and Pearson, W. R. (1985). Rapid and Sensitive Protein Similarity Searches. *Science*, 227(4693), 1435–1441.

Lu, X.-J. (2020). DSSR-enabled innovative schematics of 3D nucleic acid structures with PyMOL. *Nucleic Acids Res*, 48(13), e74–e74.

Lu, X.-J., Bussemaker, H. J., and Olson, W. K. (2015). DSSR: an integrated software tool for dissecting the spatial structure of RNA. *Nucleic Acids Res*, 43(21), e142–e142.

Maiorov, V. N. and Crippen, G. M. (1994). Significance of Root-Mean-Square Deviation in Comparing Three-dimensional Structures of Globular Proteins. *J Mol Biol*, 235(2), 625–634.

Miao, Z., Adamiak, R. W., Antczak, M. et al. (2017). RNA-Puzzles Round III: 3D RNA structure prediction of five riboswitches and one ribozyme. *RNA*, 23(5), 655–672.

Miao, Z., Adamiak, R. W., Antczak, M. et al. (2020). RNA-Puzzles Round IV: 3D structure predictions of four ribozymes and two aptamers. *RNA*, 26(8), 982–995.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3), 443–453.

Nguyen, M. N., Sim, A. Y. L., Wan, Y. et al. (2016). Topology independent comparison of RNA 3D structures using the CLICK algorithm. *Nucleic Acids Res*, 45(1), e5–e5.

Ouzounis, C. A. and Valencia, A. (2003). Early bioinformatics: the birth of a discipline–a personal view. *Bioinformatics*, 19(17), 2176–2190.

Parisien, M., Cruz, J. A., Westhof, É. and Major, F. (2009). New metrics for comparing and assessing discrepancies between RNA 3D structures and models. *RNA*, 15(10), 1875–1885.

Parisien, M. and Major, F. (2008). The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, 452(7183), 51–55.

Plavec, J. (2020). Chapter Thirteen - Quadruplex targets in neurodegenerative diseases. In S. Neidle (Ed.), *Quadruplex Nucleic Acids As Targets For Medicinal Chemistry*, volume 54 of *Annual Reports in Medicinal Chemistry* (pp. 441–483). Academic Press.

Ponty, Y. and Leclerc, F. (2014). Drawing and Editing the Secondary Structure(s) of RNA. In *Methods in Molecular Biology* (pp. 63–100). Springer New York.

Popenda, M., Miskiewicz, J., Sarzynska, J. et al. (2019). Topology-based classification of tetrads and quadruplex structures. *Bioinformatics*, 36(4), 1129–1134.

Rhodes, D. and Lipps, H. J. (2015). G-quadruplexes and their regulatory roles in biology. *Nucleic Acids Res*, 43(18), 8627–8637.

Shindyalov, I. N. and Bourne, P. E. (1998). Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng Des Sel*, 11(9), 739–747.

Spiegel, J., Adhikari, S., and Balasubramanian, S. (2020). The structure and function of DNA G-quadruplexes. *Trends Chem*, 2(2), 123–136.

Šponer, J. E., Špačková, N., Leszczynski, J. and Šponer, J. (2005). Principles of RNA Base Pairing: Structures and Energies of the Trans Watson-Crick/Sugar Edge Base Pairs. *J Phys Chem B*, 109(22), 11399–11410.

Swinnen, B., Robberecht, W., and Bosch, L. V. D. (2019). RNA toxicity in non-coding repeat expansion disorders. *EMBO J*, 39(1).

Townshend, R. J. L., Eismann, S., Watkins, A. M. et al. (2021). Geometric deep learning of RNA structure. *Science*, 373(6558), 1047–1051.

Varani, G. and McClain, W. H. (2000). The G·U wobble base pair. *EMBO Rep*, 1(1), 18–23.

Varshney, D., Spiegel, J., Zyner, K. et al. (2020). The regulation and functions of DNA and RNA G-quadruplexes. *Nat Rev Mol Cell Bio*, 21(8), 459–474.

Watson, J. D. and Crick, F. H. C. (1974). Molecular structure of nucleic acids: a structure for Deoxyribose Nucleic Acid. *Nature*, 248(5451), 765–765.

Webba da Silva, M. (2007). Geometric formalism for DNA quadruplex folding. *Chemistry*, 13(35), 9738–9745.

Webba da Silva, M., Trajkovski, M., Sannohe, Y. et al. (2009). Design of a G-quadruplex topology through glycosidic bond angles. *Ange Chem Int Edit*, 48(48), 9167–9170.

Westhof, E., Masquida, B., and Jossinet, F. (2011). Predicting and modeling RNA architecture. *Csh Perspect Biol*, 3(2), a003632–a003632.

Wiedemann, J. and Miłostan, M. (2017). StructAnalyzer - a tool for sequence vs. structure similarity analysis. *Acta Biochim Pol*, 63(4), 753—-757.

Zemora, G. and Waldsich, C. (2010). RNA folding in living cells. *RNA Biol*, 7(6), 634–641.

Zhang, N., Gorin, A., Majumdar, A. et al. (2001). V-shaped scaffold: a new architectural motif identified in an A x (G x G x G x G) pentad-containing dimeric DNA quadruplex involving stacked G(anti) x G(anti) x G(anti) x G(syn) tetrads. *J Mol Biol*, 311(5), 1063–1079.

Zhang, Y. and Skolnick, J. (2005). TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*, 33(7), 2302–2309.

Zheng, J., Xie, J., Hong, X. and Liu, S. (2019). RMalign: an RNA structural alignment tool based on a novel scoring function RMscore. *BMC Genomics*, 20(1), 276.

Zok, T., Popenda, M., and Szachniuk, M. (2013). MCQ4Structures to compute similarity of molecule structures. *Cent Eur J Oper Res*, 22(3), 457–473.

Zok, T., Popenda, M., and Szachniuk, M. (2020). ElTetrado: a tool for identification and classification of tetrads and quadruplexes. *BMC Bioinformatics*, 21(1), 40.

Published online 19 November 2021 *Nucleic Acids Research, 2022, Vol. 50, Database issue* ***D253–D258***
https://doi.org/10.1093/nar/gkab1118

ONQUADRO: a database of experimentally determined quadruplex structures

Tomasz Zok[1], Natalia Kraszewska[1], Joanna Miskiewicz[1], Paulina Pielacinska[1], Michal Zurkowski[1] and Marta Szachniuk[1,2,*]

[1]Institute of Computing Science, Poznan University of Technology, 60-965 Poznan, Poland and [2]Institute of Bioorganic Chemistry, Polish Academy of Sciences, 61-704 Poznan, Poland

Received August 11, 2021; Revised October 22, 2021; Editorial Decision October 22, 2021; Accepted October 25, 2021

ABSTRACT

ONQUADRO is an advanced database system that supports the study of the structures of canonical and non-canonical quadruplexes. It combines a relational database that collects comprehensive information on tetrads, quadruplexes, and G4-helices; programs to compute structure parameters and visualise the data; scripts for statistical analysis; automatic updates and newsletter modules; and a web application that provides a user interface. The database is a self-updating resource, with new information arriving once a week. The preliminary data are downloaded from the Protein Data Bank, processed, annotated, and completed. As of August 2021, ONQUADRO contains 1,661 tetrads, 518 quadruplexes, and 30 G4-helices found in 467 experimentally determined 3D structures of nucleic acids. Users can view and download their description: sequence, secondary structure (dot-bracket, classical diagram, arc diagram), tertiary structure (ball-and-stick, surface or vdw-ball model, layer diagram), planarity, twist, rise, chi angle (value and type), loop characteristics, strand directionality, metal ions, ONZ, and Webba da Silva classification (the latter by loop topology and tetrad combination), origin structure ID, assembly ID, experimental method, and molecule type. The database is freely available at https://onquadro.cs.put.poznan.pl/. It can be used on both desktop computers and mobile devices.

INTRODUCTION

G-quadruplexes (G4s) are unique structures folded in G-rich nucleic acids (1,2), found in eukaryotic, prokaryotic, and viral genomes (1,3,4). In biological processes, they play crucial regulatory roles by participating in telomere maintenance, the regulation of gene expression, DNA replication, etc. (2,3,5,6). A recent hypothesis, coined as the quadruplex world, suggests that G4s may have been the molecules to initiate life on Earth (7). It takes its cue from the ability of guanines to form stable G-tetrads, the basic building units of a quadruplex. In a tetrad, four guanines arranged in the same plane connect via hydrogen bonds such that each acts as a donor of two hydrogen bonds at the Watson-Crick edge and an acceptor at the Hoogsteen edge (1,8,9). A complete G4 is assembled from at least two G-tetrads stacked one above another, and is stabilised by monovalent cations located in the ion channel (5,9). The stacking interactions of G-tetrads occurring independently of backbone connectivity define a G4-helix (10).

The general definition of what constitutes a quadruplex does not capture the complexity of its structure and the diversity of its features (2,8,9,11–14). Meanwhile, the latter is subjected to various studies, aiming - among others - to associate the motif's conformation with its function, find the relationship between its sequence and the higher-level structure, cluster and classify quadruplexes, learn about and fully describe their properties. We already know that tetrads can form from guanine as well as non-guanine nucleotides (15). Their spatial arrangement to their stacking neighbours is diverse, defined by the rise and twist parameters. The topologies of secondary structures differ in both the tetrad and the quadruplex set, as reflected in the ONZ classification (16). They are influenced by the number of strands contributing to the motif, their lengths, and directionality. Strands may form loops, which can be a part of the quadruplex - cf. Webba da Silva formalism (17). The list of analysed attributes also includes the glycosidic bond angles, the groove width, the number of stacked tetrads, or G-tract continuity, and is probably not yet complete (11).

In the past decade, the unique structure of the quadruplex has focused the attention of many researchers, especially in medical sciences. G4s have become therapeutic targets, that is, for cancer and antiviral treatment (18–20). In the latter case, increased interest in targeting G-quadruplexes in viral genomes was prompted by the COVID-19 pandemic. The frequency and localisation of putative quadruplex sequences in different viral taxa, G4-binding viral domains,

[*]To whom correspondence should be addressed. Tel: +48 616653030; Fax: +48 618771525; Email: marta.szachniuk@cs.put.poznan.pl

and the potential of G4s as viral biosensors were investigated (20–25). These and other quadruplex studies provided a wealth of data for collection, organisation, and further analysis (26–28). It has initiated the development of computational methods and bioinformatics tools dedicated to G4s. Most of these deal with sequence data storage and processing (15,29–35). A few address higher-level structures (10,36–39), including databases that store G4-related data (36,40,41) - none, however, collects complete information about quadruplex structures at all levels of their organisation.

ONQUADRO is a new comprehensive database system that collects and shares data on tetrads, quadruplexes, and G4-helices, whose three-dimensional structures have been determined experimentally. Baseline data are regularly downloaded from the Protein Data Bank (42) and supplemented with parameters computed by specialised procedures of the system's engine. The incorporated programs prepare visualisations of the secondary and tertiary structure models of each motif. The analytical module generates statistics of the distribution of the structural parameters in the set of tetrads and quadruplexes. The system allows users to subscribe to a newsletter about all database newcomers. ONQUADRO, designed for use on desktop computers and mobile devices, is freely available at https://onquadro.cs.put.poznan.pl/.

METHOD OUTLINE

Every Thursday, the update module of ONQUADRO connects to the PDB FTP site and searches for new information about nucleic acids (including protein-nucleic acid complexes). Next, it queries PDBe (43) for biological assemblies to associate them with the items found. The module creates a list with identifiers of newly added, modified, or deleted structures containing tetrads, quadruplexes, and G4-helices. Changes to the ONQUADRO database are made from the list of modified and deleted structures; entries for new motifs are created and added. The process of new data preparation takes place in several steps (Figure 1).

For each quadruplex, it derives the secondary structure and prepares its representation in two-line extended dot-bracket notation, computes the rise and twist parameters, identifies the number of contributing strands and tetrads, determines the strand direction, finds loops to calculate their lengths and types, classifies it according to Webba da Silva formalism based on its loop topology and tetrad combination (17), and assigns an ONZ class from the secondary structure topology (16). If the nucleic acid contains metal ions, the procedure determines their position relative to the quadruplex. Then, it describes every tetrad by planarity, chi angle value and type, ONZ class, and tetrad combination. In the next step, graphical models of the secondary and tertiary structure of every motif are prepared. These include a classical diagram, arc diagram, layer diagram, and 3D molecule models. After calculating all the parameters and preparing graphical models, the system populates the database and maintains the relationships between the entries.

Once the database is updated, the statistical analysis module generates graphs and tables of the data distribution.

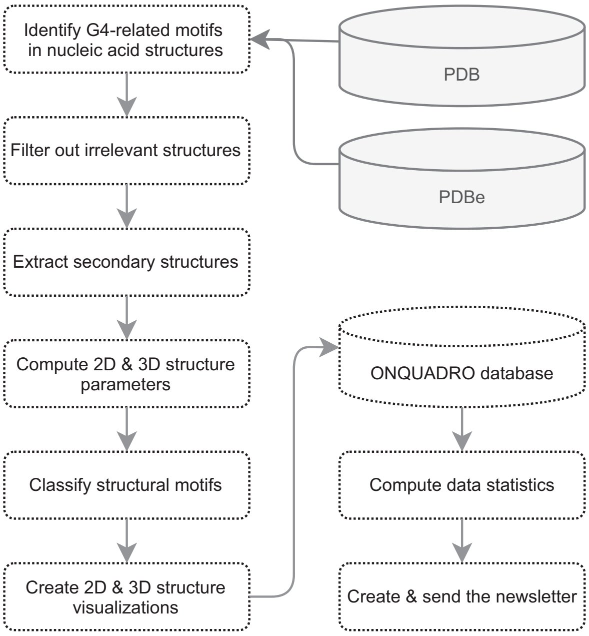

Identify G4-related motifs in nucleic acid structures
Filter out irrelevant structures
Extract secondary structures
Compute 2D & 3D structure parameters
Classify structural motifs
Create 2D & 3D structure visualizations
PDB
PDBe
ONQUADRO database
Compute data statistics
Create & send the newsletter

Figure 1. Data flow during the weekly ONQUADRO update.

Statistics available for G4s are (i) the number of quadruplexes as a function of the number of constituent tetrads; (ii) the abundance of the set of uni-, bi-, and tetramolecular quadruplexes; (iii) ONZ class coverage by uni-, bi-, and tetramolecular quadruplexes; (iv) the geometric class distribution based on glycosidic bond angles and loop topology; (v) the loop length distribution in the subsets of lateral, propeller and diagonal loops and (vi) the distribution of the twist and rise values. Statistics prepared for tetrads include: (i) the distribution of tetrads concerning their sequence and molecule type; (ii) the coverage of ONZ classes by uni-, bi-, and tetramolecular tetrads; (iii) the chi angle value distribution in the ONZ classes; (iv) ONZ class coverage by ions and (v) the planarity value distribution in the tetrad set.

Finally, the system creates a hypertext newsletter listing all changes to the database and sends it to subscribers.

IMPLEMENTATION

The ONQUADRO system consists of the database, web application, and computational engine. It runs on a quad-core machine with 8GB RAM in a Ubuntu GNU/Linux environment, hosted and maintained by the Institute of Computing Science, Poznan University of Technology.

Database

ONQUADRO has been developed as a relational database in PostgreSQL. It is composed of tables that correspond to PDB structures, G4-helices, quadruplexes, tetrads, tetrad pairs, base pairs, nucleotides, tracts, loops, and ions. The database stores the following information about every nucleic acid structure: PDB ID, assembly ID, experimental method, resolution, deposition, release and revision

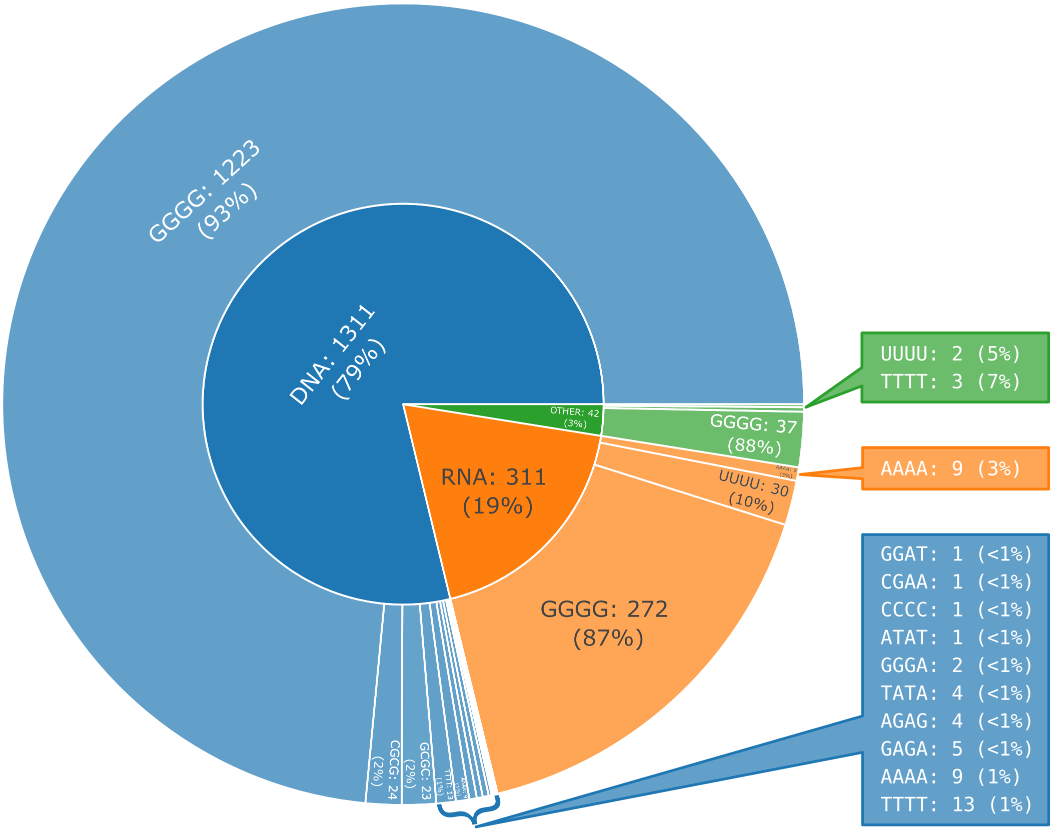


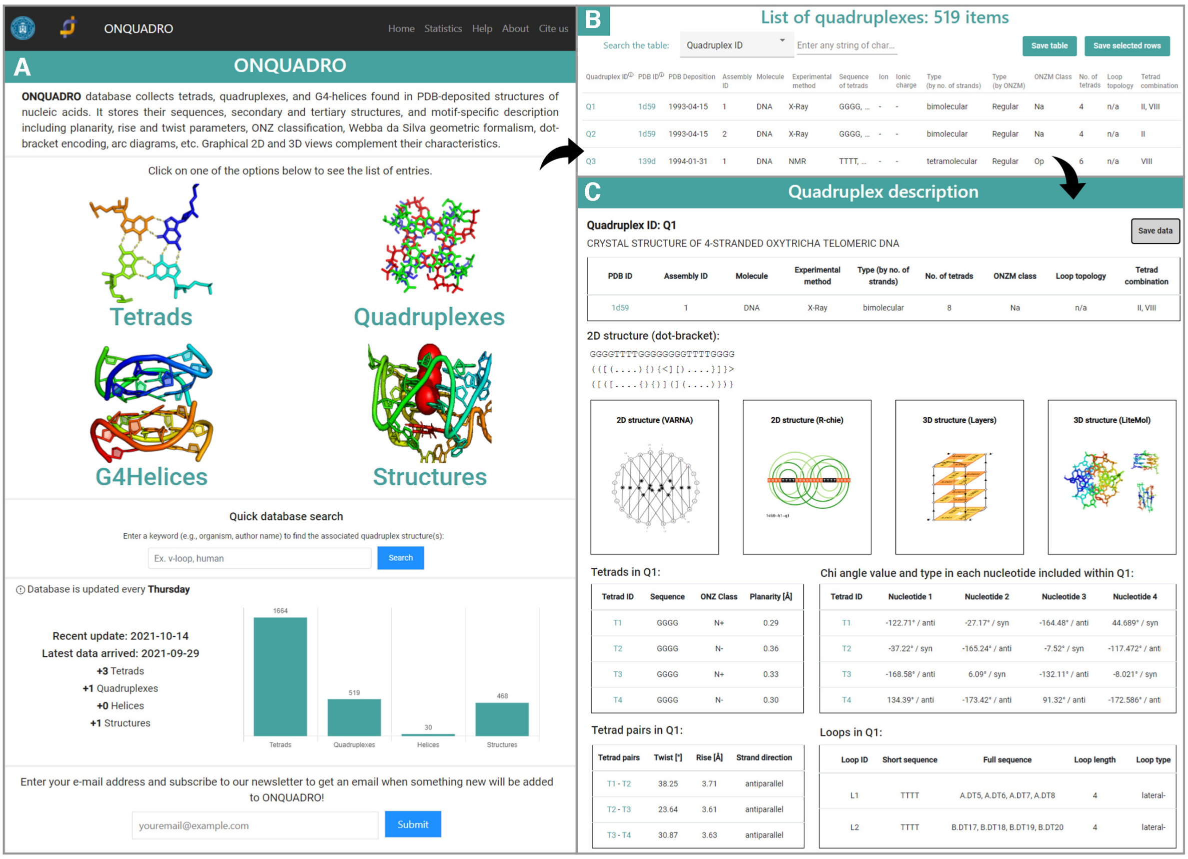
Figure 2. Example statistics generated by ONQUADRO: quadruplexes by the number of constituent tetrads.

dates, molecule name, a secondary structure diagram, and a 3D structure. The secondary and tertiary structures are collected separately for tetrads, quadruplexes, and G4-helices. Additionally, the database contains data on the ONZ class, type (uni-, bi-, or tetramolecular), loops, and ions for quadruplexes, ONZ class and planarity for tetrads, strand direction, rise and twist for tetrad pairs, stericity and edge names for base pairs, model, chain, and glycosidic bond for nucleotides.

All sequences in ONQUADRO are coded in a one-letter format, in the 5′–3′ direction. The secondary structures of tetrads, quadruplexes, and G4-helices are represented using dot-bracket notation - an unpaired nucleotide corresponds to a dot and a base pair to a pair of opening and closing brackets. Since in the considered motifs, each nucleotide pairs with two others, basic dot-bracket notation is not sufficient to unambiguously encode the secondary structure of a tetrad, so for a quadruplex or G4-helix. Therefore, in (16), we introduced a two-line dot-bracket and used an extended set of brackets to label paired nucleotides. This includes parentheses ( ), square brackets [ ], curly brackets { } and angle brackets ⟨ ⟩. An arc diagram representing the secondary structure is also adjusted to unambiguously reflect all pairings. It is associated with dot-bracket notation - the top of the diagram corresponds to the first line of the dot-bracket notation, and the bottom part corresponds to the second line (16). The secondary structure of every motif is also visualised in a classical diagram. The 3D structure is represented by a layer diagram and three molecular models (ball-and-stick, surface, vdw-balls).

Computational engine

The computational engine is composed of scripts utilising in-house and third-party procedures, responsible for data collection, quadruplex identification, computation of structure parameters, secondary structure annotation, visualisation of the secondary and tertiary structure models, database queries, generation of statistics, and newsletter preparation. DSSR (`--pair-only` mode) (36) and ElTetrado (39) functionalities are applied to identify quadruplexes, tetrads, and G4-helices in nucleic acid structures. Procedures from ElTetrado (39) and the BioCommons library (44) compute a variety of structure parameters. The VARNA-based routine (45) creates a classical diagram of the secondary structure. The R-Chie-driven function (46) produces a top-down arc diagram. The embedded LiteMol (47) module generates models (ball-and-stick, surface, vdw-balls) of the three-dimensional structure. The Python script draws a layer diagram of the quadruplex based on the data in JSON format obtained from ElTetrado. Every nucleotide in the diagram is colour-coded – yellow indicates the anti conformation, orange is for syn. The script optimises the quadruplex position in three-dimensional space to get a



ONQUADRO

ONQUADRO database collects tetrads, quadruplexes, and G4-helices found in PDB-deposited structures of nucleic acids. It stores their sequences, secondary and tertiary structures, and motif-specific description including planarity, rise and twist parameters, ONZ classification, Webba da Silva geometric formalism, dot-bracket encoding, arc diagrams, etc. Graphical 2D and 3D views complement their characteristics.

Click on one of the options below to see the list of entries.

Quick database search

Enter a keyword (e.g., organism, author name) to find the associated quadruplex structure(s):

Ex. v-loop, human

ⓘ Database is updated every **Thursday**

Recent update: 2021-10-14
Latest data arrived: 2021-09-29
+3 Tetrads
+1 Quadruplexes
+0 Helices
+1 Structures

Enter your e-mail address and subscribe to our newsletter to get an email when something new will be added to ONQUADRO!

youremail@example.com

List of quadruplexes: 519 items

Search the table: Quadruplex ID ▾ Enter any string of char...

