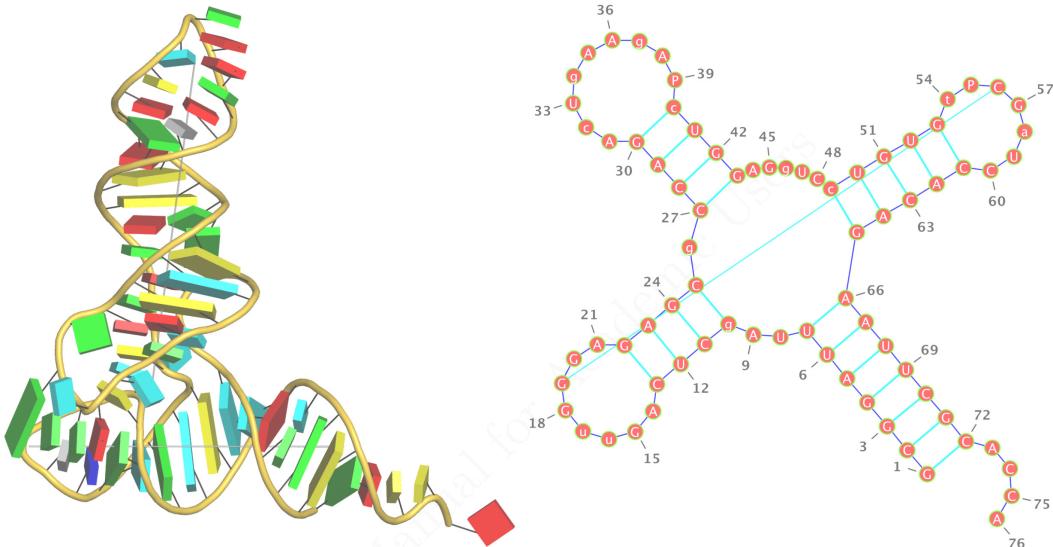




DSSR: an integrated software tool for Dissecting the Spatial Structure of RNA

User Manual for DSSR v2.6.0 (2025-07-24)

Licensed by Columbia University to Academic Users Only



Software to identify, annotate, and quantify structural features of nucleic acids, including:
modified nucleotides, non-canonical base pairs, helices, stems, coaxial stacks
hairpin/internal/junction loops, kink turns, G-quadruplexes, i-motifs, pseudoknots
comprehensive characterization of DNA-protein and RNA-protein interactions
cartoon-block innovative schematics in PyMOL, SQL-like feature queries in Jmol
in silico base mutations, regular models, customized rebuilding, template-based modeling

Xiang-Jun Lu¹, Harmen J. Bussemaker¹, and Wilma K. Olson²

¹Department of Biological Sciences, Columbia University, New York, NY 10027

²Department of Chemistry and Chemical Biology, Rutgers University, Piscataway, NJ 08854



- DSSR is licensed by Columbia University. It is distributed in two variants:
 1. DSSR for academic users covers features described in the DSSR papers (Lu *et al.*, 2015; Hanson and Lu, 2017; Lu, 2020), the 3DNA papers (Lu and Olson, 2003, 2008; Li *et al.*, 2019), and new SNAP functionalities for the analysis of protein-nucleic acid structures (Kribelbauer *et al.*, 2020).
 2. DSSR Pro for commercial users includes advanced capabilities for model building, additional features in structural analyses, annotations, and visualizations, plus direct support.
- This manual documents DSSR available to *Academic users*.
- The manual is continuously revised and expanded, based on user feedback and further development of DSSR.
- DSSR is designed to work as documented in the manual; any identified discrepancies will be promptly addressed.
- Issues related to the manual can be reported on the 3DNA Forum or directed to xiangjun@x3dna.org.