Quadruplex ID	PDB ID	PDB Deposition	Assembly ID	Molecule	Experimental method	Sequence of tetrads	Ion	Ionic charge	Type (by no. of strands)	Type (by ONZM)	ONZM Class	No. of tetrads	Loop topology	Tetrad combination
Q1	1d59	1993-04-15	1	DNA	X-Ray	GGGG, ...	-	-	bimolecular	Regular	Na	4	n/a	II, VIII
Q2	1d59	1993-04-15	2	DNA	X-Ray	GGGG, ...	-	-	bimolecular	Regular	Na	4	n/a	II
Q3	139d	1994-01-31	1	DNA	NMR	TTTT, ...	-	-	tetramolecular	Regular	Op	6	n/a	VIII

Quadruplex description

Quadruplex ID: Q1

CRYSTAL STRUCTURE OF 4-STRANDED OXYTRICHA TELOMERIC DNA

PDB ID	Assembly ID	Molecule	Experimental method	Type (by no. of strands)	No. of tetrads	ONZM class	Loop topology	Tetrad combination
1d59	1	DNA	X-Ray	bimolecular	8	Na	n/a	II, VIII

2D structure (dot-bracket):

```
GGGGTTTTGGGGGGGGTTTTGGGG
(([(....){){<][)....}]}>
([([....{){)](](....)})}
```

2D structure (VARNA)

2D structure (R-chie)

3D structure (Layers)

3D structure (LiteMol)

Tetrads in Q1:

Tetrad ID	Sequence	ONZ Class	Planarity [Å]
T1	GGGG	N+	0.29
T2	GGGG	N-	0.36
T3	GGGG	N+	0.33
T4	GGGG	N-	0.30

Chi angle value and type in each nucleotide included within Q1:

Tetrad ID	Nucleotide 1	Nucleotide 2	Nucleotide 3	Nucleotide 4
T1	-122.71° / anti	-27.17° / syn	-164.48° / anti	44.689° / syn
T2	-37.22° / syn	-165.24° / anti	-7.52° / syn	-117.472° / anti
T3	-168.58° / anti	6.09° / syn	-132.11° / anti	-8.021° / syn
T4	134.39° / anti	-173.42° / anti	91.32° / anti	-172.586° / anti

Tetrad pairs in Q1:

Tetrad pairs	Twist [°]	Rise [Å]	Strand direction
T1 - T2	38.25	3.71	antiparallel
T2 - T3	23.64	3.61	antiparallel
T3 - T4	30.87	3.63	antiparallel

Loops in Q1:

Loop ID	Short sequence	Full sequence	Loop length	Loop type
L1	TTTT	A.DT5, A.DT6, A.DT7, A.DT8	4	lateral-
L2	TTTT	B.DT17, B.DT18, B.DT19, B.DT20	4	lateral-

Figure 3. ONQUADRO interface: (**A**) main page, (**B**) quadruplex table and (**C**) quadruplex details based on 1D59 structure.

clear view of the G4, with the least number of crossing strands. The optimisation algorithm has been implemented in C++. Statistics are generated using the script in R and the Plotly library. They self-update whenever new entries appear in the database (Figure 2).

Web application

The web application provides an interface for the ONQUADRO system. The client application has been created using the Angular framework and Bootstrap styling sheet; the server is implemented in C#. The client and the server communicate via REST API.

Figure 3 presents screenshots of the ONQUADRO system. On the homepage (https://onquadro.cs.put.poznan.pl/), users can see brief information about the database resources and the latest update. From this page, users can access the sets of tetrads, quadruplexes, G4-helices or structures. The selected subpage displays a list of items with a basic description. Some data are clickable - they allow detailed structural information to be viewed about the selected element or link to the corresponding page in the Protein Data Bank. The table can be sorted (ascending or descending) for the contents of any column by clicking on the column header. Users can search the list of items by using the *Search the table* option and typing the string of interest. Searching runs in real-time. The element counter at the top of the page shows how many elements contain the queried string. These elements are displayed in the table. Users can save the content of each table as a whole (*Save table* button) or a selected part (*Save selected rows* button after clicking on the check-boxes in the rightmost column). Tabular data (from any subpage) are downloaded in a CSV file, structure visualisations can be saved in SVG format.

The *Statistics* option in the menu bar leads to a page listing statistics for tetrads and quadruplexes. User-selected stats are displayed in graphical (pie chart, tree map, or bar plot) and tabular form. Plots can be saved in HTML format. They are interactive - upon clicking the plot, users can enlarge fragments and see selected parts of the data.

CONCLUSIONS

ONQUADRO gathers information about all tetrads, quadruplexes, and G4-helices found in experimentally determined nucleic acid structures deposited in the Protein Data Bank (42). The system's computational engine combines self-developed procedures to annotate these motifs, derive their secondary structures, classify them according to geometric formalism (17) and topological ONZ nomenclature (16), represent the secondary structure in dot-bracket notation and a specially adjusted top-down arc diagram, draw a 3D model in a schematic layer diagram, and trigger statistics. Some are G4-adapted routines applied in our previously released tools; others are brand new and have not yet been published (e.g. automatic creation of layer diagrams - a much-needed function in the research community). The user-friendly interface allows browsing of the database contents divided into four subsets (tetrads, quadruplexes, G4-helices, PDB structures), searching and sorting the data by various parameters and keywords, displaying and downloading detailed structural information on selected motifs, viewing and downloading statistics in graphical and textual form. ONQUADRO is a unique online resource that takes a comprehensive approach to collecting and sharing quadruplex information. We hope it will facilitate the study of G4 structures and their modelling *in silico* - a great challenge for modern structural bioinformatics.

DATA AVAILABILITY

ONQUADRO is a continuously maintained, weekly self-updating resource available at https://onquadro.cs.put.poznan.pl. No registration or login is required to access the data and take full advantage of the system's functionality.

FUNDING

National Science Center, Poland [2019/35/B/ST6/03074 to M.S.]; statutory funds of Poznan University of Technology; Institute of Bioorganic Chemistry, PAS. Funding for open access charge: Poznan University of Technology.

Conflict of interest statement. None declared.

REFERENCES

1. Malgowska,M., Czajczynska,K., Gudanis,D., Tworak,A. and Gdaniec,Z. (2016) Overview of RNA G-quadruplex structures. *Acta Biochim. Pol.*, **63**, 609–621.
2. Kwok,C. and Merrick,C. (2017) G-Quadruplexes: prediction, characterization, and biological application. *Trends Biotechnol.*, **35**, 997–1013.
3. Joachimi,A., Benz,A. and Hartig,J. (2009) A comparison of DNA and RNA quadruplex structures and stabilities. *Bioorg. Med. Chem.*, **17**, 6811–6815.
4. Antonio,M., Ponjavic,A., Radzevicius,A., Ranasinghe,R., Catalano,M., Zhang,X., Shen,J., Needham,L., Lee,S., Klenerman,D. *et al.* (2020) Single-molecule visualization of DNA G-quadruplex formation in live cells. *Nat. Chem.*, **12**, 832–837.
5. Webba da Silva,M., Trajkovski,M., Sannohe,Y., Ma'ni Hessari,N., Sugiyama,H. and Plavec,J. (2009) Design of a G-quadruplex topology through glycosidic bond angles. *Angew. Chem. Int. Ed. Engl.*, **48**, 9167–9170.
6. Kolesnikova,S. and Curtis,E. (2019) Structure and Function of Multimeric G-Quadruplexes. *Molecules*, **24**, 3074.
7. Kankia,B. (2021) Quadruplex world. *Orig. Life Evol. Biosphys.*, **51**, 273–286.
8. Dvorkin,S., Karsisiotis,A. and Webba da Silva,M. (2018) Encoding canonical DNA quadruplex structure. *Sci. Adv.*, **4**, eaat3007.
9. Spiegel,J., Adhikari,S. and Balasubramanian,S. (2020) The structure and function of DNA G-quadruplexes. *Trends Chem.*, **2**, 123–136.
10. Lu,X. (2020) DSSR-enabled innovative schematics of 3D nucleic acid structures with PyMOL. *Nucleic Acids Res.*, **48**, e74.
11. Burge,S., Parkinson,G., Hazel,P., Todd,A. and Neidle,S. (2006) Quadruplex DNA: sequence, topology and structure. *Nucleic Acids Res.*, **34**, 5402–5415.
12. Cang,X., Sponer,J. and Cheatham,T. (2011) Explaining the varied glycosidic conformational, G-tract length and sequence preferences for anti-parallel G-quadruplexes. *Nucleic Acids Res.*, **39**, 4499–4512.
13. Mukundan,V. and Phan,A. (2013) Bulges in G-quadruplexes: broadening the definition of G-quadruplex-forming sequences. *J. Am. Chem. Soc.*, **135**, 5017–5028.
14. Lightfoot,H., Hagen,T., Tatum,N. and Hall,J. (2019) The diverse structural landscape of quadruplexes. *FEBS Lett.*, **593**, 2083–2102.
15. Miskiewicz,J., Sarzynska,J. and Szachniuk,M. (2021) How bioinformatics resources work with G4 RNAs. *Brief Bioinform.*, **22**, bbaa201.
16. Popenda,M., Miskiewicz,J., Sarzynska,J., Zok,T. and Szachniuk,M. (2020) Topology-based classification of tetrads and quadruplex structures. *Bioinformatics*, **36**, 1129–1134.
17. Webba da Silva,M. (2007) Geometric formalism for DNA quadruplex folding. *Chemistry*, **13**, 9738–9745.
18. Carvalho,J., Mergny,J., Salgado,G., Queiroz,J. and Cruz,C. (2020) G-quadruplex, friend or foe: the role of the G-quartet in anticancer strategies. *Trends Mol. Med.*, **26**, 848–861.
19. Hansel-Hertsch,R., Simeone,A., Shea,A., Hui,W., Zyner,K., Marsico,G., Rueda,O., Bruna,A., Martin,A., Zhang,X. *et al.* (2020) Landscape of G-quadruplex DNA structural regions in breast cancer. *Nat. Genet.*, **52**, 878–883.
20. Panera,N., Tozzi,A. and Alisi,A. (2020) The G-quadruplex/helicase world as a potential antiviral approach against COVID-19. *Drugs*, **80**, 941–946.
21. Lavezzo,E., Berselli,M., Frasson,I., Perrone,R., Palu,G., Brazzale,A., Richter,S. and Toppo,S. (2018) G-quadruplex forming sequences in the genome of all known human viruses: A comprehensive guide. *PLoS Comput. Biol.*, **14**, e1006675.
22. Ji,D., Juhas,M., Tsang,C., Kwok,C., Li,Y. and Zhang,Y. (2020) Discovery of G-quadruplex-forming sequences in SARS-CoV-2. *Brief Bioinform.*, **22**, 1150–1160.
23. Wang,S., Min,Y., Wang,J., Liu,C., Fu,B., Wu,F., Wu,L., Qiao,Z., Song,Y., Xu,G. *et al.* (2016) A highly conserved G-rich consensus sequence in hepatitis C virus core gene represents a new antihepatitis C target. *Sci. Adv.*, **2**, e1501535.
24. Tan,J., Vonrhein,C., Smart,O., Bricogne,G., Bollati,M., Kusov,Y., Hansen,G., Mesters,J., Schmidt,C. and Hilgenfeld,R. (2009) The SARS-unique domain (SUD) of SARS coronavirus contains two macrodomains that bind G-quadruplexes. *PLoS Pathog.*, **5**, e1000428.
25. Xi,H., Juhas,M. and Zhang,Y. (2020) G-quadruplex based biosensor: a potential tool for SARS-CoV-2 detection. *Biosens Bioelectron.*, **167**, 112494.
26. Gudanis,D., Popenda,L., Szpotkowski,K., Kierzek,R. and Gdaniec,Z. (2016) Structural characterization of a dimer of RNA duplexes composed of 8-bromoguanosine modified CGG trinucleotide repeats: a novel architecture of RNA quadruplexes. *Nucleic Acids Res.*, **44**, 2409–2416.
27. Andralojc,W., Malgowska,M., Sarzynska,J., Pasternak,K., Szpotkowski,K., Kierzek,R. and Gdaniec,Z. (2018) Unraveling the structural basis for the exceptional stability of RNA G-quadruplexes capped by a uridine tetrad at the 3′ terminus. *RNA*, **25**, 121–134.
28. Frelih,T., Wang,B., Plavec,J. and Sket,P. (2020) Pre-folded structures govern folding pathways of human telomeric G-quadruplexes. *Nucleic Acids Res.*, **48**, 2189–2197.
29. Wong,H., Stegle,O., Rodgers,S. and Huppert,J. (2010) A toolbox for predicting G-quadruplex formation and stability. *J. Nucleic Acids*, **2010**, 564946.
30. Garant,J., Luce,M., Scott,M. and Perreault,J. (2015) G4RNA: an RNA G-quadruplex database. *Database*, **2015**, bav059.
31. Bedrat,A., Lacroix,L. and Mergny,J. (2016) Re-evaluation of G-quadruplex propensity with G4Hunter. *Nucleic Acids Res.*, **4**, 1746–1759.
32. Sahakyan,A., Chambers,V., Marsico,G., Santner,T., Di Antonio,M. and Balasubramanian,S. (2017) Machine learning model for sequence-driven DNA G-quadruplex formation. *Sci. Rep.*, **7**, 14535.
33. Ge,F., Wang,Y., Li,H., Zhang,R., Wang,X., Li,Q., Liang,Z. and Yang,L. (2019) Plant-GQ: an integrative database of G-quadruplex in plant. *J. Comput. Biol.*, **26**, 1013–1019.
34. Lombardi,E. and Londono-Vallejo,A. (2020) A guide to computational methods for G-quadruplex prediction. *Nucleic Acids Res.*, **48**, 1603–1603.
35. Kudla,M., Gutowska,K., Synak,J., Weber,M., Bohnsack,K., Lukasiak,P., Villmann,T., Blazewicz,J. and Szachniuk,M. (2020) Virxicon: a lexicon of viral sequences. *Bioinformatics*, **36**, 5507–5513.

36. Lu,X., Bussemaker,H. and Olson,W. (2015) DSSR: an integrated software tool for dissecting the spatial structure of RNA. *Nucleic Acids Res.*, **43**, e142.
37. Rybarczyk,A., Szostak,N., Antczak,M., Zok,T., Popenda,M., Adamiak,R., Blazewicz,J. and Szachniuk,M. (2015) New in silico approach to assessing RNA secondary structures with non-canonical base pairs. *BMC Bioinformatics*, **16**, 276.
38. Patro,L., Kumar,A., Kolimi,N. and Rathinavelan,T. (2017) 3d-NuS: a web server for automated modeling and visualization of non-canonical 3-dimensional nucleic acid structures. *J. Mol. Biol.*, **429**, 2438–2448.
39. Zok,T., Popenda,M. and Szachniuk,M. (2020) ElTetrado: a tool for identification and classification of tetrads and quadruplexes. *BMC Bioinformatics*, **21**, 40.
40. Li,Q., Xiang,J., Yang,Q., Sun,H., Guan,A. and Tang,Y. (2012) G4LDB: a database for discovering and studying G-quadruplex ligands. *Nucleic Acids Res.*, **41**, D1115–D1123.
41. Mishra,S., Tawani,A., Mishra,A. and Kumar,A. (2016) G4IPDB: a database for G-quadruplex structure forming nucleic acid interacting proteins. *Sci. Rep.*, **6**, 38144.
42. Berman,H., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T., Weissig,H., Shindyalov,I. and Bourne,P. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
43. Velankar,S., Alhroub,Y., Best,C., Caboche,S., Conroy,M., Dana,J., Montecelo,M., van Ginkel,G., Golovin,A., Gore,S. *et al.* (2011) PDBe: Protein Data Bank in Europe. *Nucleic Acids Res.*, **42**, D445–D452.
44. Zok,T. (2021) BioCommons: a robust Java library for RNA structural bioinformatics. *Bioinformatics*, **37**, 2766–2767.
45. Darty,K., Denise,A. and Ponty,Y. (2009) VARNA: interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, **25**, 1974–1975.
46. Lai,D., Proctor,J., Zhu,J. and Meyer,I. (2012) R-CHIE: a web server and R package for visualizing RNA secondary structures. *Nucleic Acids Res.*, **40**, e95.
47. Sehnal,D., Deshpande,M., Varekova,R., Mir,S., Berka,K., Midlik,A., Pravda,L., Velankar,S. and Koca,J. (2017) LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data. *Nat. Methods*, **14**, 1121–1122.

Bioinformatics, 38(15), 2022, 3835–3836
https://doi.org/10.1093/bioinformatics/btac394
Advance Access Publication Date: 15 June 2022
Applications Note

Structural bioinformatics

DrawTetrado to create layer diagrams of G4 structures

Michal Zurkowski[1], **Tomasz Zok** [1] **and Marta Szachniuk** [1,2,*]

[1]Institute of Computing Science, Poznan University of Technology, 60-965 Poznan, Poland and [2]Institute of Bioorganic Chemistry, Polish Academy of Sciences, 61-704 Poznan, Poland

*To whom correspondence should be addressed.

Associate Editor: Yann Ponty

Received on February 7, 2022; revised on May 13, 2022; editorial decision on June 10, 2022; accepted on June 13, 2022

Abstract

Motivation: Quadruplexes are specific 3D structures found in nucleic acids. Due to the exceptional properties of these motifs, their exploration with the general-purpose bioinformatics methods can be problematic or insufficient. The same applies to visualizing their structure. A hand-drawn layer diagram is the most common way to represent the quadruplex anatomy. No molecular visualization software generates such a structural model based on atomic coordinates.

Results: DrawTetrado is an open-source Python program for automated visualization targeting the structures of quadruplexes and G4-helices. It generates static layer diagrams that represent structural data in a pseudo-3D perspective. The possibility to set color schemes, nucleotide labels, inter-element distances or angle of view allows for easy customization of the output drawing.

Availability and implementation: The program is available under the MIT license at https://github.com/RNApolis/drawtetrado.

Contact: mszachniuk@cs.put.poznan.pl

1 Introduction

Quadruplexes are multilayered motifs occurring in nucleic acid structures. They tend to fold in guanine-rich regions, hence their abbreviated name G4. Every layer forms when four nucleotides arrange on a tetragonal plane and each of them makes pairings with two adjacent ones. Such layout of nucleotides is called a tetrad. Quadruplexes have a multitude of properties described by structural parameters. They include sequence, G-tracts, secondary structure topology, base-pair classification, nucleoside conformations, the number of stacked tetrads, tetrad planarity deviations, rise, twist, right- and left-handedness, torsion angles, the number of nucleic acids strands, strand polarity, type and length of loops, type and position of metal ions, etc. (Jana *et al.*, 2021; Zok *et al.*, 2022).

The complexity and specificity of the quadruplex are easier to understand if we have an appropriate visual model of its structure. However, not all models developed for nucleic acids are equally well suited to represent G4. Therefore, to visualize the secondary structure of tetrad or quadruplex, we introduced a dedicated top-down arc diagram, a two-line dot-bracket and a modified VARNA diagram (c.f. Fig. 1A)—basic versions of these representations could not reflect all base pairs that make up G4 motifs (Darty *et al.*, 2009; Popenda *et al.*, 2020). The 3D structure of G4 can be shown using either of the existing visual models. The visualization type most often used in presentations and scientific publications is a layer diagram (c.f. Fig. 1B). To our knowledge, it cannot be automatically generated by any molecular visualization software. It presents a simplified model of a quadruplex highlighting its selected features (e.g. the number of tetrads, nucleoside conformations, the course of the strand, the presence and types of loops). So far, the only visual model designed for the 3D structure of quadruplexes is cartoon-block schematics (Fig. 1C). These models are generated by DSSR-PyMOL integration and presented as static images of the structure viewed from six perspectives (Lu, 2020).

Here, we present DrawTetrado—an application to create layer diagrams of quadruplexes in DNA and RNA structures. They show the tetrads as a stack, each having four nucleobases colored according to anti or syn conformation. Strand directions are marked with arrows that support a visual identification of individual strands and determination of loop types (lateral, diagonal, propeller, V-shaped). The program automatically optimizes the model layout to give a readable image, even for complex cases like V-loops and G4-helices (quadruplex dimers). It allows customizing the diagrams and saving them in publication-quality SVG files. DrawTetrado is freely available at the GitHub repository.

2 Materials and methods

The DrawTetrado algorithm operates on the following data: G4-helix components, quadruplex components (the number of tetrads), G-tract components, tetrad types according to ONZ classification and nucleotide descriptions (type and conformation). These data are determined from the input 3D structure by the automatically run functions derived from ElTetrado (Zok *et al.*, 2020) and BPNet (Roy and Bhattacharyya, 2022). Then, DrawTetrado creates a layer diagram drawing in a multi-step procedure. At first, the algorithm determines the orientation of each tetrad. Then, it calculates

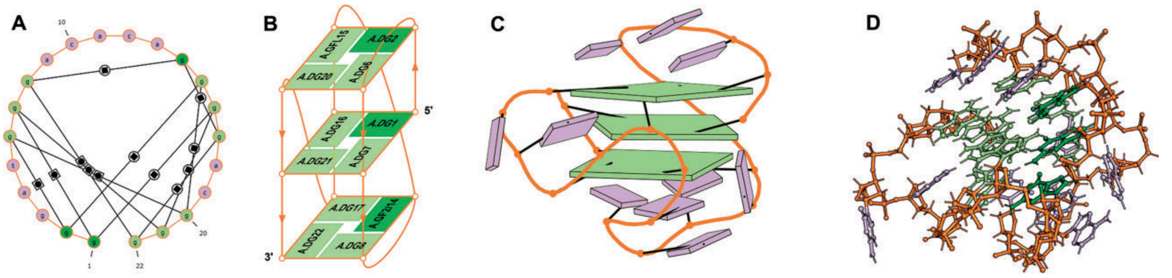


Fig. 1. Example visualizations of quadruplex structure, PDB ID: 6TCG (Haase and Weisz, 2020): (**A**) secondary structure diagram, (**B**) layer diagram from DrawTetrado, (**C**) cartoon-block schematics and (**D**) balls-and-sticks model

and draws the inter-tetrad connections located at the back and on the left-hand side of the diagram. In the next step, the algorithm connects nucleotides from the same layer. On top of this, it superimposes the shapes of nucleotides (parallelograms). Next, it frames the tetrads and draws the remaining inter-tetrad connections (the front and right-hand side ones). Finally, it labels all nucleotides in the diagram.

Connections between the tetrads are approximated by Bezier curves determined from the position of connected points and the curve orientation. The latter follows the polymer chain direction. Each connection is drawn separately. The algorithm distinguishes several types of links depending on the course and position of the curve. It tries to optimize the drawing for readability. Therefore, it prioritizes short vertical connections and applies penalties for the diagonal ones, especially those that run on the front of the diagram. The optimization takes place in the first step of the procedure when the rotation of tetrads is computed.

3 Using DrawTetrado

DrawTetrado works on all operating systems. It is written in Python 3.6+ and utilizes four extra modules—pycairo, svgwrite, orjson and eltetrado. The latter ones are automatically downloaded from the module repository while DrawTetrado installation. The internal optimization routine, implemented in C++, requires Cython and a C++20-compliant compiler. As the input, the program processes PDB and PDBx/mmCIF files. It can also accept JSON files generated by ElTetrado (Zok *et al.*, 2020)—these contain quadruplex structural metadata determined from atomic coordinates, including contact network computed by the BPNet algorithm (Roy and Bhattacharyya, 2022).

The program is run via CLI (Command Line Interface) with one mandatory parameter—input file path—and many optional ones. It outputs layer diagrams in Scalable Vector Graphics (SVG) files. One can further edit them in any vector graphics software without quality loss.

Users can customize the drawing by modifying several parameters in the configuration file. They include the size (side lengths) of the nucleotide-representing parallelogram, the conformation-dependent color of the nucleotide (syn, anti, unrecognized), nucleotide label (label composition, font—typeface, color, size), spacing between nucleotides in the tetrad, the distance between layers (tetrads), the color of the tetrad frame, chain color and the viewing angle. The nucleotide label may consist of a chain identifier, a nucleotide name (short or long) and a nucleotide number. Changes to the configuration file are optional. The default parameters have been optimized to make the drawing readable and colorblind-friendly.

4 Conclusion

Bioinformatics resources are essential for studying biological data. So far, the quadruplex-dedicated ones have mainly focused on collecting and processing PQS (putative quadruplex sequences) (Miskiewicz *et al.*, 2021). A few computational tools target 2D and 3D structures of G4s, including one for the 3D structure visualization (Lu, 2020). DrawTetrado responds to a growing demand for automated visualization of quadruplex structures in the most popular form—a layer diagram. It complements the collection of G4-dedicated tools created by the RNApolis team (Szachniuk, 2019). Since mid-2021, it has worked as a component of the ONQUADRO system (Zok *et al.*, 2022) to visualize experimental PDB-derived quadruplex structures. Until now, it has created visualizations for 36 G4-helices and 599 quadruplexes stored in this database (data as of February 4, 2022). Available as a standalone program, it enables the creation of diagrams for arbitrary experimental and *in silico* G4 models.

Funding

We acknowledge support from Poznan University of Technology (statutory funds), and the National Science Centre, Poland [2019/35/B/ST6/03074].

Conflict of Interest: none declared.

References

Darty,K. *et al.* (2009) Varna: interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, **25**, 1974–1975.

Haase,L. and Weisz,K. (2020) Switching the type of V-loop in sugar-modified G-quadruplexes through altered fluorine interactions. *Chem. Commun. (Camb.)*, **56**, 4539–4542.

Jana,J. *et al.* (2021) Structural motifs and intramolecular interactions in non-canonical G-quadruplexes. *RSC Chem. Biol.*, **2**, 338–353.

Lu,X.-J. (2020) DSSR-enabled innovative schematics of 3D nucleic acid structures with PyMOL. *Nucleic Acids Res.*, **48**, e74.

Miskiewicz,J. *et al.* (2021) How bioinformatics resources work with G4 RNAs. *Brief Bioinform.*, **22**, bbaa201.

Popenda,M. *et al.* (2020) Topology-based classification of tetrads and quadruplex structures. *Bioinformatics*, **36**, 1129–1134.

Roy,P. and Bhattacharyya,D. (2022) Contact networks in RNA: a structural bioinformatics study with a new tool. *J. Comput. Aided Mol. Des.*, **36**, 131–140.

Szachniuk,M. (2019) RNApolis: computational platform for RNA structure analysis. *FCDS*, **44**, 241–257.

Zok,T. *et al.* (2020) ElTetrado: a tool for identification and classification of tetrads and quadruplexes. *BMC Bioinformatics*, **21**, 40.

Zok,T. *et al.* (2022) ONQUADRO: a database of experimentally determined quadruplex structures. *Nucleic Acids Res.*, **50**, D253–D258.

Published online 9 May 2023 *Nucleic Acids Research, 2023, Vol. 51, Web Server issue* ***W607–W612***
https://doi.org/10.1093/nar/gkad346

WebTetrado: a webserver to explore quadruplexes in nucleic acid 3D structures

Bartosz Adamczyk[1], Michal Zurkowski[1], Marta Szachniuk[1,2,*] and Tomasz Zok[1,*]

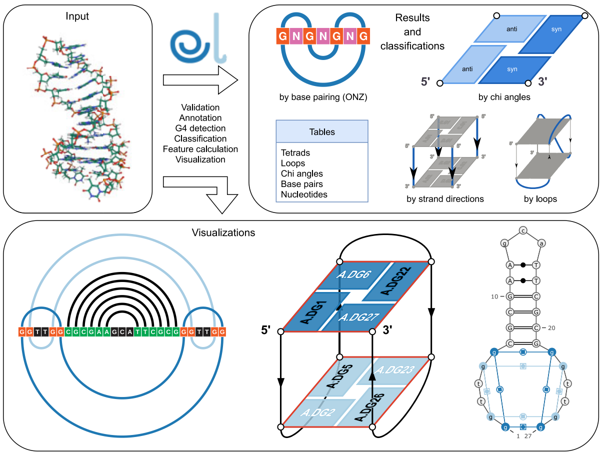
[1]Institute of Computing Science and European Centre for Bioinformatics and Genomics, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland and [2]Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12/14, 61-704 Poznan, Poland

Received March 24, 2023; Revised April 14, 2023; Editorial Decision April 20, 2023; Accepted April 24, 2023

ABSTRACT

Quadruplexes are four-stranded DNA/RNA motifs of high functional significance that fold into complex shapes. They are widely recognized as important regulators of genomic processes and are among the most frequently investigated potential drug targets. Despite interest in quadruplexes, few studies focus on automatic tools that help to understand the many unique features of their 3D folds. In this paper, we introduce WebTetrado, a web server for analyzing 3D structures of quadruplex structures. It has a user-friendly interface and offers many advanced features, including automatic identification, annotation, classification, and visualization of the motif. The program applies to the experimental or *in silico* generated 3D models provided in the PDB and PDBx/mmCIF files. It supports canonical G-quadruplexes as well as non-G-based quartets. It can process unimolecular, bimolecular, and tetramolecular quadruplexes. WebTetrado is implemented as a publicly available web server with an intuitive interface and can be freely accessed at https://webtetrado.cs.put.poznan.pl/.

GRAPHICAL ABSTRACT



INTRODUCTION

Quadruplexes are four-stranded DNA and RNA motifs that form in genomic regions rich in guanine. They are involved in many genomic processes, including transcription, replication, and epigenetic regulation (1). Numerous studies point to their association with the growth and progression of cancer and other diseases. All this makes quadruplexes promising targets in drug design and interesting subjects of structural studies (2–5).

In 2020 Popenda *et al.* proposed a classification scheme derived from base pairing patterns in tetrads (6). They defined three classes, O, N and Z, named after the shape observed in the tetrad visualizations. Each class has clockwise and anticlockwise progression, indicated with + and −, respectively. Next, they proposed classifying a quadruplex as O, N or Z if all of its tetrads are of this type or M (mixed) otherwise. In addition, they added a suffix p, a or h for the parallel, antiparallel, or hybrid orientation of the strands, respectively.

*To whom correspondence should be addressed. Tel: +48 616652999; Fax: +48 618771525; Email: tomasz.zok@cs.put.poznan.pl
Correspondence may also be addressed to Marta Szachniuk. Email: marta.szachniuk@cs.put.poznan.pl

The topologies underlying the classification of quadruplexes and other parameters of their structures can be analyzed using a few computational tools. DSSR (7) was the first to target the detection of G-quadruplexes in 3D structure data saved in PDB and PDBx/mmCIF files and to describe their features. It runs systematically on all entries in the Protein Data Bank and collects motifs found in the DSSR-G4DB database. ElTetrado (8) can identify and analyze G4s and other kinds of tetrads and quadruplexes, classify them, and compute their parameters. It is the core of the computation pipeline running within the ONQUADRO database system (9). The most recent tool for processing atom coordinates in the search for quadruplexes is ASC-G4 (10). It calculates more features than DSSR and ElTetrado, but is limited to unimolecular quadruplexes and supports only the PDB format.

In this paper, we introduce WebTetrado, a web server for analyzing 3D structures of quadruplexes. It has a user-friendly interface and offers many new features compared to its command-line predecessor, ElTetrado. Novelties include dedicated visualizations thanks to tight integration with our advanced tool DrawTetrado (11).

METHOD OUTLINE

The first step in the WebTetrado pipeline (see Figure 1) is to read the input data and the configuration parameters. The front-end feeds these data to the back-end on the basis of the input form on the main page. The validation protocol ensures that the main input is a correct PDB or PDBx/mmCIF file and that all other analysis parameters have viable values. The back-end stores successfully validated inputs in a database and enqueues a computing task. This step involves the generation of a unique identifier, which the front-end embeds in a URL. Initially, the URL displays a loading page with the option to turn on browser notification upon the task's successful completion. Later, the same URL shows the results for the next seven days, after which it expires.

The WebTetrado engine supports parallel processing, so it can handle multiple requests at the same time. The central part of its pipeline starts with reading the configuration metadata from the database. The 3D structure is then loaded and interpreted in terms of its chain, residue, and atom composition. This includes the calculation of the glycosidic bond angle and the classification of each nucleobase as *anti* or *syn*. Next, WebTetrado applies geometrical rules (i.e. constraints on atomic distances, planar and (pseudo)torsion angles) to find stacking and base-base interactions together with their Leontis-Westhof classification. The result of this step allows for the building of a directed graph of nucleotide interactions in which cycles of length four correspond to tetrads in the analyzed structure. This leads to the next step in which the stacking information determined previously is applied on top of the tetrads to locate the N4 helices. Based on chain composition rules, these are divided into distinct quadruplexes, for which WebTetrado traces loop progression and strand connectivity. Moreover, the engine recognizes cations, which play significant roles in quadruplex stability, and proceeds to analyze their proximity to tetrad centers or external sites. Next, the engine classifies the tetrads and quadruplexes according to all its supported schemes and computes quadruplex-related features such as inter-tetrad twist, rise, or planarity deviation. Finally, the quadruplex motif is represented in the two-line dot-bracket format.

These results are stored in the WebTetrado database and a separate drawing task is added to the queue for each supported visualization tool, VARNA (12), R-Chie (13) and DrawTetrado (11). This approach allows for the parallel preparation of all static visualizations. Each drawing task starts by reading the metadata and the computing task's results from the database.

The VARNA-based procedure uses a set of in-house modifications on top of VARNA software to apply custom coloring and Leontis-Westhof visual annotation, making the quadruplex visualization clear. WebTetrado precomputes four variants of VARNA-based visualization: (i) with interactions constituting tetrads only, (ii) with the addition of canonical pairs outside tetrads, (iii) with all non-canonical interactions and (iv) with all canonical and non-canonical interactions.

The R-Chie-based visualization draws arcs above and below the sequence to display two simultaneous interactions for every in-tetrad nucleotide. This is necessary because G-quartets are based on multiplet base pairing patterns (i.e. each in-tetrad nucleotide has two interacting partners). WebTetrado precomputes two R-Chie-based variants, with and without canonical base pairs outside the tetrads. Unlike tetrad-involved interactions, which use distinct colors for every ONZ class, the arcs representing canonical base pairs are black.

The last tool—DrawTetrado—is coupled the most with WebTetrado, as the computing task's results directly influence its working. DrawTetrado prepares a 2.5D view of each G4-helix and quadruplex, showing stacking information, *anti*/*syn* conformation, and loop progression.

WEB APPLICATION

WebTetrado consists of three modules designed to provide flexibility and stability. The service core (engine) is responsible for processing user requests. It is built on top of the lightweight Flask server framework and integrates the ElTetrado tool (8) to identify and process quadruplex data. The next module, the back-end, uses database-driven middleware to manage, queue, and store user requests. It uses the Django web server framework (version 4.1) and the Redis task queue broker, enabling fast processing of concurrent workloads. The engine and the back-end use a Python 3.10 environment with dedicated bioinformatics libraries. They communicate via an OpenAPI-specified interface, which allows automatic validation. The web-accessible front-end is based on TypeScript's React 17 framework, extended with ant-design components. It provides a series of structure visualizations prepared with four incorporated graphical tools: VARNA (12), R-Chie (13), DrawTetrado (11) and Mol* (14). We designed WebTetrado to work on any modern web browser, either mobile or desktop. It is hosted and maintained by the Institute of Computing Science, Poznan University of Technology, using the Docker container service.

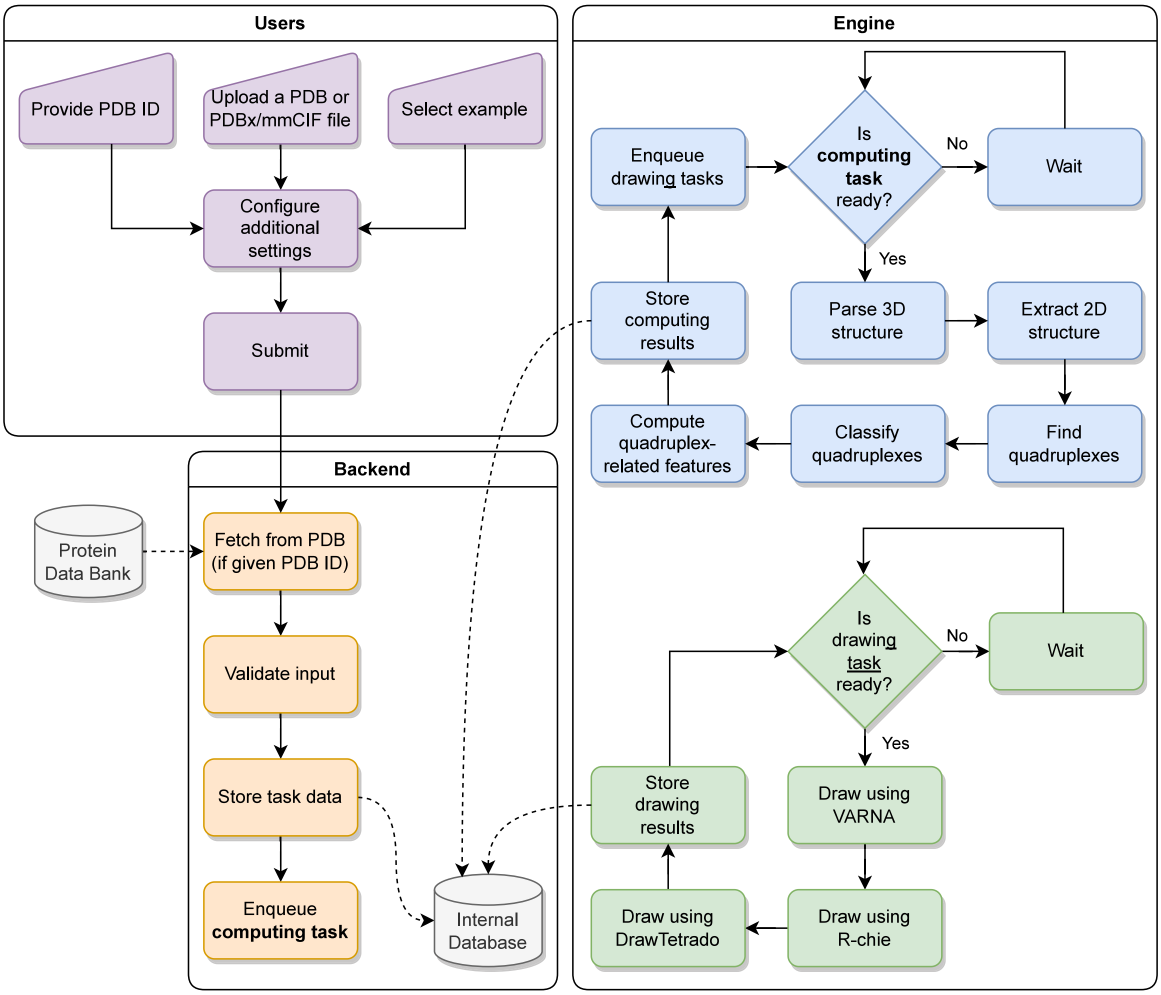
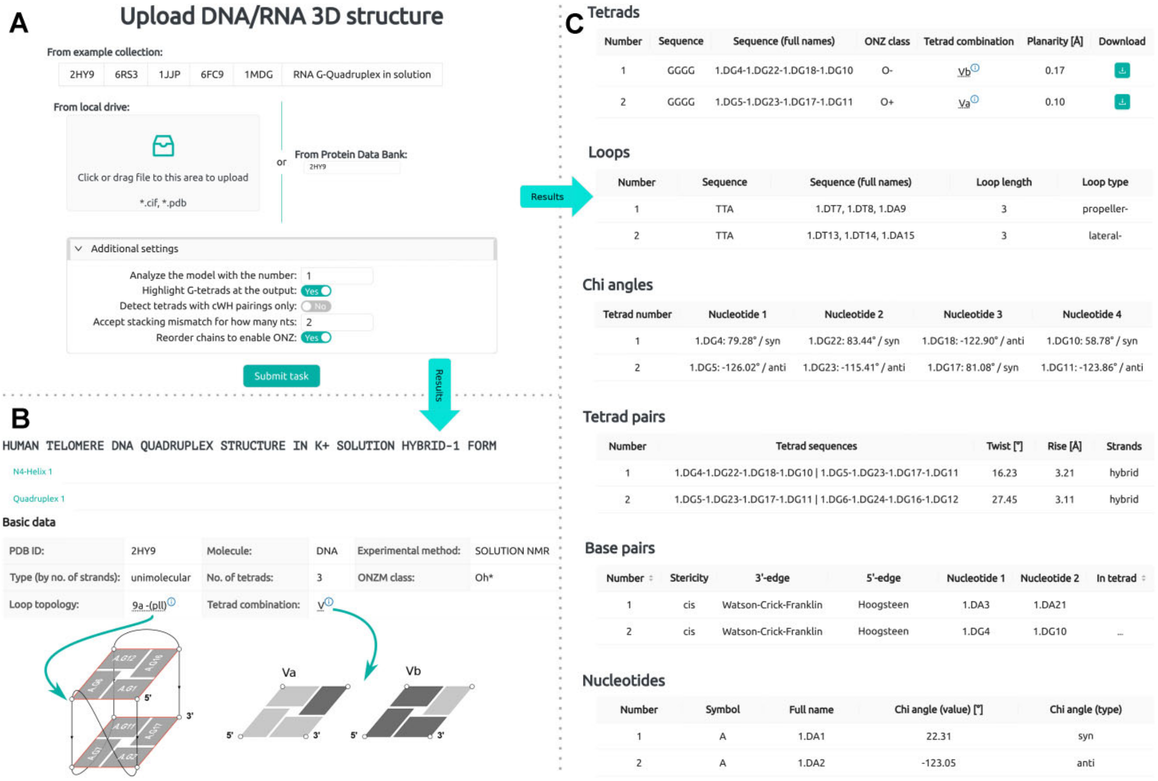


Figure 1. WebTetrado workflow.

Input and output description

The input for WebTetrado is the tertiary structure of the nucleic acid given as the atomic coordinates in a PDB or PDBx/mmCIF file. Users upload the file from a local drive or provide the PDB id of a structure. In the latter case, the back-end automatically downloads the corresponding file from the Protein Data Bank (15). Six ready-to-use examples are also available in the system to familiarize users with the tool's capabilities. Additional settings condition the identification of tetrads and quadruplexes in the input structure and their classification. We provide sensible defaults, but optionally users can modify their values.

Users can select a particular model to analyze in the case of a multi-model input file. Next, they can instruct the system to turn off G-tetrads highlightning, i.e. canonical ones composed of exactly four guanines. By default, WebTetrado does not make assumptions about nucleotide composition and finds all types of quartet, but it highlights the canonical G4s among them. This behavior can be disabled. In addition, the next setting controls whether tetrads are detected with cWH pairings only. Again, these pairings are present in the usual G4 tetrads, but by default WebTetrado generalizes the search for quartets and looks for all kinds of pairs between in-tetrad nucleobases. In addition, users can set how many nucleotides to accept for stacking mismatch. It controls how sensitive WebTetrado should be to inherent uncertainty in stacking interaction detection. In a perfect, canonical quadruplex, each tetrad pair contains four pairs of stacked nucleobases. However, for several reasons, this might not be detected as such. For example, if the structure resolution is low or if it is an intermediate stage taken from the molecular dynamics trajectory, then most likely not all four nucleobase pairs will be recognized as stacked. To alleviate this issue, WebTetrado makes it possible to set a mismatch threshold. By default, at least two pairs of nucleobases stacked between tetrads allow them to be treated as part of the same quadruplex. Finally, users can disable chain reordering, required to classify bi/tetramolecular quadruplexes, which is enabled in default runs. Keeping the



Upload DNA/RNA 3D structure

From example collection:

2HY9 | 6RS3 | 1JJP | 6FC9 | 1MDG | RNA G-Quadruplex in solution

From local drive:

Click or drag file to this area to upload

*.cif, *.pdb

or

From Protein Data Bank: 2HY9

Additional settings

Analyze the model with the number: 1
Highlight G-tetrads at the output: Yes
Detect tetrads with cWH pairings only: No
Accept stacking mismatch for how many nts: 2
Reorder chains to enable ONZ: Yes

Submit task

HUMAN TELOMERE DNA QUADRUPLEX STRUCTURE IN K+ SOLUTION HYBRID-1 FORM

N4-Helix 1

Quadruplex 1

Basic data

PDB ID:	2HY9	Molecule:	DNA	Experimental method:	SOLUTION NMR
Type (by no. of strands):	unimolecular	No. of tetrads:	3	ONZM class:	Oh*
Loop topology:	9a -(pll)	Tetrad combination:	V		

Tetrads

Number	Sequence	Sequence (full names)	ONZ class	Tetrad combination	Planarity [Å]	Download
1	GGGG	1.DG4-1.DG22-1.DG18-1.DG10	O-	Vb	0.17	
2	GGGG	1.DG5-1.DG23-1.DG17-1.DG11	O+	Va	0.10	

Loops

Number	Sequence	Sequence (full names)	Loop length	Loop type
1	TTA	1.DT7, 1.DT8, 1.DA9	3	propeller-
2	TTA	1.DT13, 1.DT14, 1.DA15	3	lateral-

Chi angles

Tetrad number	Nucleotide 1	Nucleotide 2	Nucleotide 3	Nucleotide 4
1	1.DG4: 79.28° / syn	1.DG22: 83.44° / syn	1.DG18: -122.90° / anti	1.DG10: 58.78° / syn
2	1.DG5: -126.02° / anti	1.DG23: -115.41° / anti	1.DG17: 81.08° / syn	1.DG11: -123.86° / anti

Tetrad pairs

Number	Tetrad sequences	Twist [°]	Rise [Å]	Strands
1	1.DG4-1.DG22-1.DG18-1.DG10 | 1.DG5-1.DG23-1.DG17-1.DG11	16.23	3.21	hybrid
2	1.DG5-1.DG23-1.DG17-1.DG11 | 1.DG6-1.DG24-1.DG16-1.DG12	27.45	3.11	hybrid

Base pairs

Number	Stericity	3'-edge	5'-edge	Nucleotide 1	Nucleotide 2	In tetrad
1	cis	Watson-Crick-Franklin	Hoogsteen	1.DA3	1.DA21	
2	cis	Watson-Crick-Franklin	Hoogsteen	1.DG4	1.DG10	...

Nucleotides

Number	Symbol	Full name	Chi angle (value) [°]	Chi angle (type)
1	A	1.DA1	22.31	syn
2	A	1.DA2	-123.05	anti

Figure 2. User interface of WebTetrado: (**A**) submission form, (**B**) result summary, (**C**) tables with detailed data.

original order of chains, as given in the PDB or PDBx/mmCIF file, depends on the input settings.

The result page has a dedicated, bookmarkable URL that allows users to return up to 7 days after completing the task. It displays all gathered quadruplex-related information and visualizations: (i) metadata concerning the structure (PDB id, molecule type, experimental method), (ii) the sequence of the input molecule and its secondary structure in a two-line dot-bracket with colored G-tracts, (iii) quadruplex description (sequence, number of tetrads, type by number of strands, loop description, tetrad combination, rise, twist, type by strand orientation, ONZM class), (iv) tetrad description (sequence, nucleotides, planarity, χ angles, base pairs with Leontis-Westhof classification, ONZ class) and (v) visualizations of the secondary and tertiary structures with ONZ-related coloring (classical, arc and layer diagrams, a cartoon model).

Users can download the results in CSV format for tabular data and SVG or PNG formats for 2D and 3D structure visualizations.

RESULTS AND DISCUSSION

User interface

Figure 2 shows screenshots of the WebTetrado service. Panel 2A shows the screen of the submission form, which allows specifying structure calculations. Submitting a task redirects to a self-refreshing waiting page, allowing users to enable browser notifications. If enabled, the browser will show a message when WebTetrado finishes processing the request.

The remaining panels are the main parts of the result page. The panel 2B shows a table with general information about a quadruplex. Above the table, two tab selectors make it possible to show a different N4 helix or a different quadruplex. Panel 2C shows the content of the result page. It includes several tables with details about tetrads, loops, χ angles, tetrad pairs, base pairs and nucleotides.

Analysis of the major G-quadruplex form of HIV-1 LTR

G-quadruplex-forming sequences are widespread in genomes, including viral ones. Human immunodeficiency virus 1 (HIV-1) has a 5’-LTR (long terminal repeat) promoter, which plays an important role in the viral replication cycle and is regulated by G-quadruplexes (16). In particular, the LTR-III fragment forms the most stable G-quadruplex. In 2018, Butovskaya *et al.* reported the NMR structure of LTR-III in a K+ solution and deposited it in the Protein Data Bank with PDB id 6H1K (17). The reported structure has several unique and difficult-to-identify features, all of which the WebTetrado can find. First, it contains an elongated loop that folds into a stem-loop motif, making the entire structure a quadruplex-duplex combination (see Figures 3A, B and D). Such quadruplex-duplex motifs

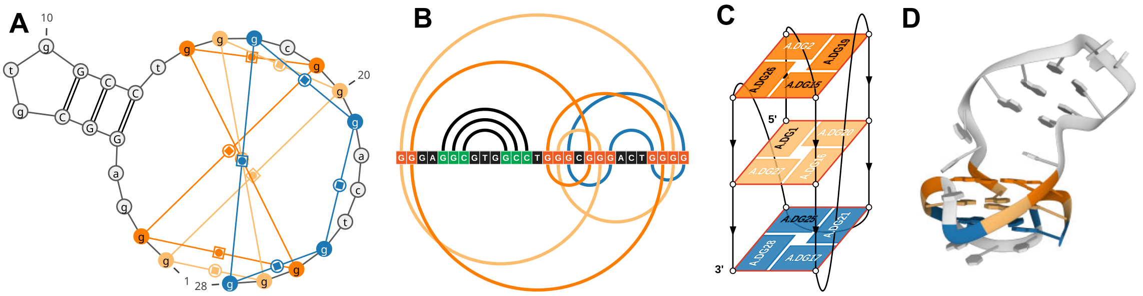


Figure 3. Major G-quadruplex form of HIV-1 LTR (PDB id: 6H1K) visualized in WebTetrado: (**A**) 2D diagram with a visible stem-loop, (**B**) arc diagram with stem–loop as black arcs, (**C**) 2.5D visualization allowing to trace the quadruplex fold, (**D**) 3D image color-coded as the other ones.

have been actively investigated due to their features and potential applications in medicine and biotechnology (18). Furthermore, the HIV-1 LTR-III quadruplex includes a V-shaped loop, which occurs when the 5'-endmost tetrad lies in the middle of the G-quartet stack (see Figure 3C). In addition, it has a hybrid pattern of strand orientations and a combination of 1 nt propeller, 3 nt lateral and 12 nt diagonal loops (see Figure 3C).