Contents

1	Introduction	6
1.1	Structural analysis and annotation	7
1.2	Integrations into Jmol and PyMOL	9
1.3	Versatile modeling capabilities	9
1.4	Quality control of DSSR	10
2	Download and installation	11
3	Analysis and annotation	12
3.1	Command-line help	12
3.2	Default run on 1msy (27-nt RNA), with detailed explanations	15
3.2.1	Overview of command runs and outputs	16
3.2.2	The summary section	17
3.2.3	Base pairs	18
M+N vs. M–N pairs	19	
Common names	22	
Saenger classification	22	
Leontis-Westhof (LW) classification	22	
DSSR classification	24	
LW vs. DSSR classifications	24	
3.2.4	Multiplets (higher-order coplanar base associations)	25
3.2.5	Helices (stacked pairs of any type or connectivity)	26
3.2.6	Stems (canonical pairs with continuous backbones)	27
3.2.7	Isolated canonical pairs	28
3.2.8	Base stacks	28
3.2.9	Atom-base capping interactions	29
3.2.10	Various loops	29
3.2.11	Single-stranded fragments	31
3.2.12	2D structure in dot-bracket notation (.dbn) and .ct format	32
3.2.13	Summary of structural features per nucleotide	33
3.2.14	Backbone conformations	34
Main chain conformational parameters	34	
Virtual torsion angles	35	

Sugar conformational parameters	36
Assignment of backbone suite names	37
3.3 Default run on 1ehz (76-nt tRNA ^{Phe}), summary notes	38
3.3.1 Brief summary	39
3.3.2 Modified nucleotides	40
3.3.3 The four triplets	40
3.3.4 Coaxial stacks (relationship between helix and stems)	41
3.3.5 Three hairpin loops	41
3.3.6 One four-way junction loop	41
3.3.7 Splayed-apart conformations	42
3.3.8 Pseudoknot	43
3.4 Default run on 1jj2 (50S subunit), four motifs	44
3.4.1 Kissing loops	45
3.4.2 A-minor motifs	46
3.4.3 Ribose zippers	49
3.4.4 Kink turns	50
3.5 The U-turn motifs	51
3.6 Identification and characterization of G-quadruplexes	52
3.6.1 Existing issues	53
3.6.2 DSSR solves known problems	53
3.6.3 Survey of G-quadruplexes in the PDB	57
3.7 Detection and characterization of i-motifs	57
3.8 Identification and removal of pseudoknots	58
3.8.1 Higher-order pseudoknots	59
3.8.2 Pseudoknot removal	60
3.9 The --pair-list options	60
The --pair-list option by default	60
The --pair-list-outfile option	61
The --pair-list-inpfile option	62
3.10 The --more option	62
3.10.1 Extra characterizations of base pairs	62
3.10.2 Orientation of helices/stems	63
3.10.3 Base-pair morphology parameters for helices/stems	64
3.11 The --idstr option	65

3.12 The <code>--json</code> option	66
3.13 The <code>--nmr</code> option	67
3.14 The <code>--pair-only</code> option	69
3.15 The <code>--non-pair</code> option (interactions other than pairing)	69
3.16 The <code>--po4</code> option (phosphate interactions)	70
3.17 Miscellaneous options	71
3.17.1 The <code>--symmetry</code> option	71
3.17.2 The <code>--isolated-pair</code> option	71
3.17.3 The <code>--nt-mapping</code> option	72
3.17.4 The <code>--prefix</code> option	72
3.17.5 The <code>--auxfile</code> option	72
3.17.6 The <code>--dbn-break</code> option	73
3.17.7 The <code>--sugar-pucker</code> option	73
3.17.8 The <code>--torsion360</code> option	73
3.17.9 The <code>--raw-xyz</code> option	73
3.17.10 The <code>--cleanup</code> option	74
3.18 The <code>analyze</code> module (following 3DNA)	74
3.18.1 Analysis of double helices	74
3.18.2 Analysis of single-stranded structures	78
4 Visualization features	80
4.1 DSSR-Jmol integration	80
4.2 DSSR-PyMOL integration	81
4.3 Visualization-related options	83
4.3.1 The <code>--hbfile-pymol</code> option	83
4.3.2 The <code>--hbfile-jmol</code> option	83
4.3.3 The <code>--helical-axis</code> option	83
4.3.4 The <code>--simple-junction</code> option	85
5 Modeling capabilities	85
5.1 <i>In silico</i> mutations (<code>mutate</code>)	86
5.1.1 Introduction	86
5.1.2 The <code>--enum</code> option	87
5.1.3 The <code>--list</code> option	89
5.1.4 The <code>--entry</code> option	89

5.1.5	The <code>--mutate-type</code> option	90
5.2	Regular helical models (<code>fiber</code>)	90
5.2.1	Introduction	90
5.2.2	The <code>--list</code> option	91
5.2.3	The <code>--model</code> option	93
5.2.4	The <code>--sequence</code> option	93
5.2.5	Other options and some examples	93
5.3	Customized structures (<code>rebuild</code> , following 3DNA)	94
5.3.1	Introduction	94
5.3.2	The <code>--backbone</code> option	95
5.3.3	Other options and some examples	96
6	Utilities of general purposes	99
6.1	The <code>--select</code> option	99
6.2	The <code>--view</code> option	100
6.3	The <code>--frame</code> option	100
6.4	The <code>--get-hbond</code> option	101
7	Frequently asked questions (FAQs)	103
7.1	What does DSSR stand for?	103
7.2	How does DSSR compare with other tools?	103
7.3	How is DSSR related to 3DNA?	104
8	SNAP for nucleic-acid/protein complexes	105
9	Acknowledgements	106
References		106

List of Figures

1	Definitions of key nucleic acid structural components in DSSR	8
2	Cover images of the <i>RNA Journal</i> in 2025	10
3	3D and 2D structures of PDB entry 1msy	15
4	The M–N vs. M+N classification of base pairs	19

5	Pictorial definitions of rigid-body base morphology parameters	20
6	Six types of G·A pairs involving the minor-groove edge of G	21
7	Issues with the three edges in the Leontis-Westhof definition	23
8	Base triplet GUA in PDB entry 1msy	26
9	Common RNA secondary structural elements	30
10	3D and 2D structures of PDB entry 1ehz (tRNA ^{Phe})	39
11	Splayed-apart unit with four nucleotides in 1ehz (tRNA ^{Phe})	43
12	Base pentaplet (AUUAG) identified in PDB entry 1jj2 (50S subunit)	45
13	Kissing-loop motif identified in PDB entry 1jj2 (50S subunit)	47
14	Types I and II A-minor motifs in PDB entry 1jj2 (50S subunit)	48
15	Ribose zipper identified in PDB entry 1jj2 (50S subunit)	49
16	A standard kink-turn identified in PDB entry 1jj2 (50S subunit)	51
17	Three different types of U-turns	52
18	Definition of various types of DSSR blocks	54
19	DSSR-enabled innovative visualization of G-quadruplexes	55
20	Topological descriptors of G-quadruplexes	56
21	Schematic representations of two i-motifs	58
22	Higher-order pseudoknots in PDB entry 1ddy	59
23	DSSR-Jmol featured in the cover image of NAR	82
24	Molecular images with DSSR-detected H-bonds	84
25	RNA rebuilding with DSSR	98

1 Introduction

As the number of experimentally solved RNA-containing structures deposited in the PDB (Burley *et al.*, 2018) grows, it is becoming increasingly important to characterize the geometric features of these molecules consistently and efficiently. Existing RNA bioinformatics tools (Lemieux and Major, 2002; Yang *et al.*, 2003; Sarver *et al.*, 2008) are fragmented, and suffer in either scope or usability. DSSR (Lu *et al.*, 2015) is an integrated software tool for Dissecting the Spatial Structure of RNA. It has been designed from ground up to streamline the analyses, annotations, visualization, and modeling of 3D RNA structures. Despite its acronym, DSSR is actually not limited to RNA; it works on any nucleic-acid-containing

structures, including DNA-protein and RNA-protein complexes. While PDB entries represent a typical use case, 3D nucleic acids structures derived from other sources, including molecular dynamic (MD) simulations (Tan *et al.*, 2018), can also be handled by DSSR.

DSSR delivers a great user experience by solving problems and saving time. It is a ‘solid software product’ for structural bioinformatics of nucleic acids.