The VARNA and R-Chie visualizations are semi-interactive—the user may reconfigure them using switches placed above them in the user interface. These switches change the visibility of base pairs outside the tetrads. In particular, for the HIV-1 LTR-III quadruplex, the visualization of the duplex fragment can be disabled to focus only on the quadruplex part. All four visualizations are color-coded according to the ONZ scheme, which makes it easier to understand the tetrad features in different contexts.

WebTetrado automatically finds all the confirmations of the unique quadruplex topology in the 6H1K PDB structure. In addition, it classifies the tetrads and quadruplex according to Webba da Silva (19) and the ONZ scheme (6). According to it, the HIV-1 LTR-III structure contains two `Z` and one `O` tetrad, making it an `Mh` (mixed hybrid) class quadruplex. The mixed class encompasses the rarest and most complex quadruplex topologies. WebTetrado also computes several quantitative features of the G4 and shows the structural data: nucleobase conformations and base-pairing information both in the tetrads and in the stem–loop motif.

CONCLUSIONS

WebTetrado is a new web server for analyzing structures containing quadruplexes, four-stranded DNA/RNA motifs of high functional significance that fold into complex shapes. It supports automatic identification and advanced analyses of all types of quadruplexes based only on atomic coordinates. WebTetrado provides a wealth of data computed from the given input file, including classification schemes recognized by the G4 community. In addition, it shows visualizations specially designed to represent quadruplexes. The tool is free and open to anyone interested in the analysis of DNA/RNA structures that include quadruplex motifs.

DATA AVAILABILITY

WebTetrado is implemented as a publicly available web server with an intuitive interface and can be freely accessed at https://webtetrado.cs.put.poznan.pl/.

ACKNOWLEDGEMENTS

We thank M. Antczak and J. Sarzynska for testing WebTetrado and V. Hlushchenko for support in adapting Mol*.

FUNDING

National Science Center, Poland [2019/35/B/ST6/03074]; Poznan University of Technology, and Institute of Bioorganic Chemistry, PAS [statutory funds]. Funding for open access charge: National Science Center, Poland [2019/35/B/ST6/03074].

Conflict of interest statement. None declared.

REFERENCES

1. Varshney,D., Spiegel,J., Zyner,K., Tannahill,D. and Balasubramanian,S. (2020) The regulation and functions of DNA and RNA G-quadruplexes. *Nat. Rev. Mol. Cell Biol.*, **21**, 459–474.
2. Plavec,J. (2020) Quadruplex targets in neurodegenerative diseases. In: *Annual Reports in Medicinal Chemistry*. Elsevier, Vol. **54**, pp. 441–483.
3. Neidle,S., (ed.) (2020) In: *Quadruplex Nucleic Acids as Targets for Medicinal Chemistry*. Academic Press.
4. Spiegel,J., Adhikari,S. and Balasubramanian,S. (2020) The structure and function of DNA G-quadruplexes. *Trends Chem.*, **2**, 123–136.
5. Miskiewicz,J., Sarzynska,J. and Szachniuk,M. (2021) How bioinformatics resources work with G4 RNAs. *Brief. Bioinform.*, **22**, bbaa201.
6. Popenda,M., Miskiewicz,J., Sarzynska,J., Zok,T. and Szachniuk,M. (2020) Topology-based classification of tetrads and quadruplex structures. *Bioinformatics*, **36**, 1129–1134.
7. Lu,X.-J., Bussemaker,H.J. and Olson,W.K. (2015) DSSR: an integrated software tool for dissecting the spatial structure of RNA. *Nucleic Acids Res.*, **43**, e142.
8. Zok,T., Popenda,M. and Szachniuk,M. (2020) ElTetrado: a tool for identification and classification of tetrads and quadruplexes. *BMC Bioinformatics*, **21**, 40.
9. Zok,T., Kraszewska,N., Miskiewicz,J., Pielacinska,P., Zurkowski,M. and Szachniuk,M. (2022) ONQUADRO: a database of experimentally determined quadruplex structures. *Nucleic Acids Res.*, **50**, D253–D258.
10. Farag,M., Messaoudi,C. and Mouawad,L. (2023) ASC-G4, an algorithm to calculate advanced structural characteristics of G-quadruplexes. *Nucleic Acids Res.*, **51**, 2087–2107.

11. Zurkowski,M., Zok,T. and Szachniuk,M. (2022) DrawTetrado to create layer diagrams of G4 structures. *Bioinformatics*, **38**, 3835–3836.
12. Darty,K., Denise,A. and Ponty,Y. (2009) VARNA: interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, **25**, 1974–1975.
13. Lai,D., Proctor,J.R., Zhu,J.Y.A. and Meyer,I.M. (2012) R-chie: a web server and R package for visualizing RNA secondary structures. *Nucleic Acids Res.*, **40**, e95.
14. Sehnal,D., Bittrich,S., Deshpande,M., Svobodova,R., Berka,K., Bazgier,V., Velankar,S., Burley,S., Koca,J. and Rose,A. (2021) Mol* viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic Acids Res.*, **49**, W431–W437.
15. Berman,H., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T., Weissig,H., Shindyalov,I. and Bourne,P. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
16. Perrone,R., Nadai,M., Frasson,I., Poe,J.A., Butovskaya,E., Smithgall,T.E., Palumbo,M., Palù,G. and Richter,S.N. (2013) A dynamic G-quadruplex region regulates the HIV-1 long terminal repeat promoter. *J. Med. Chem.*, **56**, 6521–6530.
17. Butovskaya,E., Heddi,B., Bakalar,B., Richter,S.N. and Phan,A.T. (2018) Major G-quadruplex form of HIV-1 LTR reveals a (3 + 1) folding topology containing a stem-loop. *J. Am. Chem. Soc.*, **140**, 13654–13662.
18. Vianney,Y.M. and Weisz,K. (2022) High-affinity binding at quadruplex–duplex junctions: rather the rule than the exception. *Nucleic Acids Res.*, **50**, 11948–11964.
19. Webba da Silva,M. (2007) Geometric formalism for DNA quadruplex folding. *Chem. Eur. J.*, **13**, 9738–9745.

Bioinformatics, 2023, **39(5)**, btad315
https://doi.org/10.1093/bioinformatics/btad315
Advance access publication 11 May 2023
Original Paper

Structural bioinformatics

High-quality, customizable heuristics for RNA 3D structure alignment

Michal Zurkowski[1], **Maciej Antczak**[1,2,*], **Marta Szachniuk**[1,2,*]

[1]Institute of Computing Science, Poznan University of Technology, 60-965 Poznan, Poland
[2]Department of Structural Bioinformatics, Institute of Bioorganic Chemistry, Polish Academy of Sciences, 61-704 Poznan, Poland

*Corresponding author. Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland.
E-mail: mantczak@cs.put.poznan.pl (M.A.), mszachniuk@cs.put.poznan.pl (M.S.)

Associate Editor: Yann Ponty

Abstract

Motivation: Tertiary structure alignment is one of the main challenges in the computer-aided comparative study of molecular structures. Its aim is to optimally overlay the 3D shapes of two or more molecules in space to find the correspondence between their nucleotides. Alignment is the starting point for most algorithms that assess structural similarity or find common substructures. Thus, it has applications in solving a variety of bioinformatics problems, e.g. in the search for structural patterns, structure clustering, identifying structural redundancy, and evaluating the prediction accuracy of 3D models. To date, several tools have been developed to align 3D structures of RNA. However, most of them are not applicable to arbitrarily large structures and do not allow users to parameterize the optimization algorithm.

Results: We present two customizable heuristics for flexible alignment of 3D RNA structures, geometric search (GEOS), and genetic algorithm (GENS). They work in sequence-dependent/independent mode and find the suboptimal alignment of expected quality (below a predefined RMSD threshold). We compare their performance with those of state-of-the-art methods for aligning RNA structures. We show the results of quantitative and qualitative tests run for all of these algorithms on benchmark sets of RNA structures.

Availability and implementation: Source codes for both heuristics are hosted at https://github.com/RNApolis/rnahugs.

1 Introduction

Comparing the 3D structures of RNA molecules is one of the important problems in computational biology and bioinformatics. Comparative analysis of polymer folds is based primarily on structural alignment, and followed by hunting for similarities between data objects given as clouds of atoms. It applies to establish homology between molecules following their 3D conformations (Dietmann and Holm 2001; Blazewicz et al. 2005), discover conserved 3D structure motifs (Miskiewicz et al. 2017; Valdes-Jimenez et al. 2019), identify tertiary structure families and perform structure-based classifications (Lo Conte et al. 2000), assess the quality of algorithms that predict 3D models of molecules (Lukasiak et al. 2015; Gong et al. 2019), create non-redundant sets for benchmarking and structural studies (Leontis and Zirbel 2012; Adamczyk et al. 2022), etc. Structural alignment consists in such an arrangement of one structure vs. the other in a 3D space that the average distance between the corresponding atoms is as small as possible. If the arrangement depends on the sequence, that is, optimizes the distance between the corresponding nucleotides of two molecules having the same sequences, we call it superimposition. Otherwise, when the matching is sequence-independent, we perform structural alignment. Structures can be aligned in a rigid or flexible way. The former involves a rigid transformation (rotations and translations) of an entire structure that is superimposed onto the other. Unfortunately, it has the disadvantage of leaving entire regions without alignment, even if they are similar and could superpose very well locally. To overcome these cons, aligners of the new generation usually apply flexible alignment. It consists of building an alignment through a sequence of local transformations focused on fragments of structures. If alignment aims to find local similarities and solve the problem of maximum common substructures, the result should include the location and length of the matched, implicitly similar fragments and their RMSD, root mean square deviation (Kabsch 1978). The latter measures the quality of alignment (Maiorov and Crippen 1994; Lukasiak et al. 2013). Otherwise, aligners can be also applied to assess a global similarity of the structures. Therefore, when selecting the alignment algorithm, one should take into account the problem to be solved—whether the chains compared have the same length, whether we know their sequences and if they are similar, whether the alignment is dependent or independent on the sequence, and whether we need rigid or flexible alignment. All of this impacts the alignment optimization procedure, including the objective function and computational complexity.

To date, several algorithms have challenged the problem of aligning 3D RNA structures. They include ARTS (Dror et al. 2005), SARA (Capriotti and Marti-Renom 2008), LaJolla (Bauer et al. 2009), R3D Align (Rahrig et al. 2010), iPARTS (Wang et al. 2010), SETTER (Hoksza and Svozil 2012), STAR3D (Ge and Zhang 2015), SupeRNAlign (Piatkowski

Received: October 20, 2022. **Revised:** March 11, 2023. **Editorial Decision:** May 7, 2023. **Accepted:** May 10, 2023

et al. 2017), Rclick (Nguyen et al. 2017), RMalign (Zheng et al. 2019), and RNA-align (Gong et al. 2019). SARA and iPARTS are no longer available. ARTS is the only program in the pool that implements rigid superposition; all the others apply flexible alignment. ARTS depends on the commercial DSRR software (Lu and Olson 2003), which makes it inaccessible to many users. LaJolla generates sparse output (only a PDB file with transformed coordinates of structures, but no indication of which nucleotides were aligned), which limits its usefulness and disables benchmarking. Other algorithms produce results with a wide range of quality. Most perform quite well when aligning structures that show high similarity, and noticeably worse when dealing with homologously distant RNAs. Each follows a different scheme and relies on different initial assumptions; for example, some work in sequence-dependent and independent modes, others only in one of them; a few accept the RMSD threshold at the input, while most do not; some allow processing large datasets as they are available as standalone applications; the other do not, etc. All of them use coarse-grained models to align, but the grain differs between methods. Table 1 selects the important features of these programs.

In this work, we introduce geometric search (GEOS) and genetic algorithm (GENS), two novel algorithms to solve the flexible alignment problem of 3D RNA structures. GEOS is a dedicated geometry-oriented heuristics; GENS applies a genetic search approach. Both algorithms work in two modes, sequence-dependent and sequence-independent. They are complementary in applications. GENS performs better for similar structures and, therefore, is more useful in the alignment of distant homologs or various models of the same structure. GEOS is better for structures that differ significantly, so we recommend it for the problem of finding maximal common substructures. GEOS returns one solution; GENS can find multiple alignments of similar quality if they exist. Both allow users to define the maximum RMSD of the alignment to be found. The effectiveness of GEOS and GENS was confirmed in tests on a set of more than 1000 RNA structures. In this article, we show the results of these computational experiments and compare our algorithms with other available methods that address the problem of aligning RNA tertiary structures.

2 Materials and methods

Let M denote the 3D RNA model to align with the target structure T with an RMSD that does not exceed the threshold value U. Both GEOS and GENS operate on a coarse-grained representation of the tertiary structure of RNA. Therefore, their first step is data preprocessing, which involves the transformation of M and T from a full atom to a 3-bead coarse-grained model. In the latter case, each nucleotide is represented by three pseudoatoms: one for the phosphate group, one for the ribose group, and one for the nitrogenous base. The spatial coordinates of the pseudoatom determine the geometric center in the set of corresponding atoms.

Both methods return the longest alignment found within the given RMSD threshold U. If more alignments of the same length satisfy the threshold, the one with the lowest actual RMSD is returned.

2.1 Geometric search (GEOS)

The Geometric Search algorithm follows a three-step procedure. In the first step, it finds promising alignment kernels; next, it expands kernel-based alignments and compares them to select the best (cf. Supplementary Fig. S1). The best solution is the longest alignment with RMSD $\leq U$. By default $U = 3.5$ Å but users can select a different value between 0 and 20Å and specify it as input parameter. The default value has been chosen based on the experiences of CASP and RNA-Puzzles where structures with RMSD ≤ 3.5 Å are considered similar (Antczak et al. 2016).

Identification of kernels: Kernel K should be made up of three pairs of well-aligned nucleotides, K={N_{TA}–N_{MA}, N_{TB}–N_{MB}, N_{TC}–N_{MC}}; rmsd(K) $\ll U$. In each pair, one nucleotide belongs to the target T and its partner to the model M; N_{TA}, N_{TB}, N_{TC} $\in T$; N_{MA}, N_{MB}, N_{MC} $\in M$. Nucleotides of the kernel that are members of the same structure do not have to be adjacent to each other in the polymer chain. Moreover, a single nucleotide can belong to more than one kernel. Each promising kernel has a high probability of being part of an optimal solution.

A single kernel is searched as follows. Two nucleotides, N_{TA} and N_{TB}, are drawn in the target structure T. Then, any two nucleotides, N_{MA} and N_{MB}, are drawn in model M and optimally aligned with N_{TA} and N_{TB}. The algorithm checks whether rmsd(N_{TA}–N_{MA}, N_{TB}–N_{MB}) $<$ U_2; U_2 is the RMSD threshold defined for two pairs of nucleotides, $U_2 < U$, by default U_2=0.65 Å. If not, it rejects N_{MA} and N_{MB} and continues to draw nucleotides in the model until it finds those that fit the RMSD threshold U_2. GEOS then completes the kernel by adding the third pair. It takes a random nucleotide N_{TC} from the target, $N_{TC} \notin \{N_{TA}, N_{TB}\}$ and a random nucleotide N_{MC} from the model, $N_{MC} \notin \{N_{MA}, N_{MB}\}$, and adds them to the kernel. Next, it checks whether rmsd(N_{TA}–N_{MA}, N_{TB}–N_{MB}, N_{TC}–N_{MC}) $<$ U_3; U_3 is the RMSD threshold for three pairs of nucleotides, $U_2 < U_3 < U$, by default U_3=1.0 Å. If the inequality is not satisfied, GEOS continues to draw the third nucleotide in the model. If it cannot find such a nucleotide in M, it discards the third target nucleotide from the kernel, takes the other random N_{TC} from T, and starts drawing its partner from the model again. Following this scheme, GEOS creates many independent kernels, K_1, K_2, K_3…, which are passed to the second stage. The algorithm then builds structural alignments operating on these kernels and selects the best as a result of the computation.

Building kernel-based alignments: The search for alignment proceeds independently for each kernel found in the previous step. Kernel K_i initiates the creation of a structural alignment L_i between the target T and the model M. The procedure is as follows. The K_i kernel is added to the alignment L_i. The entire structure of the model is transformed (translated and rotated) to align with the target. GEOS performs rigid body alignment using the rotation and translation matrices calculated for the kernel. This means that the transformation of M aims to minimize RMSD only between the nucleotides that make up the kernel. Next, L_i is extended by a new pair of nucleotides N_{TD}–N_{MD}. This is the pair with the smallest Euclidean distance in the set of all non-L_i pairs. The algorithm computes RMSD of current alignment, rmsd(Li). As long as rmsd(Li) $<$ U and there are still non-L_i nucleotides, the algorithm continues to add nucleotide pairs to L_i and recalculates the RMSD of the current solution. GEOS builds multiple independent alignments in parallel, compares them, and selects

Table 1. Selected features of available algorithms to align 3D RNA structures.

	Application type	Input format	RMSD threshold	Processing mode		Coarse-grained model	Alignment based on	Multiple alignments	Alignment-related output data		
				seq-dep	seq-indep				Aligned nts	Actual RMSD	3D alignment
LaJolla	Standalone	PDB			✓	2 angles/3nts	Torsion angles				✓
R3D Align	Webserver, standalone	PDB			✓	4-nt neighborhoods	3D coordinates		✓		✓
SETTER	Webserver	PDB	✓		✓	1 bead (P)	2D units		✓	✓	✓
STAR3D	Standalone	PDB	✓		✓	1 bead (pseudoatom)	2D and 3D geometry		✓	✓	✓
SupeRNAlign	Webserver, standalone	PDB			✓	4-nt neighborhoods	3D coordinates		✓		✓
Rclick	Webserver	PDB, mmCIF	✓		✓	1 bead (C3')	3D coordinates	✓	✓	✓	✓
RMalign	Standalone	PDB			✓	1 bead (C3')	2D and 3D geometry		✓	✓	✓
RNA-align	Webserver, standalone	PDB, mmCIF		✓	✓	1 bead (C3')	2D and 3D geometry		✓	✓	✓
GEOS (seq-dep)	Standalone	PDB, mmCIF	✓	✓	✓	3 beads (pseudoatoms)	3D coordinates		✓	✓	✓
GEOS (seq-indep)	Standalone	PDB, mmCIF	✓	✓	✓	3 beads (pseudoatoms)	3D coordinates		✓	✓	✓
GENS (seq-dep)	Standalone	PDB, mmCIF	✓	✓	✓	3 beads (pseudoatoms)	3D coordinates	✓	✓	✓	✓
GENS (seq-indep)	Standalone	PDB, mmCIF	✓	✓	✓	3 beads (pseudoatoms)	3D coordinates	✓	✓	✓	✓

the best. The best solution is the longest alignment found in all threads. If more alignments have the same length, the one with the lowest RMSD is kept. If multiple solutions have the same length and RMSD score, the first one is returned. It works until it meets one of the stopping criteria (cf. Section 2.3).

2.2 Genetic search (GENS)

Genetic algorithms (GA) are randomized optimization techniques guided by the principles of evolution, with the ability to implicitly parallelize (Booker et al. 1989). They operate in a search space containing chromosomes, that is, potential solutions to a problem encoded in a specific data structure. GA starts by creating a random initial population. Each individual (potential solution) in the population is evaluated using a fitness function. Selected ones go for crossover and mutation and—after application of these operators—yield a new generation. The algorithm then iterates the evaluation, selection, crossover, and mutation until the stop condition is met. The following paragraphs describe the parameters of GENS, the genetic algorithm dedicated to the 3D RNA alignment problem.

Chromosome: An individual is represented as a vector V of length n, where n is the number of nucleotides in the target structure T. If the j-th nucleotide of model M has been aligned with the i-th nucleotide of T, then $V[i] = j$; otherwise $V[i] = 0$.

Initial population: The population consists of c randomly generated individuals, by default $c = 200$. At first, each individual I_k is represented by a vector $V_k = [0, 0, \ldots, 0]$, $k = 1.c$. Next, each vector is randomly filled with the indices of the unassigned nucleotides of M (all nucleotides in the model have the same probability of being selected). The continuity of the chain is preserved; adjacent indexes—except those of unaligned nucleotides—form a unique and monotonic sequence of consecutive numbers; in this sense, the vector $[0, 0, 2, 4, 3, 0]$ represents an inadmissible solution.

Fitness function: When evaluating individuals (i.e. structural alignments), the fitness function takes into account three criteria that describe the quality and length of the alignment. The algorithm seeks to minimize the root mean square deviation, maximize the number of aligned nucleotides, and minimize the number of incorrectly aligned nucleotides.

Selection: Fifteen percent of the fittest individuals in the current population are selected as the most promising seed for a new generation. They are subject to mutations (with probability $P = .74$), crossovers (with $P = .25$), and random seeding (with $P = .01$) to populate the new generation.

Crossover: Crossover is a probabilistic process that generates offspring chromosomes by exchanging information between parents. Two random individuals, I_i and I_j (parents), of the current population are crossed to create a new individual I_k that is added to the population. Crossover involves drawing two numbers $x, y; x < y$ and $x, y \in < 1, n >$. The subvector $I_i[x..y]$ is copied to $I_k[x..y]$; the remaining cells of I_k are filled with values taken from the corresponding cells in I_j. If I_k contains two identical values, one (chosen at random) is converted to 0.

Mutation: Randomly selected individuals are subjected to single-point mutations (with 65% chance), double-point mutations (with 15% chance), triple-point or quadruple-point mutations (both with 10% chance). The following mutation types can be applied (with the same probability): (i) unassign previously assigned nucleotide of T; (ii) assign any available nucleotide of M to the unassigned nucleotide of T; (iii) assign any available nucleotide of M to the already assigned nucleotide of T; and (iv) swap assignments between two randomly selected nucleotides of T. Simple diagrams showing four mutation variants are presented in Supplementary Fig. S2.

2.3 Stopping criteria of GEOS and GENS

Both algorithms stop if one of the following criteria is satisfied: (i) all target residues have been aligned with a given RMSD threshold; (ii) processing time has reached the upper bound c_l (by default $c_l = 300s$); (iii) GENS only: the best current alignment has not been improved or refined for at least c_g generations (by default $c_g = 300$); and (iv) The best current alignment has not been improved or refined for the b_i amount of time.

The size of the time buffer b_i increases by time unit δt if the alignment improves, that is, its length increases or RMSD decreases. δt depends on several parameters. Parameter values can be set in the configuration file.

2.4 Implementation

GEOS and GENS are multi-threaded algorithms. By default, they compute by making use of all available CPU cores. The number of cores involved can be limited by the user, who can set the appropriate parameter value in the configuration file. The performance of both algorithms depends on a number of parameters. Some of them are input parameters; the others can be set via the configuration file. All parameters are described in the readme file available on GitHub (https://github.com/RNApolis/rnahugs). GEOS and GENS are single command-line applications. They are run with input data; the mandatory ones are two files in PDB/mmCIF format with 3D RNA structures to be aligned. Both algorithms were implemented in Java using the Maven package and tested with Java 11 and Maven 3.6.3.

3 Results

We verified the performance of GEOS and GENS and compared them with those of other algorithms that align 3D RNA structures. We conducted several computational experiments to examine various properties of the algorithms. In the first, quantitative (Section 3.1), we ran standalone apps in the sequence-independent mode for a benchmark set from RNA-Puzzles. In the second set of experiments (Section 3.2), we applied all the methods for selected 3D structures in sequence-dependent and sequence-independent modes. In the above experiments, we looked at the length of the alignments, their quality, the variety of solutions, and the ability of the algorithms to process structures of various sizes. Finally, we conducted experiments focusing exclusively on GEOS and GENS. We computed their execution times depending on the instance size and checked the repeatability of the results of both heuristics (Section 3.3).

3.1 Quantitative analysis of alignments

In this multi-model experiment, we used the benchmark set S (https://github.com/RNA-Puzzles/standardized_dataset) available within the RNA-Puzzles resources (Magnus et al. 2020). The collection contains standardized data for 1028 structures and is divided into 22 subsets, $S = \cup_{i=1..22} S_i$. Each of them corresponds to one RNA-Puzzles challenge and includes a reference structure T_i and a set M_i of models generated computationally by various tools, $M_i = \cup_{j=1..k} M_{ij}$, where $k = |S_i| - 1$.

Table 2. Percentage of sequence-independent alignments with RMSD (Å) falling in defined ranges found for RNAs from the RNA-Puzzles set.

	Solutions with RMSD in the range				Unresolved cases
	[0, 3)	[3, 6)	[6, 10)	[10, ∞)	
R3D Align	7.0%	13.5%	24.1%	53.6%	0.8%
STAR3D	17.8%	81.3%	0.2%	—	0.7%
SupeRNAlign	2.9%	8.95%	12.8%	—	74.3%
RMalign	4.9%	85.0%	10.1%	—	—
RNA-align	5.8%	86.4%	7.7%	0.1%	—

The collection contains structures 41–188 nucleotides long. These data were parsed and preprocessed to match the requirements of third-party alignment programs. The changes consisted of renumbering models and atoms in multichain structures.

GEOS and GENS were compared with algorithms implemented as standalone applications that provide detailed output data on alignments, making them comparable. They include R3D Align (Rahrig et al. 2010), STAR3D (Ge and Zhang 2015), SupeRNAlign (Piatkowski et al. 2017), RMalign (Zheng et al. 2019), and RNA-align (Gong et al. 2019). Among them, only STAR3D allows users to define the maximum RMSD of the alignment to be searched for, although the algorithm occasionally returns solutions that exceed the threshold. GEOS and GENS use the RMSD threshold as a hard constraint. Thus, they return solutions of expected quality: fragments that match below the given threshold. In the experiment, we used this option to level the playing field for all the methods tested. We performed a comparative analysis based on the lengths of alignments of the same quality found by different algorithms. The procedure was run separately for each competitive algorithm $A_k \in$ {R3D Align, STAR3D, SupeRNAlign, RMalign, RNA-align} in the following way: (i) the algorithm A_k was run for each model $M_{ij} \in S$ to align it with the corresponding target T_i; (ii) for each M_{ij}, we calculated $rmsd_k(M_{ij}, T_i)$, the RMSD of each alignment found by A_k; (iii) for each $M_{ij} \in S$, GEOS and GENS were run in sequence-independent mode with threshold $U = rmsd_k(M_{ij}, T_i)$; (iv) for each M_{ij}, we compared the lengths of alignments found by the three algorithms (A_k, GEOS and GENS); and (v) for each M_{ij}, we calculated the actual RMSD of the alignment found by GEOS and GENS. The solutions found for the models in the set S had quality in the various ranges; R3D Align: 0.65–59.56 Å, STAR3D: 1.22–8.52 Å; SupeRNAlign: 1.83–8.72 Å; RMAlign: 2.00–9.98 Å, and RNA-align: 2.00–10.92 Å (Table 2). RMalign and RNA-align use exactly the same algorithm for sequence-independent alignment, so they give similar results. For some models, three algorithms failed or aligned single nucleotides (<10% of the model). These cases (8 models for R3D Align, 7 for STAR3D, and 764 for SupeRNAlign) were classified as outliers and discarded from further study. Thus, the analysis was performed for the entire set (1028 structures) when comparing GEOS and GENS with RMalign and RNA-align, and for a subset including 1020/1021/264 models when comparing our algorithms with R3D Align/STAR3D/SupeRNAlign.

With the resulting alignments, we focused on their lengths. First, we computed the percentage coverage of the targets by alignments (Fig. 1). Compared to the others, GEOS and GENS find noticeably longer alignments of the same quality. GEOS emerges as the winner here. Fragments found by this algorithm are, in total, twice as long as the solutions generated by R3D Align and 40% longer than those of SupeRNAlign. Among competing algorithms, RMalign and RNA-align work best. Their alignments are 8–13% shorter than those of GEOS and GENS. Supplementary Fig. S3 allows us to analyze the alignments from the perspective of each subset S_i (each challenge of the RNA-Puzzles) separately. It shows the percentage of each target structure T_i covered by the respective fragments aligned by each algorithm. The coverage was calculated as the average in the set of all models in M_i. In the case of SupeRNAlign, no data are shown for some puzzles because the algorithm did not find solutions there.

For each model $M_i \in S$, we checked which algorithm found the longest alignment. We made a pairwise comparison; the algorithm that aligned a longer fragment than its competitor was credited with winning the duel and the other with a loss. We counted the number of times each algorithm won or lost the duel. The aggregate results showing the performance of GEOS and GENS versus the others are shown in Table 3, details in Supplementary Table S1. GEOS won the most duels (90.55%) and lost the least (4.08%). GENS scored 86.97% of wins and 10.52% of losses. Furthermore, GEOS and GENS have fought each other 4361 times. In these skirmishes, 2679 wins (61.43%) went to GEOS and 480 (11%) to GENS.

3.2 Example alignments of RNA 3D structures

In the second set of experiments, we aligned examples of RNA structures using GEOS, GENS, and competitive algorithms, including those available only through webservers. As the first example, we chose structures from the third challenge of RNA-Puzzles (Cruz et al. 2012). In this puzzle, the predictors targeted the tertiary structure of a glycine riboswitch (PDB id: 3OWZ) (Huang et al. 2010) and submitted 12 *in silico* generated models of this molecule. We took model 2 from Das group (PZ3-Das-2; RMSD = 12.19 Å) to align it with the crystal structure of the target T (84 nts). We ran RMalign—the best according to quantitative analysis (cf. Fig. 1)—to align PZ3-Das-2 with T regardless of the sequence. It aligned 49 nucleotides of the model with an RMSD score 4.59 Å. This value served as a threshold for STAR3D, Rclick, GEOS, and GENS, which were also executed in sequence-independent mode. SETTER uses threshold values, but it turns out that they apply to the alignment of single nucleotides and not entire fragments. For this reason, this program was run with the default settings. Figure 2 shows the alignments obtained with the actual RMSD and the length listed aside.

Next, we tested the algorithms on quadruplexes, specific, highly polymorphic structures found in nucleic acids. We searched the database of experimentally determined conformations of these motifs (Zok et al. 2022) and selected two instances with a similar secondary structure topology (Popenda et al. 2020). Both are hybrids of a duplex and a 2-tetrad unimolecular G-quadruplex. The first, a 26-mer DNA, comes from a complex with human alpha thrombin (PDB id: 6GN7) (Troisi et al. 2018) and entered the algorithms as a target. The second, a 27-mer DNA (PDB id: 2M8Z) (Lim and Phan 2013), was treated as a model; it was the one that would be transformed during alignment. We ran competing programs with default settings. RMalign and RNA-align provided exactly the same solution with actual RMSD = 2.38 Å. We set this value as the threshold for the GEOS and GENS algorithms. Both found alignment that included the same subset of nucleotides. Rclick, which requires a threshold value,

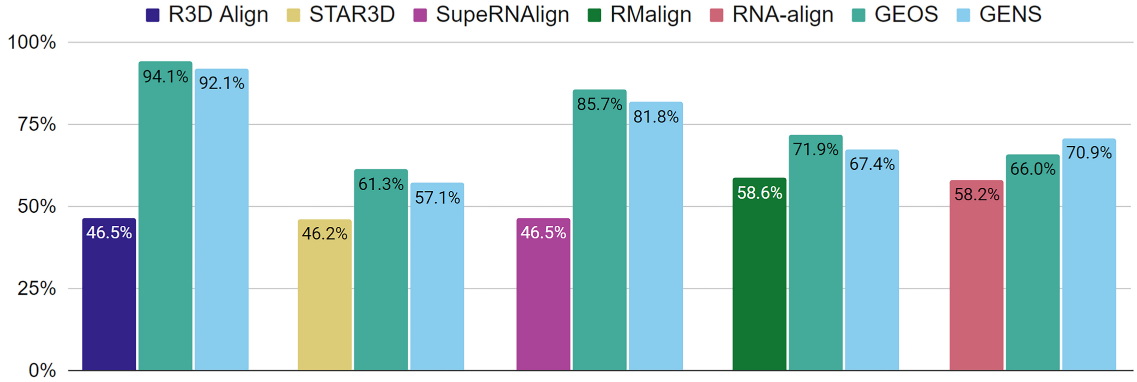


Figure 1. The coverage of target structures by sequence-independent alignments found by R3D Align, STAR3D, SupeRNAlign, RMalign, RNA-align, GEOS, and GENS for 3D RNA structures from the RNA-Puzzles set. The percentage for each algorithm was calculated for all aligned models from the RNA-Puzzles set.

Table 3. GEOS and GENS against other algorithms.

	Duels held	Duels won		Duels lost	
	#	#	%	#	%
R3D Align	2040	6	0.29	2024	99.21
STAR3D	2042	102	4.99	1862	91.21
SupeRNAlign	528	1	0.18	522	98.86
RMalign	2056	259	12.59	1613	78.45
RNA-align	2056	269	13.08	1718	83.56
GEOS	4361	3949	90.55	178	4.08
GENS	4361	3793	86.97	459	10.52

was run with a threshold of 3 Å—the input value accepted by this algorithm must be in the range of 3–6 Å. SupeRNAlign did not return any result. Figure 3 shows all the solutions returned in the experiment.

In the third experiment, we looked at different solutions generated by the GENS algorithm. Again, we took the target from challenge 3 of RNA-Puzzles and one of the models submitted there, namely PZ3-Chen-1. GENS was run in sequence-dependent mode with a threshold equal to 3 Å and found two disjoint solutions. In the first one (Fig. 4A), the aligned fragments of the PZ3-Chen-1 model have a total length of 37 nucleotides, representing 44% of the size of the whole molecule. They are located at the 3′ and 5′ ends of the chain. The actual RMSD of this alignment scores 2.96 Å. The alternative solution (Fig. 3B), located in the middle of the chain, is shorter, with 22 nucleotides (26% of the 84 nt-long structure) aligned with the corresponding fragment of the target. The actual RMSD of this solution is 2.62 Å. All fragments identified in these alignments show a high similarity to the target. However, we would not catch them easily when applying a global alignment approach. That is why GENS is helpful here—the greatest advantage of this algorithm is its ability to find all substructures aligning at a given threshold of the distance measure threshold, that is, all similar fragments in the compared structures.

3.3 Execution time and repeatability of the results

In these experiments, we focused exclusively on GEOS and GENS. First, we checked the repeatability of their results. Both algorithms are heuristics. Thus, they find suboptimal solutions that can vary between different runs for the same input data. In the experiment, we selected the structures of three targets from the benchmark set (targets in Puzzle 1, 3, and 4) and three models predicted per each targeted sequence (PZ1-Bujnicki-4, PZ1-Das-4, PZ1-Santalucia-4, PZ3-Chen-1, PZ3-Das-1, PZ3-Dokholyan-2, PZ4-Adamiak-4, PZ4-Bujnicki-1, PZ4-Mikolajczak-1). We searched for alignment for each model-target pair with the default RMSD threshold (3.5 Å). Each heuristic participated in 18 experiments—9 sequence-dependent alignments and 9 sequence-independent ones. Each experiment was repeated 125 times. We compared the results from all 125 runs for a given instance to compute the minimum and maximum lengths of alignment, the average, and the standard deviation. These results are presented in Supplementary Table S2. We found that GEOS was usually repeatable for the particular pair of 3D structures and the given values of the configuration parameters. Only in one case out of 18 did it find different alignments, with a small difference of 1 nucleotide. GENS, by design, generates multiple alternative alignments and can align various fragments of the structures. The experiment proved this property. In 14 experiments, GENS found various alignments. They varied in length by 2–38% (see also Supplementary Fig. S4).

Separately, we analyzed the execution times of GEOS and GENS as a function of the instance size. First, we computed the times for all instances in the set of RNA-Puzzles containing 22 target structures and 1006 RNA 3D models predicted *in silico*. Supplementary Fig. S5 estimates the trendline determined based on processing the RNA-Puzzles dataset. As the sizes of structures in this collection do not exceed 200 nts, we performed an additional experiment to test the efficiency of GEOS and GENS for larger RNAs. To collect the data for this experiment, we searched RNAsolo (Adamczyk et al. 2022) and BGSU RNA Hub (Leontis and Zirbel 2012), and we selected 8 relevant equivalence classes. Each of them contained homologous structures (with sizes between 100 and 700 nts), one of them being a representative of the class (see Supplementary Table S3). The algorithms looked for an alignment between the representative and other members of the same class. The results of this experiment are presented in Supplementary Fig. S6. For structures up to 200 nts, GEOS finishes computation before 60 s and GENS executes within a maximum of 35 s. For larger structures (>500 nts), GEOS still performs very well and finishes the computation before reaching the stop criterion. GENS is computationally too expensive for large structures and we do not recommend it for molecules above 300 nucleotides.

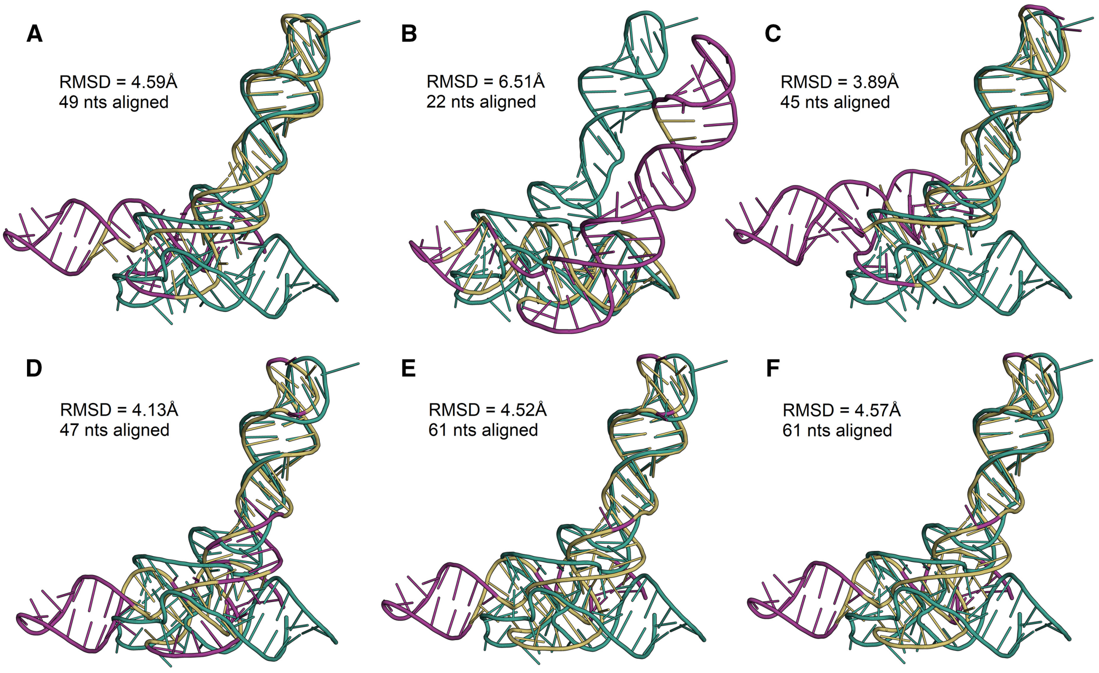


Figure 2. Sequence-independent alignment of the PZ3-Das-2 model with Puzzle 3 target found by (A) RMalign, (B) SETTER, (C) STAR3D, (D) Rclick, (E) GEOS, and (F) GENS. RMSD of the RMalign's solution (4.59 Å) was the threshold for the other algorithms. The target structure is colored green, the aligned fragment of the model is yellow, and the non-aligned one—magenta.

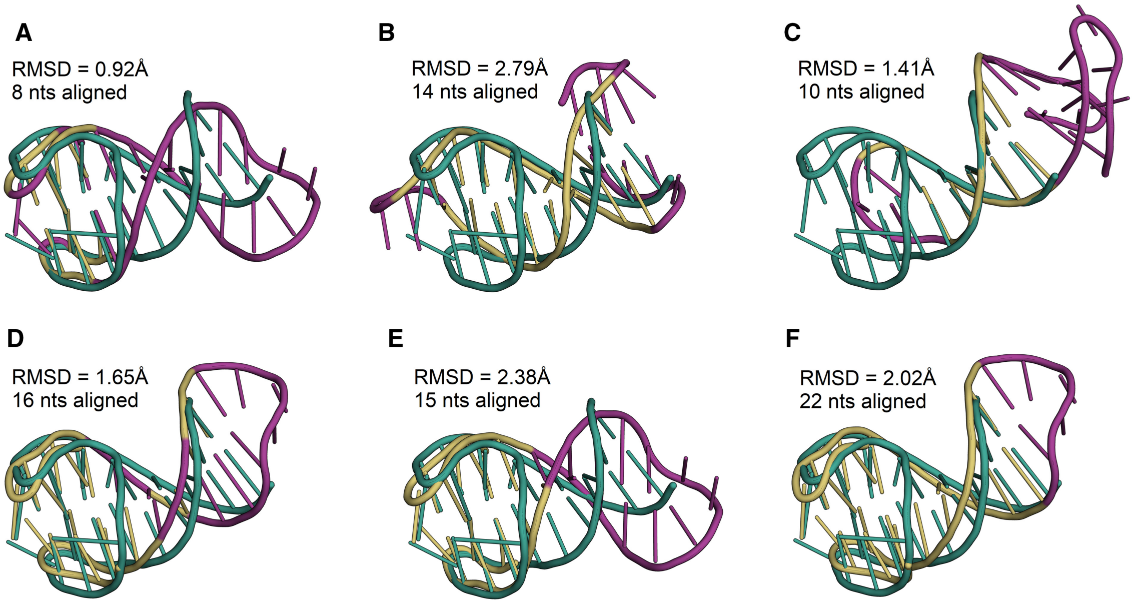
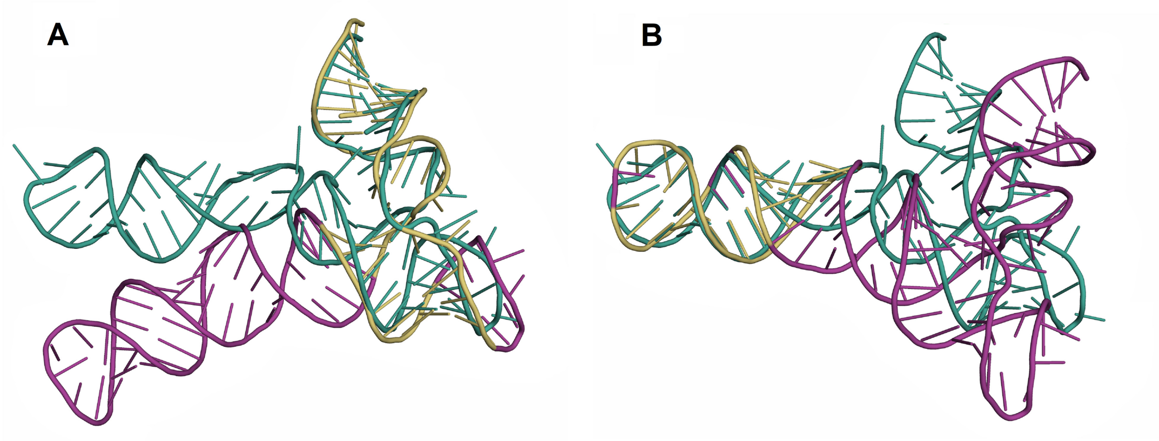


Figure 3. Two structures containing quadruplexes (PDB IDs: 6GN7, 2M8Z) aligned by (A) R3D Align, (B) SETTER, (C) STAR3D, (D) Rclick, (E) RMalign/RNA-align, and (F) GENS/GEOS. The 6GN7 structure is colored green, aligned fragments of 2M8Z are yellow, and non-aligned are magenta.

4 Conclusion

In this article, we address the flexible alignment of 3D RNA structures. The problem is computationally hard, as demonstrated for its protein version (Li 2013). This means that no exact algorithm can find its optimal solution in polynomial time. To date, several computational methods have been developed to solve this problem, SARA (Capriotti and Marti-Renom 2008), LaJolla (Bauer et al. 2009), R3D Align (Rahrig et al. 2010), iPARTS (Wang et al. 2010), SETTER (Hoksza and Svozil 2012), STAR3D (Ge and Zhang 2015), Rclick (Nguyen et al. 2017), SupeRNAlign (Piatkowski et al. 2017), RMalign (Zheng et al. 2019), and RNA-align (Gong et al. 2019). Most of them are still available.

We introduced two heuristics, that applied concurrency processing, to flexibly align 3D RNA structures, geometric search (GEOS), and genetic search (GENS). We compared



```
TARGET:   CUCUGGAGAGAACCGUUUAAUCGGUCGCCGAAGGAGCAAGCUCUGCGGAAACGCAGAGUGAAACUCUCAGGCAAAAGGACAGAG
          |||||||||||||||       |||||||                                        |||||||||||||||
MODEL:    CUCUGGAGAGAACCG-------GGUCGCC----------------------------------------GGCAAAAGGACAGAG
```

```
TARGET:   CUCUGGAGAGAACCGUUUAAUCGGUCGCCGAAGGAGCAAGCUCUGCGGAAACGCAGAGUGAAACUCUCAGGCAAAAGGACAGAG
                                              ||||  ||||| |||||||||||||
MODEL:    ------------------------------------AAGC--UGCGC-UAUGCAGAGUGAA-----------------------
```

Figure 4. Different alignments between one of the predicted models and the target of Puzzle 3 found by GENS in sequence-dependent mode with threshold = 3.0 Å.

them with existing methods that addressed the same problem. To ensure fairness of the comparison, we applied the actual RMSDs of the solutions obtained from competitive methods to constrain our algorithms and ranked all alignments according to their lengths. High-throughput tests on the RNA-Puzzles benchmark set showed that GEOS and GENS outperformed other methods on this criterion.

GEOS and GENS are available in the GitHub repository of the RNApolis group (Szachniuk 2019), ready to be used in future experiments and incorporated as components in various bioinformatics systems. Their uniqueness results from combining features dispersed among other methods, the most important of them being two modes of operation and a user-defined RMSD threshold. Provided within a standalone application, they facilitate finding alignments in multi-model datasets. However, aware of the demand for user-friendly bioinformatics tools, we plan to prepare a web server with GEOS and GENS working in the backend layer. Among other things, it will provide support for additional input parameters, visualization of results, and automatic processing of DNA structures.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

We acknowledge the support of Poznan University of Technology (statutory funds), Institute of Bioorganic Chemistry PAS, and the National Science Centre, Poland [2019/35/B/ST6/03074].

References

Adamczyk B, Antczak M, Szachniuk M. RNAsolo: a repository of cleaned PDB-derived RNA 3D structures. *Bioinformatics* 2022;**38**:3668–70.

Antczak M, Kasprzak M, Lukasiak P *et al.* Structural alignment of protein descriptors—a combinatorial model. *BMC Bioinformatics* 2016;**17**:383.

Bauer R, Rother K, Moor P *et al.* Fast structural alignment of biomolecules using a hash table, n-Grams and string descriptors. *Algorithms* 2009;**2**:692–709.

Blazewicz J, Szachniuk M, Wojtowicz A. RNA tertiary structure determination: NOE pathway construction by tabu search. *Bioinformatics* 2005;**21**:2356–61.

Booker L, Goldberg D, Holland J. Classifier systems and genetic algorithms. *Artif Intell* 1989;**40**:235–82.

Capriotti E, Marti-Renom M. RNA structure alignment by a unit-vector approach. *Bioinformatics* 2008;**24**:i112–8.

Cruz J, Blanchet M-F, Boniecki M *et al.* RNA-Puzzles: a CASP-like evaluation of RNA three-dimensional structure prediction. *RNA* 2012; **18**:610–25.

Dietmann S, Holm L. Identification of homology in protein structure classification. *Nat Struct Biol* 2001;**8**:953–7.

Dror O, Nussinov R, Wolfson H. ARTS: alignment of RNA tertiary structures. *Bioinformatics* 2005;**21**:ii47–53.

Ge P, Zhang S. STAR3D: a stack-based RNA 3D structural alignment tool. *Nucleic Acids Res* 2015;**43**:e137.

Gong S, Zhang C, Zhang Y. RNA-align: quick and accurate alignment of RNA 3D structures based on size-independent TM-scoreRNA. *Bioinformatics* 2019;**35**:4459–61.

Hoksza D, Svozil D. Efficient RNA pairwise structure comparison by SETTER method. *Bioinformatics* 2012;**28**:1858–64.

Huang L, Serganov A, Patel D. Structural insights into ligand recognition by a sensing domain of the cooperative glycine riboswitch. *Mol Cell* 2010;**40**:774–86.

Kabsch W. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Cryst A* 1978;**34**:827–8.

Leontis N, Zirbel C. Nonredundant 3D structure datasets for RNA knowledge extraction and benchmarking. In: Leontis, N., Westhof, E. (eds) *RNA 3D Structure Analysis and Prediction*. vol **27**. Berlin, Heidelberg: Springer, 2012, 281–98.

Li S. The difficulty of protein structure alignment under the RMSD. *Algorithms Mol Biol* 2013;**8**:1.

Lim K, Phan A. Structural basis of DNA quadruplex-duplex junction formation. *Angew Chem Int Ed Engl* 2013;**52**:8566–9.

Lo Conte L, Ailey B, Hubbard T *et al.* SCOP: a structural classification of proteins database. *Nucleic Acids Res* 2000;**28**:257–9.

Lu X, Olson W. 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic Acids Res* 2003;**31**:5108–21.

Lukasiak P, Antczak M, Ratajczak T *et al.* RNAlyzer—novel approach for quality analysis of RNA structural models. *Nucleic Acids Res* 2013;**41**:5978–90.

Lukasiak P, Antczak M, Ratajczak T *et al.* RNAssess—a webserver for quality assessment of RNA 3D structures. *Nucleic Acids Res* 2015; **43**:W502–6.

Magnus M, Antczak M, Zok T *et al.* RNA-Puzzles toolkit: a computational resource of RNA 3D structure benchmark datasets, structure manipulation, and evaluation tools. *Nucleic Acids Res* 2020;**48**: 576–88.

Maiorov V, Crippen G. Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins. *J Mol Biol* 1994;**235**:625–34.

Miskiewicz J, Tomczyk K, Mickiewicz A *et al.* Bioinformatics study of structural patterns in plant microRNA precursors. *Biomed Res Int* 2017;**2017**:6783010.

Nguyen M, Sim A, Wan Y *et al.* Topology independent comparison of RNA 3D structures using the CLICK algorithm. *Nucleic Acids Res* 2017;**45**:e5.

Piatkowski P, Jablonska J, Zyla A *et al.* SupeRNAlign: a new tool for flexible superposition of homologous RNA structures and inference of accurate structure-based sequence alignments. *Nucleic Acids Res* 2017;**45**:e150.

Popenda M, Miskiewicz J, Sarzynska J *et al.* Topology-based classification of tetrads and quadruplex structures. *Bioinformatics* 2020;**36**:1129–34.

Rahrig RR, Leontis NB, Zirbel CL. R3D align: global pairwise alignment of RNA 3D structures using local superpositions. *Bioinformatics* 2010;**26**:2689–97.

Szachniuk M. RNApolis: computational platform for RNA structure analysis. *FCDS* 2019;**44**:241–57.

Troisi R, Napolitano V, Spiridonova V *et al.* Several structural motifs cooperate in determining the highly effective anti-thrombin activity of NU172 aptamer. *Nucleic Acids Res* 2018;**46**:12177–85.

Valdes-Jimenez A, Larriba-Pey J, Nunez-Vivanco G *et al.* 3D-PP: a tool for discovering conserved Three-Dimensional protein patterns. *IJMS* 2019;**20**:3174.

Wang C, Chen K, Lu C. iPARTS: an improved tool of pairwise alignment of RNA tertiary structures. *Nucleic Acids Res* 2010;**38**: W340–7.

Zheng J, Xie J, Hong X *et al.* RMalign: an RNA structural alignment tool based on a novel scoring function RMscore. *BMC Genomics* 2019;**20**:276.

Zok T, Kraszewska N, Miskiewicz J *et al.* ONQUADRO: a database of experimentally determined quadruplex structures. *Nucleic Acids Res* 2022;**50**:D253–8.

Supplementary Material

High-quality, customizable heuristics for RNA 3D structure alignment

Michal Zurkowski[1], Maciej Antczak[1,*], Marta Szachniuk[1,2,*]

[1] Institute of Computing Science, Poznan University of Technology, 60-965 Poznan, Poland

[2] Institute of Bioorganic Chemistry, Polish Academy of Sciences, 61-704 Poznan, Poland

*To whom correspondence should be addressed:

mantczak@cs.put.poznan.pl, mszachniuk@cs.put.poznan.pl

Figure S1. Flowchart of GEOmetric Search algorithm (GEOS): (A) identification of a kernel, (B) building the kernel-based alignment.

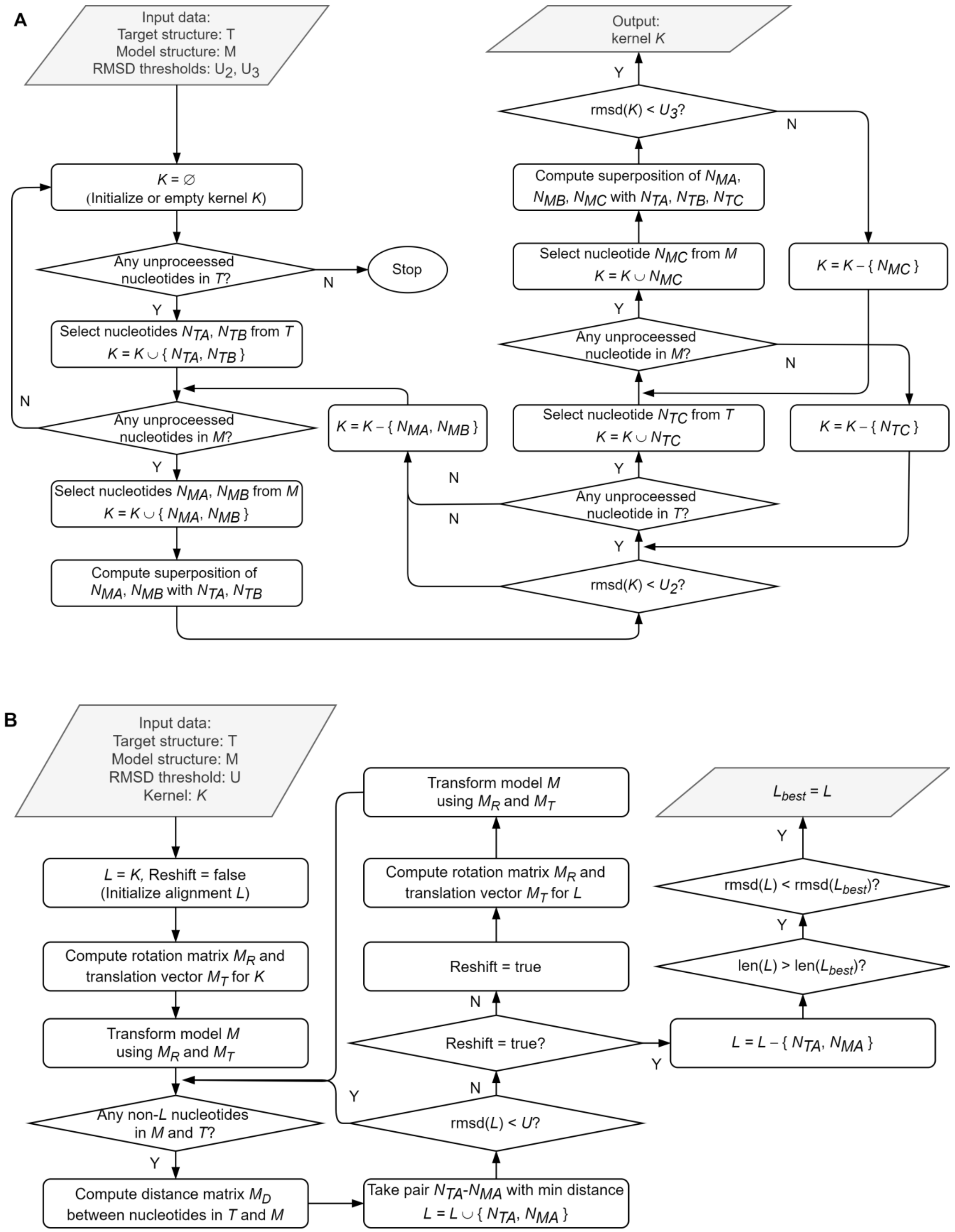

A
Input data:
Target structure: T
Model structure: M
RMSD thresholds: U_2, U_3
$K = \varnothing$
(Initialize or empty kernel K)
Any unproceessed nucleotides in T?
N
Stop
Y
Select nucleotides N_{TA}, N_{TB} from T
$K = K \cup \{ N_{TA}, N_{TB} \}$
N
Any unproceessed nucleotides in M?
Y
Select nucleotides N_{MA}, N_{MB} from M
$K = K \cup \{ N_{MA}, N_{MB} \}$
Compute superposition of N_{MA}, N_{MB} with N_{TA}, N_{TB}
$K = K - \{ N_{MA}, N_{MB} \}$
N
N
rmsd(K) < U_2?
Y
Any unproceessed nucleotide in T?
Y
Select nucleotide N_{TC} from T
$K = K \cup N_{TC}$
Any unproceessed nucleotide in M?
N
Y
Select nucleotide N_{MC} from M
$K = K \cup N_{MC}$
Compute superposition of N_{MA}, N_{MB}, N_{MC} with N_{TA}, N_{TB}, N_{TC}
rmsd(K) < U_3?
N
Y
Output:
kernel K
$K = K - \{ N_{MC} \}$
$K = K - \{ N_{TC} \}$
B
Input data:
Target structure: T
Model structure: M
RMSD threshold: U
Kernel: K
$L = K$, Reshift = false
(Initialize alignment L)
Compute rotation matrix M_R and translation vector M_T for K
Transform model M using M_R and M_T
Any non-L nucleotides in M and T?
Y
Compute distance matrix M_D between nucleotides in T and M
Take pair N_{TA}-N_{MA} with min distance
$L = L \cup \{ N_{TA}, N_{MA} \}$
rmsd(L) < U?
Y
N
Reshift = true?
N
Reshift = true
Compute rotation matrix M_R and translation vector M_T for L
Transform model M using M_R and M_T
Y
$L = L - \{ N_{TA}, N_{MA} \}$
len(L) > len(L_{best})?
Y
rmsd(L) < rmsd(L_{best})?
Y
$L_{best} = L$

Figure S2. Schematic representation of mutations in the GENS algorithm.

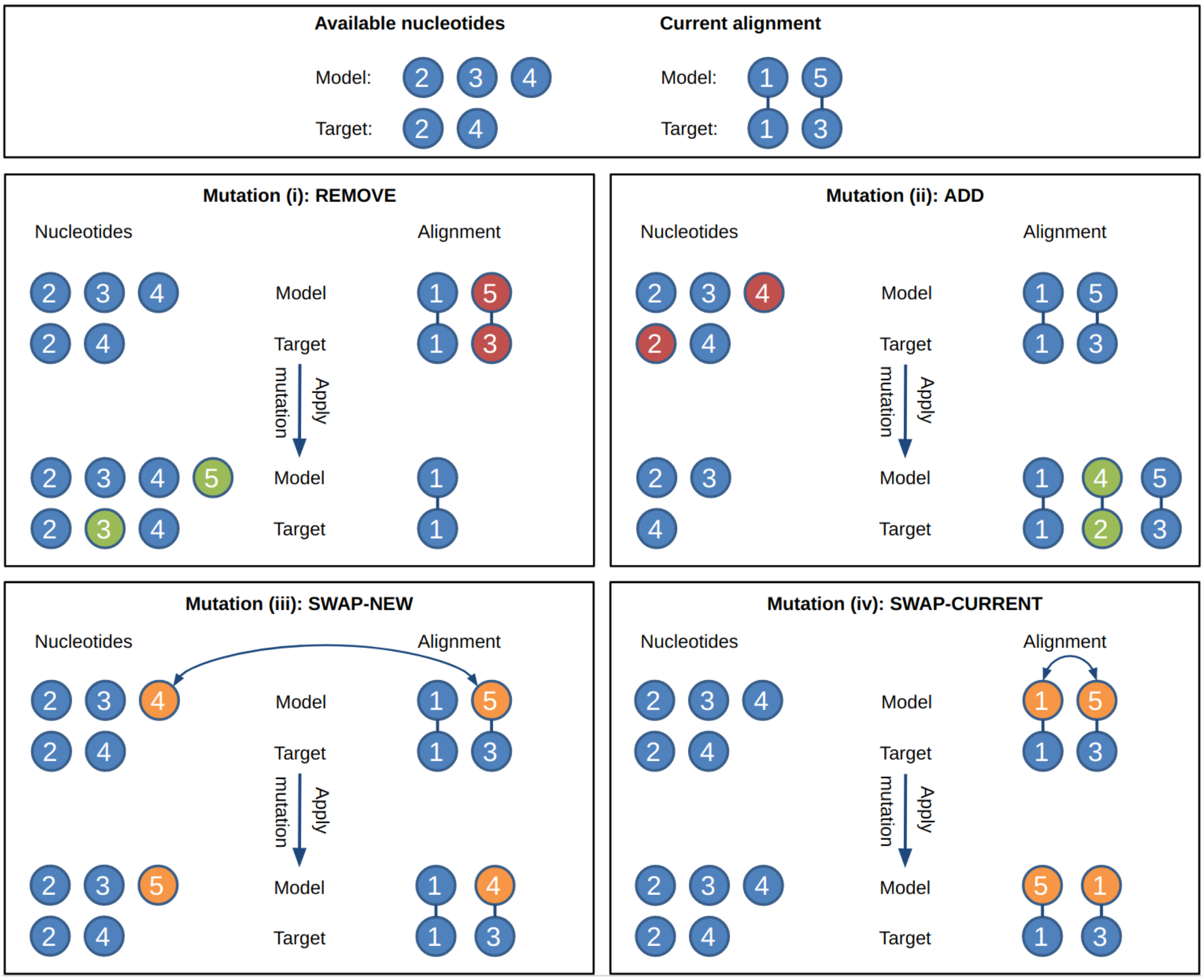
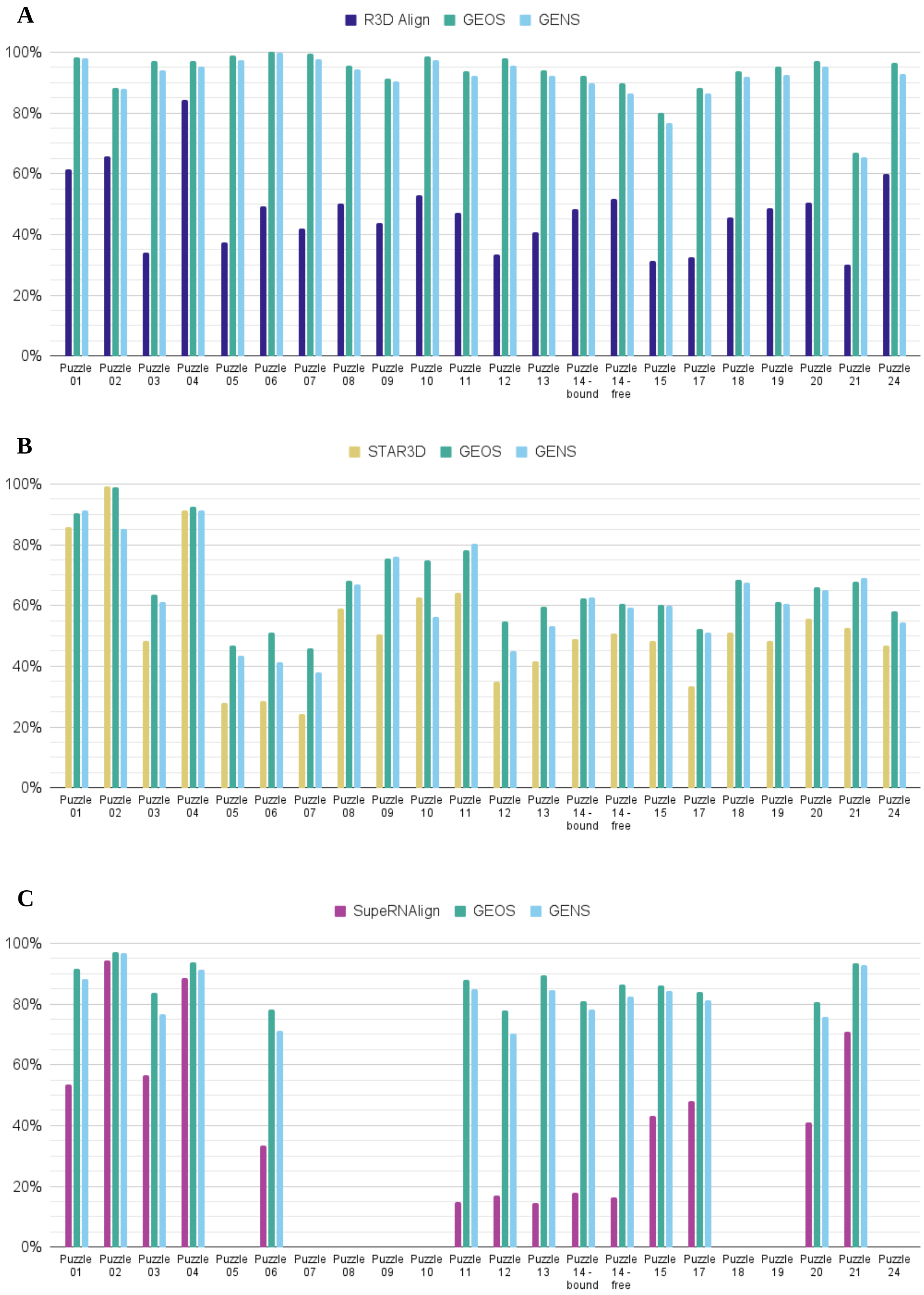


Figure S3. Average coverage of the target structure by aligned fragments of models submitted in a given puzzle. Alignments found by (A) R3D Align, (B) STAR3D, (C) SupeRNAlign, (D) RMalign, and (E) RNA-align.



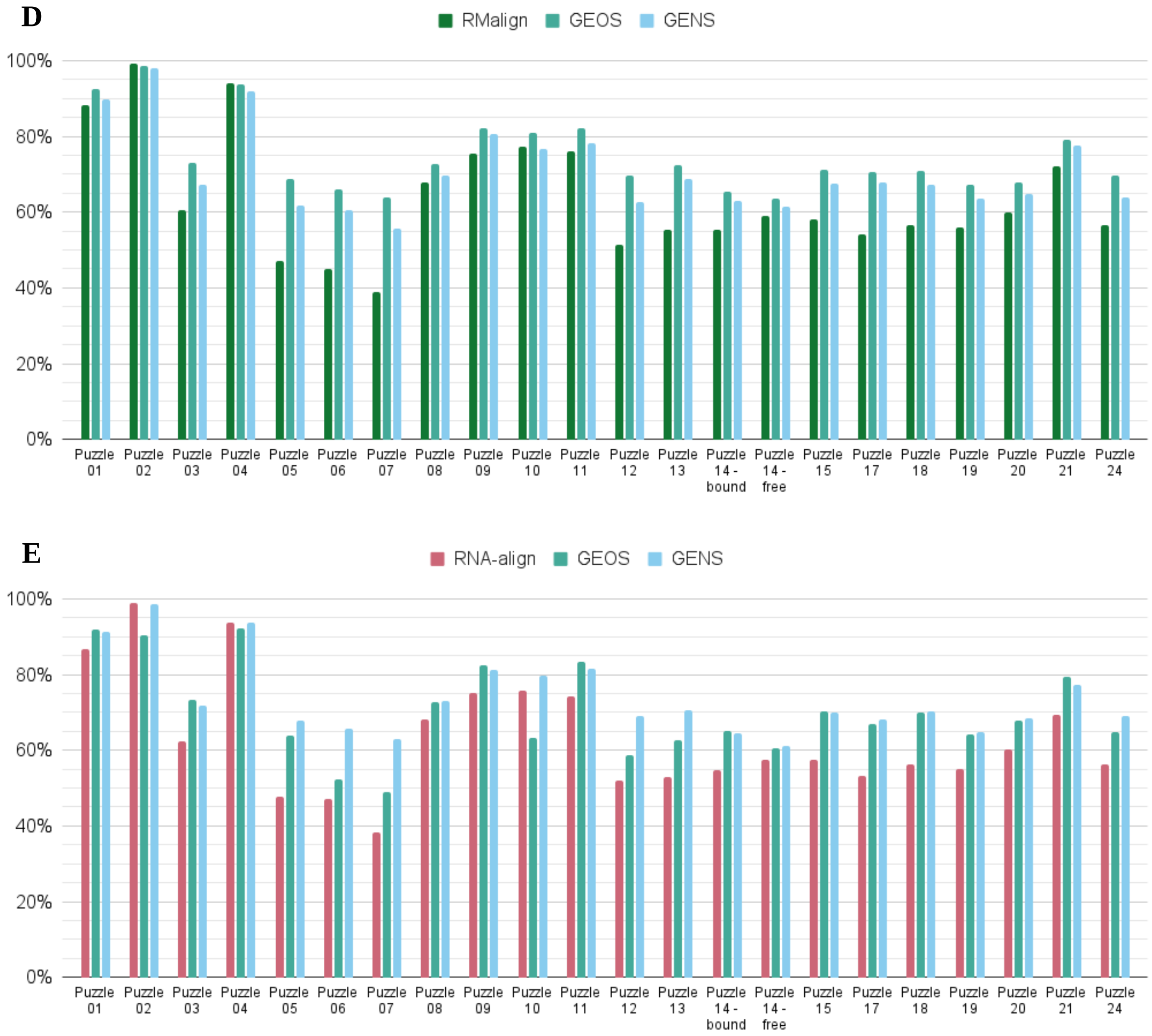

D
RMalign
GEOS
GENS
100%
80%
60%
40%
20%
0%
Puzzle 01
Puzzle 02
Puzzle 03
Puzzle 04
Puzzle 05
Puzzle 06
Puzzle 07
Puzzle 08
Puzzle 09
Puzzle 10
Puzzle 11
Puzzle 12
Puzzle 13
Puzzle 14 - bound
Puzzle 14 - free
Puzzle 15
Puzzle 17
Puzzle 18
Puzzle 19
Puzzle 20
Puzzle 21
Puzzle 24
E
RNA-align
GEOS
GENS
100%
80%
60%
40%
20%
0%
Puzzle 01
Puzzle 02
Puzzle 03
Puzzle 04
Puzzle 05
Puzzle 06
Puzzle 07
Puzzle 08
Puzzle 09
Puzzle 10
Puzzle 11
Puzzle 12
Puzzle 13
Puzzle 14 - bound
Puzzle 14 - free
Puzzle 15
Puzzle 17
Puzzle 18
Puzzle 19
Puzzle 20
Puzzle 21
Puzzle 24

Figure S4. Distribution of the lengths of alignments found by GEOS in the experiment, in which the target structure of Puzzle 01, Puzzle 03, and Puzzle 04 were aligned with three models predicted in these puzzles. GENS was run with the default RMSD threshold (3.5Å) and performed 125 times for each instance.

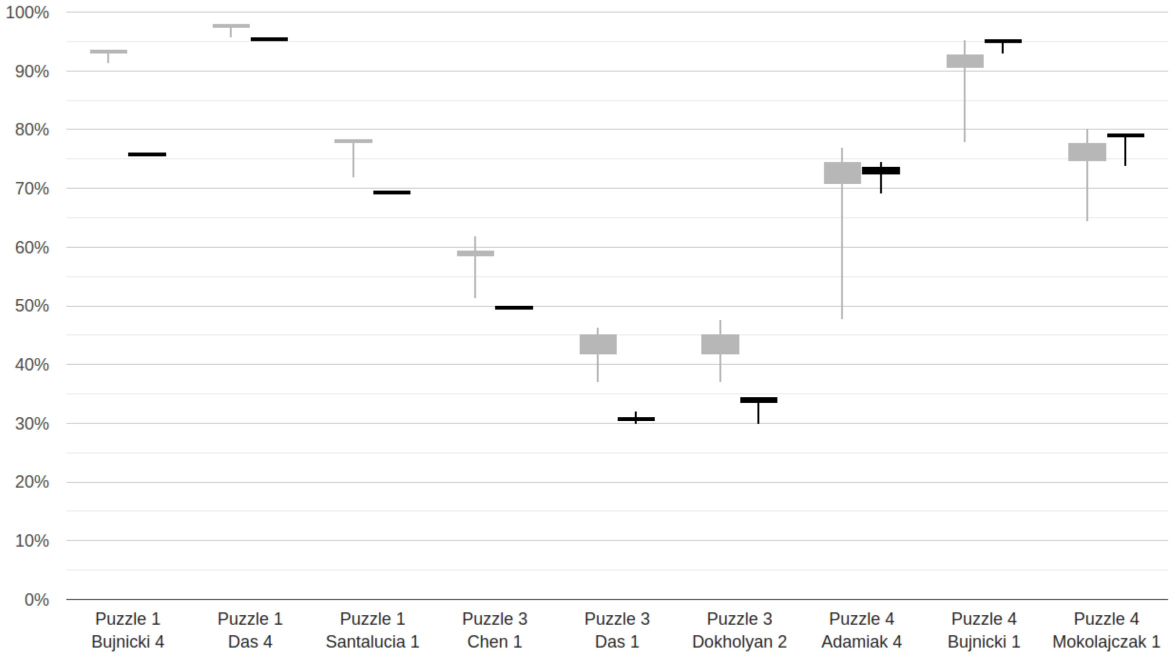
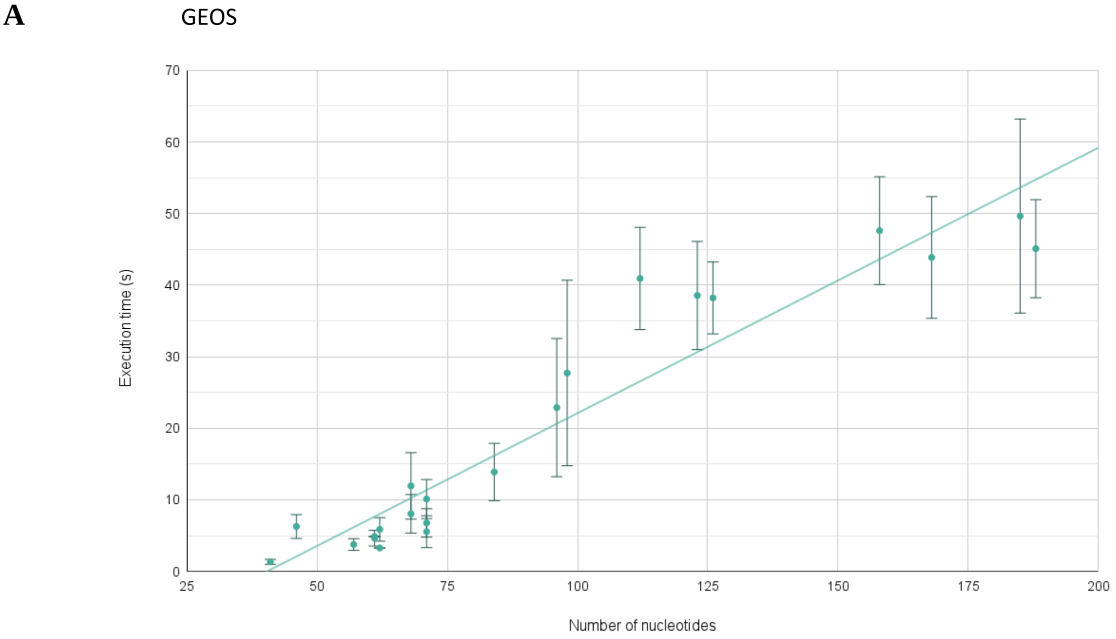


Figure S5. Trendline for execution time of GEOS (A) and GENS (B) computed when aligning small and medium-size structures from the RNA-Puzzles dataset.



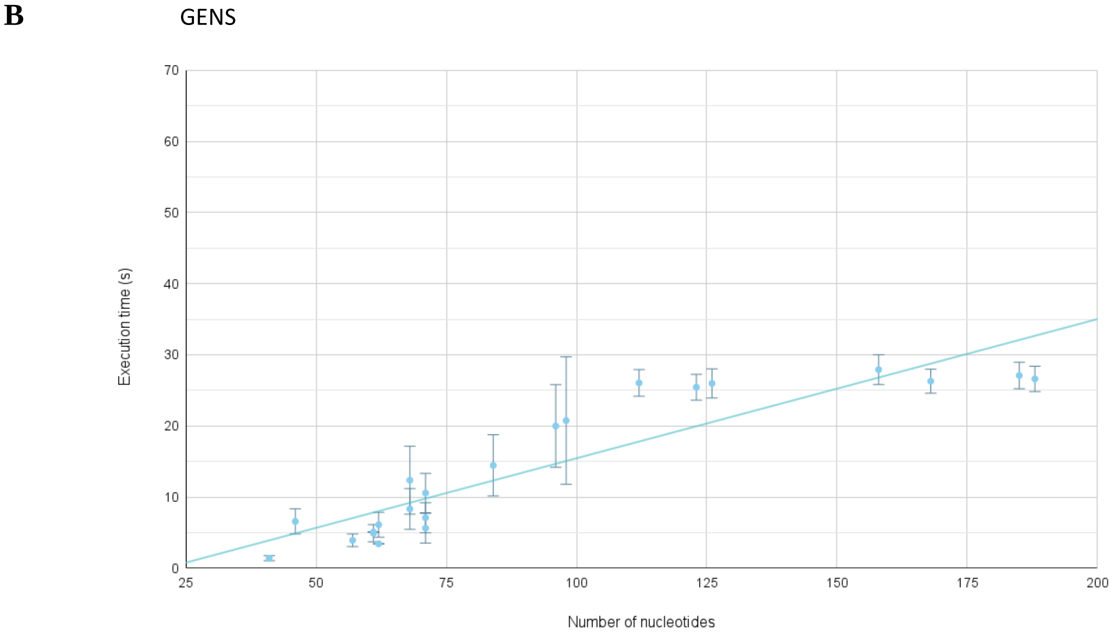


Figure S6. Trendline for execution time of GEOS (A) and GENS (B) computed when aligning large structures from the RNAsolo/BGSU dataset.

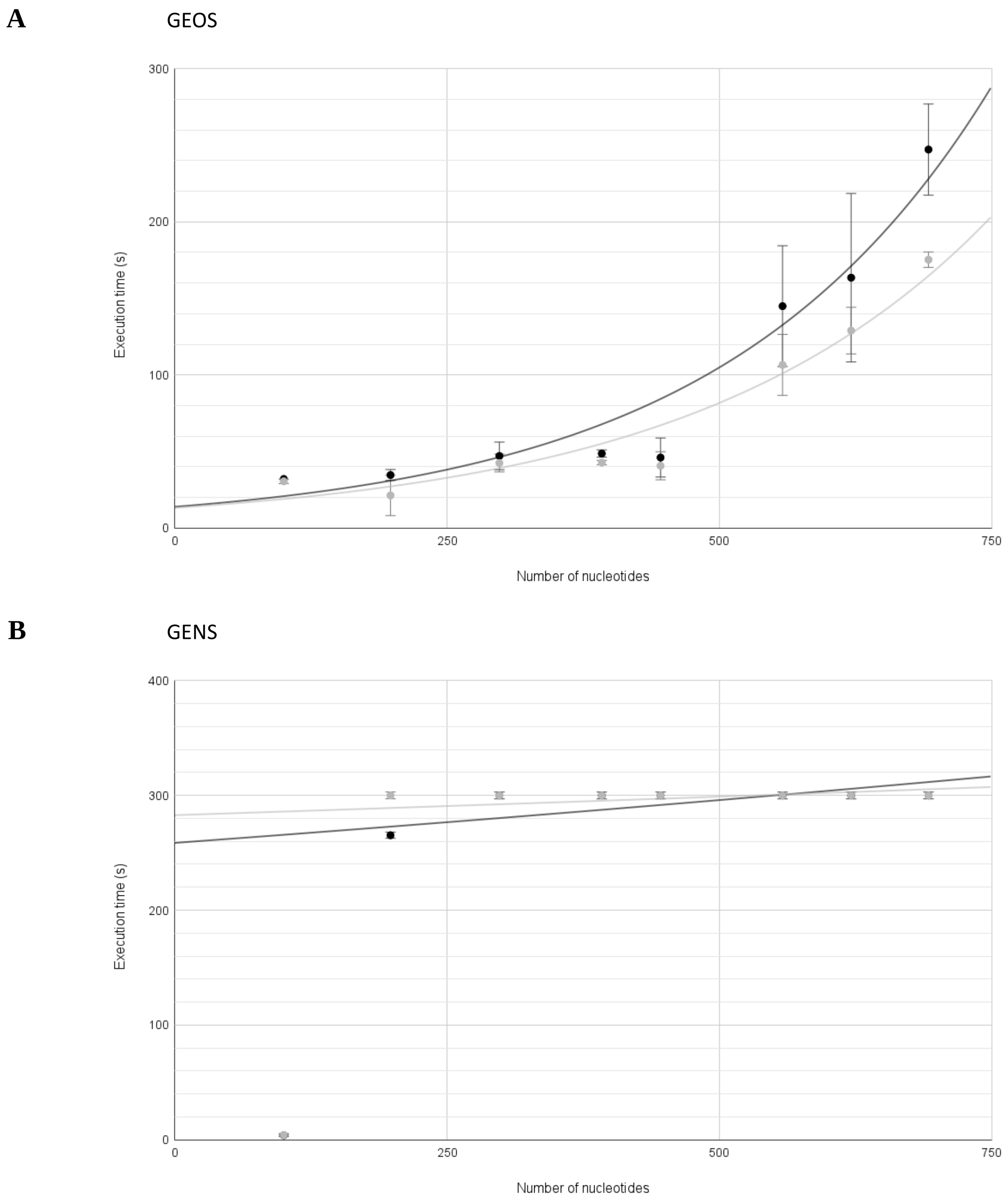


Table S1. The number of duels won and lost by each algorithm.

	GEOS	GENS	R3D Align	# Duels won
GEOS	–	580	1014	1594
GENS	16	–	1010	1026
R3D Align	2	4	–	6
# Duels lost	18	584	2024	–
	GEOS	GENS	STAR3D	# Duels won
GEOS	–	547	947	1494
GENS	200	–	915	1115
STAR3D	30	72	–	102
# Duels lost	230	619	1862	–
	GEOS	GENS	SupeRNAlign	# Duels won
GEOS	–	201	262	463
GENS	1	–	260	261
SupeRNAlign	0	1	–	1
# Duels lost	1	202	522	–
	GEOS	GENS	RMalign	# Duels won
GEOS	–	800	854	1654
GENS	1	–	759	760
RMalign	78	181	–	259
# Duels lost	79	981	1613	–
	GEOS	GENS	RNA-align	# Duels won
GEOS	–	551	872	1423
GENS	262	–	849	1111
RNA-align	68	201	–	269
# Duels lost	330	752	1718	–

Table S2. Lengths of alignments found by GEOS and GENS in three rounds of experiment. Three predicted models were aligned with the target structure of (i) Puzzle 01 (46nts), (ii) Puzzle 03 (84nts), and (iii) Puzzle 04 (126nts). Both algorithms were run with the default RMSD threshold (3.5Å) and performed 125 times for each instance.

Round (i)		GEOS (seq-ind)	GEOS (seq-dep)	GENS (seq-ind)	GENS (seq-dep)
PZ1-Bujnicki-4 vs target	Min length	44	35	42	35
	Max length	44	35	43	35
	Avg length	44.0	35.0	42.9	35.0
	St. dev.	0.0	0.0	0.2	0.0
PZ1-Das-4 vs target	Min length	45	43	44	44
	Max length	45	43	45	44
	Avg length	45.0	43.0	44.8	44.0
	St. dev.	0.0	0.0	0.4	0.0
PZ1-Santalucia- 4 vs target	Min length	36	32	33	32
	Max length	36	32	36	32
	Avg length	36.0	32.0	35.6	32.0
	St. dev.	0.0	0.0	0.9	0.0
Round (ii)					
PZ3-Chen-1 vs target	Min length	53	41	43	42
	Max length	53	42	52	42
	Avg length	53.0	41.9	49.4	42.0
	St. dev.	0.0	0.2	1.6	0.0
PZ3-Das-1 vs target	Min length	38	26	31	25
	Max length	38	26	39	27
	Avg length	38.0	26.0	36.2	25.9
	St. dev.	0.0	0.0	1.9	0.6
PZ3-Dokholyan-2 vs target	Min length	37	29	31	25
	Max length	37	29	40	29
	Avg length	37.0	29.0	36.4	28.3
	St. dev.	0.0	0.0	1.9	0.7
Round (iii)					
PZ4-Adamiak-4 vs target	Min length	97	94	60	87
	Max length	97	94	97	94
	Avg length	97.0	94.00	90.5	91.7
	St. dev.	0.0	0.000	5.5	1.3
PZ4-Bujnicki-1 vs target	Min length	119	120	98	117
	Max length	119	120	120	120
	Avg length	119.0	120.0	115.2	119.7
	St. dev.	0.0	0.0	2.7	0.7
Pz4-Mikolajczak-1 vs target	Min length	101	100	81	93
	Max length	101	100	101	100
	Avg length	101.0	100.0	95.6	98.9
	St. dev.	0.0	0.0	3.4	0.9

Table S3. Dataset used to test execution times of GEOS and GENS applied to find alignment between large RNA structures (100-700nts).

Subset	1	2	3	4	5	6	7	8
Representative	6DME_A	5JUP_EC	7B9V_2+6	7EZ2_N	6ZQc_D2	6N7R_R	6HIW_CA	8H2H_A
Size [nts]	100	198	298	392	446	558	621	692
Class id	86427.1	99969.1	1315.2	60848.2	62396.2	34961.3	60828.6	56999.3
# members	5	7	7	11	3	6	5	4

Nucleic Acids Research, 2024, **52**, W348–W353
https://doi.org/10.1093/nar/gkae259
Advance access publication date: 8 April 2024
Web Server issue

RNAhugs web server for customized 3D RNA structure alignment

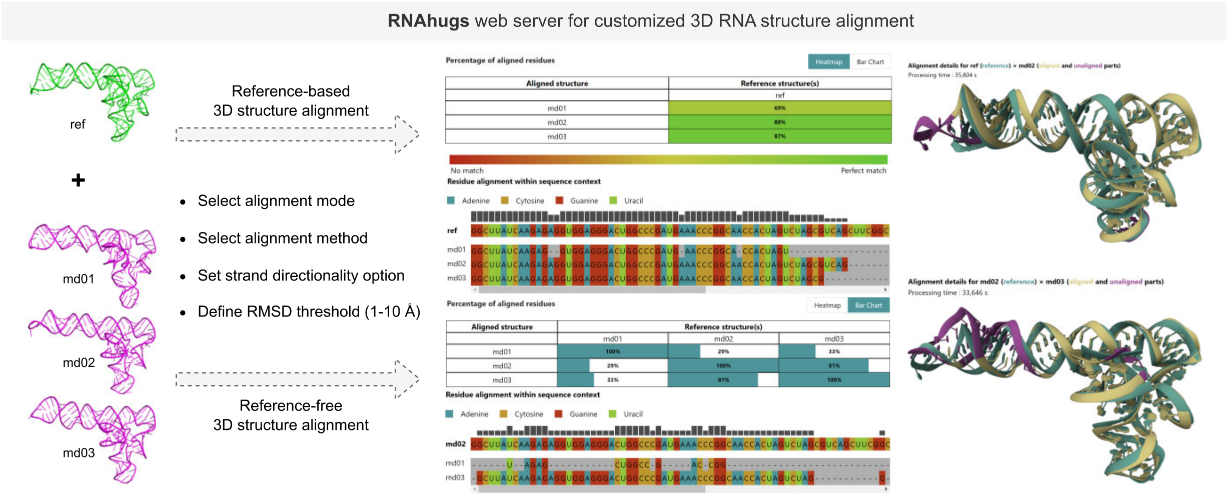
Michal Zurkowski[1], **Mateusz Swiercz**[1], **Filip Wozny**[1], **Maciej Antczak**[1,2,*] and **Marta Szachniuk**[1,2,*]

[1]Institute of Computing Science and European Centre for Bioinformatics and Genomics, Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland
[2]Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12/14, 61-704 Poznan, Poland
[*]To whom correspondence should be addressed. Tel: +48 616652999; Fax: +48 618771525; Email: marta.szachniuk@cs.put.poznan.pl
Correspondence may also be addressed to Maciej Antczak. Email: maciej.antczak@cs.put.poznan.pl

Abstract

Alignment of 3D molecular structures involves overlaying their sets of atoms in space in such a way as to minimize the distance between the corresponding atoms. The purpose of this procedure is usually to analyze and assess structural similarity on a global (e.g. evaluating predicted 3D models and clustering structures) or a local level (e.g. searching for common substructures). Although the idea of alignment is simple, combinatorial algorithms that implement it require considerable computational resources, even when processing relatively small structures. In this paper, we introduce RNAhugs, a web server for custom and flexible alignment of 3D RNA structures. Using two efficient heuristics, GEOS and GENS, it finds the longest corresponding fragments within 3D structures that may differ in sizes—given in the PDB or PDBx/mmCIF formats—that manage to align with user-specified accuracy (i.e. with an RMSD not exceeding a cutoff value given as an input parameter). A distinctive advantage of the system lies in its ability to process multi-model files and compare the results of 1–25 alignments in a single task. RNAhugs has an intuitive interface and is publicly available at https://rnahugs.cs.put.poznan.pl/.

Graphical abstract



- Select alignment mode
- Select alignment method
- Set strand directionality option
- Define RMSD threshold (1-10 Å)

Percentage of aligned residues

Aligned structure	Reference structure(s)
	ref
md01	69%
md02	88%
md03	87%

Residue alignment within sequence context

```
ref   GGCUUAUCAAGAGAGGUGGAGGGACUGGCCCGAUGAAACCCGGCAACCACUAGUCUAGCGUCAGCUUCGGC
md01  GGCUUAUCAAGAG--GUGGAGGGACUGGCCCGAUG-AACCCGGCA-CCACUAGU----------------
md02  GGCUUAUCAAGAGAGGUGGAGGGACUGGCCCGAUGAAACCCGGCAACCACUAGUCUAGCGUCAG------
md03  GGCUUAUCAAGAGAGGUGGAGGGACUGGCCCGAUGAAACCCGGCAACCACUAGUCUAGCG----------
```

Percentage of aligned residues

Aligned structure	Reference structure(s)		
	md01	md02	md03
md01	100%	29%	33%
md02	29%	100%	81%
md03	33%	81%	100%

Residue alignment within sequence context

```
md02  GGCUUAUCAAGAGAGGUGGAGGGACUGGCCCGAUGAAACCCGGCAACCACUAGUCUAGCGUCAGCUUCGGC
md01  ------U--AGAG-----------CUGGCC-G-----AC-CGG--------------------------
md03  -GCUUAUCAAGAGAGGUGGAGGGACUGGCCCGAUGAAACCCGGCAACCACUAGUCUAG----------G-
```

Alignment details for ref (reference) × md02 (aligned and unaligned parts)
Processing time : 35,804 s

Alignment details for md02 (reference) × md03 (aligned and unaligned parts)
Processing time : 33,646 s

Introduction

Recent years have resulted in significant developments in structural biology and bioinformatics. The advent of AlphaFold (1) revolutionized the field of structural biology. The burden of protein structure-centered research shifted from the determination and prediction of 3D folds to an analysis of millions of models generated by this deep learning system. Computational methods supporting such analysis gained importance, including algorithms to compare structural models, identify their common features, discover new motifs, group into families of similar structures, etc. At the same time, the success of AlphaFold has encouraged many protein-focused researchers to apply their experience in the domain of nucleic acids. Thus, RNA molecules, for years outside the mainstream

Received: January 26, 2024. **Revised:** March 5, 2024. **Editorial Decision:** March 26, 2024. **Accepted:** March 27, 2024

research because of the long-held belief in their passivity, came under the spotlight. Only in the last 3 years have more than a dozen new methods emerged to model 3D RNA structures (2), the AlphaFold team has announced its first results in predicting protein-nucleic acid complexes, and an RNA category has been launched in the CASP competition (the Critical Assessment of protein Structure Prediction) (3). This has increased the availability of computer-generated 3D RNA models, while at the same time intensifying the demand for tools to analyze them. Developers and users of predictive systems require methods to evaluate the reliability and quality of their output, especially since many programs produce flawed conformations (4,5). In turn, assessors in structure modeling contests, such as RNA-Puzzles or CASP, try various methods to evaluate submissions (3,6). Finally, tools are needed to determine structure parameters, search for similarities and differences between structures, identify common substructures and structural motifs, etc.

RNAhugs web server introduced in this paper addresses some of the mentioned necessities. Its main task is to align 3D RNA structures to allow for the assessment of their global and local similarity and to find similar substructures. The system engine runs two alternative algorithms for flexible alignment, GEOS, and GENS (7). They identify substructures whose RMSD is in the range [0, *X*] where *X* is the input parameter of the program. On request, they operate in sequence-dependent or sequence-independent mode; in the latter case, similar fragments may have different sequences. RNAhugs allows uploading 1–5 3D models and (optionally) 1–3 reference structures at a time. The structures can vary in size. The program outputs the results of pairwise alignment in numerical and graphical form, all of which are downloadable. The system interface is intuitive, and all its functionality is available without logging in.

Method outline

The RNAhugs workflow diagram is presented in Figure 1. In the first step, the system reads the input data provided via the form on the main page. The front-end then feeds them to the back-end. After the validation protocol ensures that the data are correct, the back-end stores them in a local database and enqueues a computing task. This step involves the generation of a unique URL-embedded task identifier. URL displaying a result page is populated incrementally with output data. It expires 2 weeks after the task completion.

RNAhugs performs reference-based and reference-free alignment. The first runs when users enter reference structures (set *A*) and models to align (set *B*). Each structure of *B* is tried to align with each structure of *A*. If only the set *B* is given, the reference-free alignment starts. It independently processes each pair of structures in this set. A pair of 3D RNA structures, along with configuration parameters, defines the task that is passed to the RNAhugs computational module. The first element of the pair is treated as the reference model. The second is the working structure; its fragments are flexibly aligned. The system starts with cleaning the data by removing all non-RNA chains, ions, ligands, etc., and discarding non-regular atoms. Next, both structures are transformed into a coarse-grained 3-bead representation; each residue is represented by coordinates of phosphorus, the centroid of the ribose group, and the centroid of the nitrogenous base. In the following step, the user-selected heuristic, GEOS (geometric search) or GENS (genetic search), is launched. As GENS works better for similar structures, it is recommended for the alignment of distant homologs or various models of the same structure. GEOS is shown to be better for structures that differ significantly, so we suggest using it to find the maximal common substructures (7). The algorithm runs until it meets one of the stopping criteria, that is, the processing time exceeds 5 min, or there is no improvement in the quality of the solution. The longest 3D structure alignment below the user-determined RMSD threshold is returned. It may consist of multiple disjoint fragments aligned independently of each other. Based on these fragments, RNAhugs computes the translation vector and the rotation matrix. They are used to generate the PDB/mmCIF file with superimposed structures and visualize it on the output. The complete results are stored in the internal database of the system.

Let us underline that the computational layer initializes several instances of the computational module. Their number depends on the available performance of the computing infrastructure. Each module uses concurrency processing to minimize task processing time. This allows multiple tasks to be processed at a time.

Web application

RNAhugs is a multi-container application, containerized with Docker, that ensures high scalability and automated, fast recovery after container failure. In the processing layer, several computational modules are run independently to ensure efficiency. The system engine is implemented in Java 11 with Apache Maven 3.6.3 and bioinformatics-dedicated libraries. It is managed by the control unit responsible for handling exceptions, monitoring computational modules, and recovering connections with the message broker. RabbitMQ, a reliable distributed queueing system, ensures the high server stability and persistence of all submitted tasks. The message broker integrates the processing and business logic layers developed in Python 3 using the Django framework. The back-end also provides a RESTful API via Django REST, integrates the data model layer, and stores all the input and output data in PostgreSQL. The modern, responsive, and user-friendly interface of RNAhugs is implemented in React. UI supports all modern web browsers, platforms, and mobile devices. The front-end can download RNA structures through the PDB API. The output alignments are visualized in Mol* (8). The system operates on a dedicated virtual machine with 8 GB RAM and 4 vCPUs maintained by the Institute of Computing Science, Poznan University of Technology. RNAhugs was tested using SELENIUM IDE and Pytest library.

Input and output description

PDB or PDBx/mmCIF files with RNA structures are the main input to the system. Users can upload 1–3 reference structures (optionally) and 1–5 models to align in a single task. The total size of input data cannot exceed 100 MB. Files can be uploaded from a local drive or from the Protein Data Bank (9). In the first case, one should use dedicated drag & drop panels. In the second one, PDB IDs separated by commas are entered into the edit box. Users decide whether the file contains a reference structure or a model to align, and the file contents are checked for compatibility with supported 3D structure formats. An important advantage of the system is the capacity to

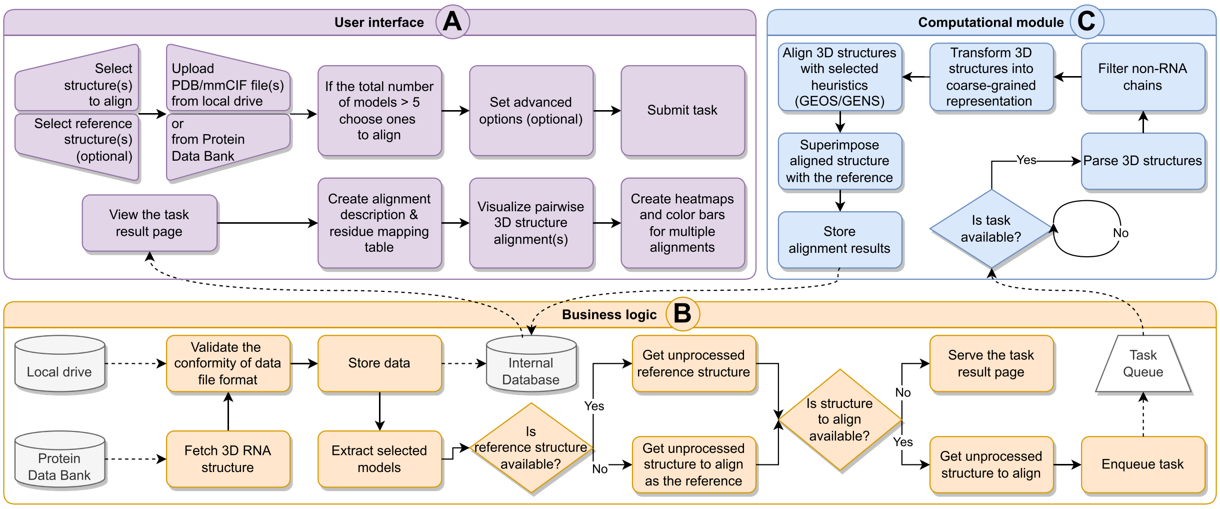

User interface
A
Select structure(s) to align
Upload PDB/mmCIF file(s) from local drive
Select reference structure(s) (optional)
or from Protein Data Bank
If the total number of models > 5 choose ones to align
Set advanced options (optional)
Submit task
View the task result page
Create alignment description & residue mapping table
Visualize pairwise 3D structure alignment(s)
Create heatmaps and color bars for multiple alignments
Computational module
C
Align 3D structures with selected heuristics (GEOS/GENS)
Transform 3D structures into coarse-grained representation
Filter non-RNA chains
Superimpose aligned structure with the reference
Parse 3D structures
Yes
Is task available?
No
Store alignment results
Business logic
B
Local drive
Validate the conformity of data file format
Store data
Internal Database
Get unprocessed reference structure
Serve the task result page
Task Queue
Yes
Is structure to align available?
No
Protein Data Bank
Fetch 3D RNA structure
Extract selected models
Is reference structure available?
No
Get unprocessed structure to align as the reference
Yes
Get unprocessed structure to align
Enqueue task

Figure 1. RNAhugs workflow. The data processing pathway goes as follows: A (provide input data) → B (preprocess the data and build the task queue) → C (compute alignment) → B (store the results in the internal database) → A (display the results).

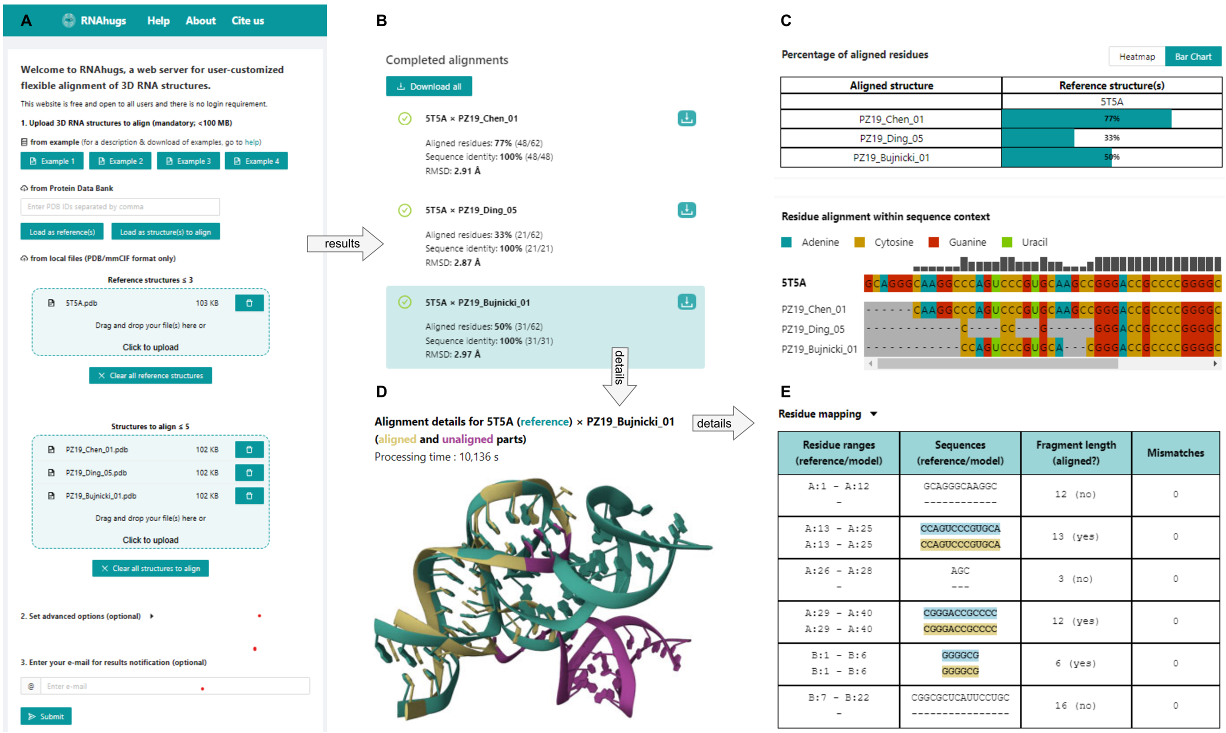
handle multi-model files. If, in such a case, the total number of uploaded models exceeds the imposed limits, the system displays a modal window allowing to reduce the set of structures for analysis (by default, the first model in each file is checked; users can change this selection). Finally, four ready-to-process examples are available for novice users to familiarize them with various system scenarios.

Users can set several advanced options that guide alignment. They include (i) alignment mode (sequence-dependent as default /sequence-independent), (ii) alignment method (geometric search (GEOS) as default /genetic algorithm (GENS)), (iii) respecting strand directionality (on by default), and (iv) RMSD threshold (3.5 Å by default). The first option determines whether the aligned fragments should match sequentially. The second allows for the choice between two heuristic algorithms implemented in the RNAhugs engine, GEOS and GENS, that complement each other (7). GEOS is very efficient for structurally distant models, and GENS is preferable for processing similar structures. Their codes are available on GitHub (https://github.com/RNApolis/rnahugs). In the third option, users decide whether both structures are aligned according to their 5'–3' strand direction. If the option is off, the resultant alignment may be slightly longer, but not necessarily acceptable from a biological point of view. The fourth option specifies the expected accuracy of the solutions, which can take values in the range of 1–10 Å. The system searches for the longest fragments that can be aligned with an RMSD that does not exceed a predefined threshold value. When this option is tested, users can easily find the trade-off between the number of aligned residues and the accuracy of the alignment. Optionally, it is possible to provide an e-mail address to which a notification is sent about the completion of the submitted task with a link to the result page.

RNAhugs supports two processing scenarios, reference-based and reference-free. They are automatically triggered depending on whether users enter the reference structure(s) (first mode) or not (second mode). Assume that *a* denotes the number of reference structures and *b* is the number of models to align. In the first scenario, every structure assigned as the model to align is processed together with the reference (the number of alignments equals $axb \leq 15$). In the second, a pairwise alignment is performed for each pair of uploaded structures (the number of alignments is equal to $bxb \leq 25$). The results are published on the result page which has a unique URL with an embedded task ID. It can be easily bookmarked in the web browser and visited up to 2 weeks after the completion of the task. The result page is divided into several sections. They are successively populated with data as more alignments are completed. The header shows the task ID and the expiration date of the page. The top-right panel displays configuration parameters. The left panel lists all completed alignments and allows navigation between them. Each alignment is described by file names that contain processed structures, the number of aligned residues, the maximum potential alignment (the minimum of the two structure sizes), the percentage of aligned residues, and the actual RMSD of the alignment. A button on the right-hand side of the description enables users to download the alignment results. Additionally, one can save the results of all alignments in a single ZIP archive by clicking on the *Download all* button at the top of the navigation bar.

The upper right panels present the results of a comparative analysis of multiple alignments. They are ready only after completing all the alignments submitted within the task. Toggling between a heat map and a bar chart allows users to see the percentage of aligned residues for all pairs of structures in two different views. The heat map (default view) applies a color scale for this purpose, from red (0%) to green (100%). The following graph presents a projection of aligned residues within the sequence context. The columns in its first (gray) row represent the number of residues in aligned structures that are the same as those in the reference structure. The next row displays the sequence of the reference structure. The following contain aligned fragments of the other structures. Each nucleobase has a unique color. The location of the residue can be seen when the mouse hovers over a cell.

The next two panels show the details for the pair of structures highlighted in the navigation bar. In the integrated Mol* viewer (8), users can see two superimposed RNAs (their cartoon models). The reference structure is colored green, the aligned fragments of the other structure yellow, and the un-



Welcome to RNAhugs, a web server for user-customized flexible alignment of 3D RNA structures.

This website is free and open to all users and there is no login requirement.

1. Upload 3D RNA structures to align (mandatory; <100 MB)

from example (for a description & download of examples, go to help)

from Protein Data Bank

Enter PDB IDs separated by comma

from local files (PDB/mmCIF format only)

Reference structures ≤ 3

5T5A.pdb	103 KB

Drag and drop your file(s) here or

Click to upload

Structures to align ≤ 5

PZ19_Chen_01.pdb	102 KB
PZ19_Ding_05.pdb	102 KB
PZ19_Bujnicki_01.pdb	102 KB

Drag and drop your file(s) here or

Click to upload

2. Set advanced options (optional)

3. Enter your e-mail for results notification (optional)

@ Enter e-mail

Completed alignments

5T5A × PZ19_Chen_01
Aligned residues: **77%** (48/62)
Sequence identity: **100%** (48/48)
RMSD: **2.91 Å**

5T5A × PZ19_Ding_05
Aligned residues: **33%** (21/62)
Sequence identity: **100%** (21/21)
RMSD: **2.87 Å**

5T5A × PZ19_Bujnicki_01
Aligned residues: **50%** (31/62)
Sequence identity: **100%** (31/31)
RMSD: **2.97 Å**

Alignment details for 5T5A (reference) × PZ19_Bujnicki_01 (aligned and unaligned parts)
Processing time : 10,136 s

Percentage of aligned residues

Aligned structure	Reference structure(s)
	5T5A
PZ19_Chen_01	77%
PZ19_Ding_05	33%
PZ19_Bujnicki_01	50%

Residue alignment within sequence context

🟦 Adenine 🟨 Cytosine 🟥 Guanine 🟩 Uracil

```
5T5A              GCAGGGCAAGGCCCAGUCCCGUGCAAGCCGGGACCGCCCCGGGGC
PZ19_Chen_01      ------CAAGGCCCAGUCCCGUGCAAGCCGGGACCGCCCCGGGGC
PZ19_Ding_05      ------------C----CC---G-------GGGACCGCCCCGGGGC
PZ19_Bujnicki_01  ------------CCAGUCCCGUGCA---CGGGACCGCCCCGGGGC
```

Residue mapping

Residue ranges (reference/model)	Sequences (reference/model)	Fragment length (aligned?)	Mismatches
A:1 - A:12 -	GCAGGGCAAGGC ------------	12 (no)	0
A:13 - A:25 A:13 - A:25	CCAGUCCCGUGCA CCAGUCCCGUGCA	13 (yes)	0
A:26 - A:28 -	AGC ---	3 (no)	0
A:29 - A:40 A:29 - A:40	CGGGACCGCCCC CGGGACCGCCCC	12 (yes)	0
B:1 - B:6 B:1 - B:6	GGGGCG GGGGCG	6 (yes)	0
B:7 - B:22 -	CGGCGCUCAUUCCUGC ----------------	16 (no)	0

Figure 2. User interface of RNAhugs: (**A**) submission form, (**B**) navigation bar with completed alignments, (**C**) bar chart with percentage of aligned residues and table showing residue alignment within sequence context for each alignment, (**D**) Mol* visualization and (**E**) residue mapping for selected alignment.

aligned ones magenta. Users can use all the Mol* options through the menu available in this application window. In the table below, each row contains the data (residue range, sequence, fragment length, and the number of mismatches - the latter can be nonzero only in the case of sequence-independent alignment) of one continuous string fragment, which is either aligned or not. The *Resubmit with new settings* button allows users to resubmit the task with the same structures and different settings of advanced options.

Results

User interface

Figure 2 shows screenshots of the RNAhugs web server. Panel in Figure 2A presents the main page with the submission form. Submitting a task redirects to a self-refreshing result page, whose main panels include the navigation bar (Figure 2B), the results of the comparative analysis of multiple alignments (Figure 2C), the visualization of the aligned structures by Mol* (Figure 2D), and the residue mapping table (Figure 2E).

Example alignment results

In the first example, we examine the alignment of two small RNAs. Both are high-resolution crystal structures of the sarcin-ricin domain from *Rattus norvegicus* 28S rRNA; PDB IDs: 430D (10) and 1Q96 (11). 430D, a single-stranded structure with a length of 29nts, has been uploaded as a reference structure. While 1Q96, which has three chains of identical sequence and length of 27nts each, has been treated as a model to align. In the experiment, we have run geometric search implemented in the RNAhugs engine in the sequence-dependent mode and investigated how the RMSD threshold value affects the resulting solution (Table 1). The computing time was around 15 s/run. Figure 3 presents two different alignments found for this pair of structures with RMSD thresholds equal to 2.5 Å and 3.5 Å. In the first case, alignment is found between 430D and 25 residues of 1Q96, chain A (Figure 3A). In the second, 25 residues from chain B and one residue from chain C of 1Q96 align to the reference structure (Figure 3B).

Table 1. GEOS alignments of 430D and 1Q96 for various RMSD thresholds

	Quality of alignment		
RMSD threshold	size [nt /%]	RMSD [Å]	Aligned parts of 1Q96
1.0–2.7	25/86%	0.99	in chain A
2.8–5.7	26/89%	2.75	in chains B, C
5.8–10.0	27/93%	5.63	in chains A, B, C

With the second example, we look at how RNAhugs can handle structures of widely varying sizes, including extremely long RNAs. We have selected the crystal structure of the large ribosomal subunit from *Haloarcula Marismortui* (PDB ID: 1FFK; resolution 2.4 Å) (12) as a reference. The structure containing the sarcin–ricin loop (SRL) from rat 28S rRNA (PDB ID: 430D; resolution 2.1 Å) (10), same as in the first example, has been chosen as the model to align. 1FFK includes 2 RNA chains (23S rRNA of length 2922nts and 5S rRNA of length 122nts) and 27 protein subunits. RNAhugs was run

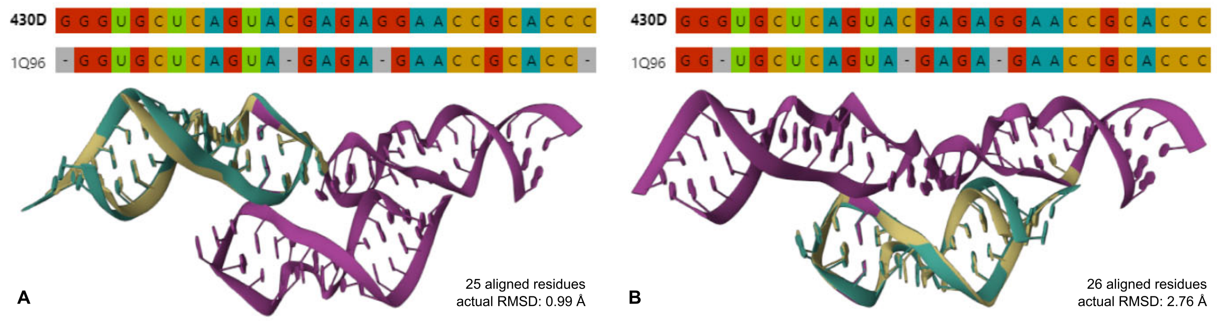

430D GGGUGCUCAGUACGAGAGGAACCGCACCC
1Q96 -GGUGCUCAGUA-GAGA-GAACCGCACC-
25 aligned residues
actual RMSD: 0.99 Å
A
430D GGGUGCUCAGUACGAGAGGAACCGCACCC
1Q96 GG-UGCUCAGUA-GAGA-GAACCGCACCC
26 aligned residues
actual RMSD: 2.76 Å
B

Figure 3. RNAhugs alignment of two sarcin-ricin domains from rat 28S rRNA (PDB IDs: 430D, 1Q96) found for RMSD threshold 2.5 Å (**A**) and 3.5 Å (**B**).

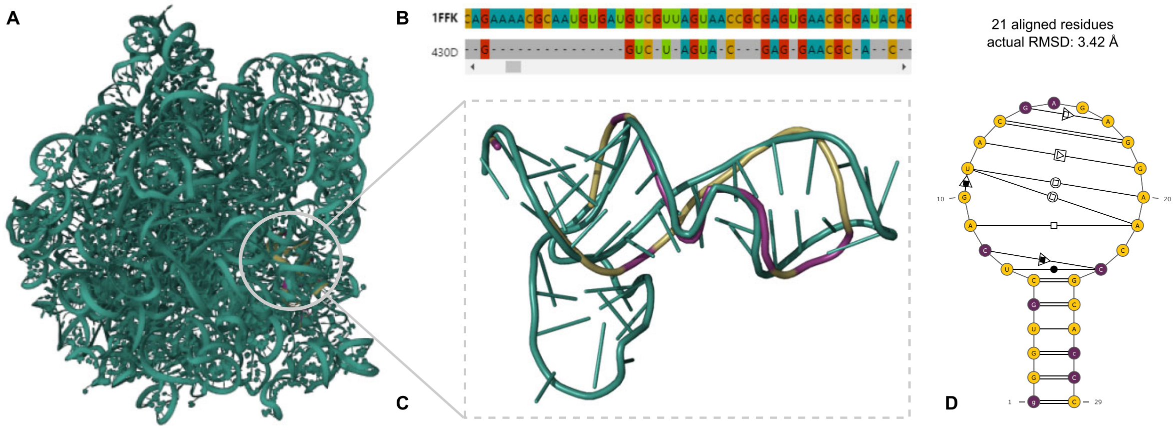


Figure 4. RNAhugs alignment of 23S rRNA from *Haloarcula Marismortui* ribosomal subunit (PDB ID: 1FFK) and SRL-containing structure from rat 28S rRNA (PDB ID: 430D). (**A**) Mol* visualization, (**B**) alignment in sequence context, (**C**) zoomed view of aligned fragments and (**D**) 2D structure of aligned model.

with the default advanced settings. All protein chains were removed from the reference structure. GEOS then looked for the longest sequence-dependent alignment with RMSD ≤3.5 Å. It aligned 21 of 29 (72%) residues of the 430D structure with the 190–235 region of the 23S rRNA chain of the 1FFK structure with the actual RMSD = 3.42 Å (Figure 4). The computation time was 95 s.

Conclusions

RNAhugs is a web server for the structural alignment of 3D RNA models. It supports automatic processing of multimodel files, handling up to 15 structures, and comparative analysis of up to 25 alignments in a single run. The system generates an easy-to-understand and aesthetically pleasing output, including heatmap, visualization of aligned 3D structures, and alignment details in tabular form. The tool is free and open to anyone interested in comparing RNA molecules aimed at identifying similarities and differences in their 3D structures. Plans for developing the system's functionality include adding the possibility to select chains and structural fragments for the alignment, improving the processing of modified residues, centering the Mol* view on a user-designated fragment, and optimizing the usage of RAM in the computational module.

Data availability

RNAhugs is implemented as a publicly available web server with an intuitive interface and can be freely accessed at https://rnahugs.cs.put.poznan.pl/.

Acknowledgements

The authors thank Julia Lamperska and Dominik Ziebicki for their contribution to the pre-alpha version of the application.

Funding

This work was supported by the statutory funds of Poznan University of Technology and the Institute of Bioorganic Chemistry, PAS. The open access publication charge for this paper has been waived by Oxford University Press – *NAR* Editorial Board members are entitled to one free paper per year in recognition of their work on behalf of the journal.

Conflict of interest statement

None declared.

References

1. Jumper,J., Evans,R., Pritzel,A., Green,T., Figurnov,M., Ronneberger,O., Tunyasuvunakool,K., Bates,R., Žídek,A., Potapenko,A., *et al.* (2021) Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**, 583–589.
2. Schneider,B., Sweeney,B.A., Bateman,A., Cerny,J., Zok,T. and Szachniuk,M. (2023) When will RNA get its AlphaFold moment?. *Nucleic Acids Res.*, **51**, 9522–9532.
3. Kryshtafovych,A., Antczak,M., Szachniuk,M., Zok,T., Kretsch,R.C., Rangan,R., Pham,P., Das,R., Robin,X., Studer,G., *et al.* (2023) New prediction categories in CASP15. *Proteins: Struct. Funct. Bioinf.*, **91**, 1550–1557.
4. Carrascoza,F., Antczak,M., Miao,Z., Westhof,E. and Szachniuk,M. (2021) Evaluation of the stereochemical quality of predicted RNA 3D models in the RNA-Puzzles submissions. *RNA*, **28**, 250–262.
5. Popenda,M., Zok,T., Sarzynska,J., Korpeta,A., Adamiak,R., Antczak,M. and Szachniuk,M. (2021) Entanglements of structure elements revealed in RNA 3D models. *Nucleic Acids Res.*, **49**, 9625–9632.
6. Magnus,M., Antczak,M., Zok,T., Wiedemann,J., Lukasiak,P., Cao,Y., Bujnicki,J., Westhof,E., Szachniuk,M. and Miao,Z. (2020) RNA-Puzzles toolkit: A computational resource of RNA 3D structure benchmark datasets, structure manipulation, and evaluation tools. *Nucleic Acids Res.*, **48**, 576–588.
7. Zurkowski,M., Antczak,M. and Szachniuk,M. (2023) High-quality, customizable heuristics for RNA 3D structure alignment. *Bioinformatics*, **39**, btad315.
8. Sehnal,D., Bittrich,S., Deshpande,M., Svobodova,R., Berka,K., Bazgier,V., Velankar,S., Burley,S., Koca,J. and Rose,A. (2021) Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic Acids Res.*, **49**, W431–W437.
9. Berman,H., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T., Weissig,H., Shindyalov,I. and Bourne,P. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
10. Correll,C., Munishkin,A., Chan,Y., Ren,Z., Wool,I. and Steitz,T. (1998) Crystal structure of the ribosomal RNA domain essential for binding elongation factors. *Proc. Natl. Acad. Sci. U.S.A.*, **95**, 13436–13441.
11. Correll,C., Beneken,J., Plantinga,M., Lubbers,M. and Chan,Y. (2003) The common and the distinctive features of the bulged-G motif based on a 1.04 Å resolution RNA structure. *Nucleic Acids Res.*, **31**, 6806–6818.
12. Ban,N., Nissen,P., Hansen,J., Moore,P. and Steitz,T. (2000) The complete atomic structure of the large ribosomal subunit at 2.4 Å resolution. *Science*, **289**, 905–920.

Received: January 26, 2024. **Revised:** March 5, 2024. **Editorial Decision:** March 26, 2024. **Accepted:** March 27, 2024

Co-author declarations

28/08/2024

Declaration

I hereby declare that the contributions to the following paper

Tomasz Zok, Natalia Kraszewska, Joanna Miskiewicz, Paulina Pielacinska, Michal Zurkowski, Marta Szachniuk (2021) *ONQUADRO: a database of experimentally determined quadruplex structures*. Nucleic Acids Research, 50(D1), D253–D258 (doi: 10.1093/nar/gkab1118)

are characterized as:

TZ, JM, MZ, and MS designed the database system. TZ created the database schema and developed the auto-update functionality. TZ and MZ implemented the backend. PP, supervised by JM, implemented the web application prototype. NK developed the final version of the web application, with structure visualizations and statistics by MZ. JM and MZ prepared the test sets and conducted extensive system testing. MS supervised the team. TZ, JM, MZ, and MS wrote and revised the manuscript. All authors reviewed and contributed to the final version of the paper.

Name	Abbreviation	Signature
Tomasz Żok	TZ	Tomasz Żok
Natalia Kraszewska	NK	Natalia Kraszewska
Joanna Miśkiewicz	JM	Joanna
Paulina Pielacińska	PP	Pielacińska
Michał Żurkowski	MZ	Michał Żurkowski
Marta Szachniuk	MS	Marta Szachniuk

28/08/2024

Declaration

I hereby declare that the contributions to the following paper

Michal Zurkowski, Tomasz Zok, Marta Szachniuk (2022) *DrawTetrado to create layer diagrams of G4 structures*. Bioinformatics 38(15), 3835–3836 (doi: 10.1093/bioinformatics/btac394)

are characterized as:

MS conceived the project. MZ designed and implemented the DrawTetrado algorithm and tested it on all available quadruplex structures. TZ supported MZ with the integration of ElTetrado and DrawTetrado and assisted in collecting data for algorithm testing. MZ drafted the first version of the manuscript, prepared the figures, and, along with TZ and MS, revised the paper. All authors reviewed and contributed to the final version of the manuscript.

Name	Abbreviation	Signature
Michał Żurkowski	MZ	Michał Żurkowski
Tomasz Żok	TZ	Tomasz Żok
Marta Szachniuk	MS	Marta Szachniuk

28/08/2024

Declaration

I hereby declare that the contributions to the following paper

Bartosz Adamczyk, Michal Zurkowski, Marta Szachniuk, Tomasz Zok (2023) *WebTetrado: a webserver to explore quadruplexes in nucleic acid 3D structures*. Nucleic Acids Research 51(W1), W607–W612 (doi: 10.1093/nar/gkad346)

are characterized as:

MZ and TZ designed the system. BA implemented the web application with support from MZ. MZ and TZ developed the backend. MZ collected structural data for the test set and validated the system through extensive benchmarking. MS conceived the project and supervised the team. MZ, MS, and TZ wrote and revised the manuscript. All authors reviewed and contributed to the final version of the publication.

Name	Abbreviation	Signature
Bartosz Adamczyk	BA	
Michał Żurkowski	MZ	Michał Żurkowski
Marta Szachniuk	MS	Marta Szachniuk
Tomasz Żok	TZ	Tomasz Żok

28/08/2024

Declaration

I hereby declare that the contributions to the following paper

Michal Zurkowski, Maciej Antczak, Marta Szachniuk (2023) *High-quality, customizable heuristics for RNA 3D structure alignment*. Bioinformatics 39(5), btad315 (doi: 10.1093/bioinformatics/btad315)

are characterized as:

MZ designed and implemented the GEOS and GENS algorithms under the supervision of MS. MZ gathered and installed all state-of-the-art applications for structural alignment and benchmarked them against GEOS and GENS. MA supported MZ in collecting test data and interpreting the results provided by competing methods. MZ drafted the first version of the manuscript and prepared all figures under the guidance of MS. All authors reviewed and contributed to the final version of the publication.

Name	Abbreviation	Signature
Michał Żurkowski	MZ	Michał Żurkowski
Maciej Antczak	MA	
Marta Szachniuk	MS	Marta Szachniuk

28/08/2024

Declaration

I hereby declare that the contributions to the following paper

Michal Zurkowski, Mateusz Swiercz, Filip Wozny, Maciej Antczak, Marta Szachniuk (2024) *RNAhugs web server for customized 3D RNA structure alignment*. Nucleic Acids Research 52(W1), W348–W353 (doi: 10.1093/nar/gkae259)

are characterized as:

MS conceived the project. MZ designed the system with support from MA. MŚ and FW implemented the frontend under MA's supervision. MZ developed the system's backend and prepared the test dataset for benchmarking. Extensive testing and validation of the final system were conducted by MZ with contributions from MA and MS. MZ drafted the first version of the paper and prepared most of the figures. All authors reviewed and contributed to the final version of the publication.

Name	Abbreviation	Signature
Michał Żurkowski	MZ	Michał Żurkowski
Mateusz Świercz	MŚ	
Filip Woźny	FW	Filip Woźny
Maciej Antczak	MA	
Marta Szachniuk	MS	Marta Szachniuk