1.1 Structural analysis and annotation

Figure 1 outlines some key algorithms underlying DSSR. Starting from a 3D nucleic-acid-containing structure in PDB or mmCIF format, DSSR uses standard atom names and base planarity to detect nucleotides, including modified ones (Figure 1A). It employs the standard base reference frame (Olson *et al.*, 2001, Figure 1B,C) and a set of simple geometric criteria (Figure 1D) to identify all existent base pairs: either canonical Watson-Crick (WC) and wobble pairs or non-canonical pairs with at least one hydrogen bond (H-bond). The latter pairs may include normal or modified bases, regardless of tautomeric or protonation state. DSSR uses the six standard rigid-body base-pair parameters (shear, stretch, stagger, propeller, buckle, and opening) to quantify the spatial disposition of any two interacting bases. Where applicable, the program also denotes a base-pair by common names (including WC, wobble G–U, reverse WC, Hoogsteen, reverse Hoogsteen, sheared G–A, Calcutta U–U, dinucleotide platform, etc.), the Saenger (1984) classification scheme of 28 H-bonding types, and the Leontis and Westhof (2001, LW) nomenclature of 12 basic geometric classes.

DSSR detects multiplets (triplets or higher-order base associations) by searching horizontally in the plane of the associated base pair for further H-bonding interactions. The program determines double-helical regions (Figure 1E) by exploring vertically in the neighborhood of selected base-pairs for stacking interactions, regardless of pairing type or backbone connection (e.g., coaxial stacks). DSSR then identifies hairpin loops, bulges, internal loops, and multi-branch (junction) loops (Figure 1F). The program outputs RNA secondary structure in three commonly used formats—dot-bracket notation (dbn), connectivity table (.ct), and base-pair sequence (.bpseq)—that can be fed directly into visualization tools such as VARNA (Darty *et al.*, 2009). DSSR derives proper dbn for RNA with higher-order pseudoknots, and it can also produce pseudoknot-free secondary structures.

In DSSR, each helix/stem is characterized by a least-squares fitted helical axis, and dinucleotide steps are classified into the most common A-, B-, or Z-type double helical forms (where appropriate) and quantified by local step and helical parameters. DSSR calculates

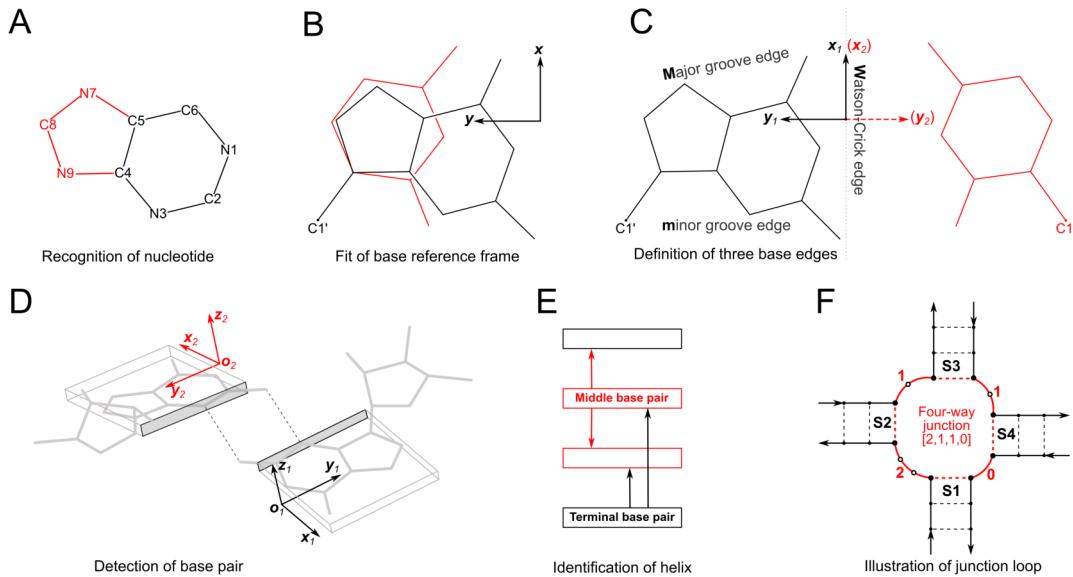


Figure 1: Definitions of key nucleic acid structural components in DSSR [Reproduced from Figure 1 of Lu *et al.* (2015)]. (A) Nucleotides are recognized using standard atom names and base planarity. This method works for both standard (A, C, G, T and U) and modified nucleotides, regardless of their tautomeric or protonation states. (B) Bases are assigned a standard reference frame (Olson *et al.*, 2001) that is independent of sequence identity: purines and pyrimidines are symmetrically placed with respect to the sugar. (C) The standard base frame is derived from an idealized Watson-Crick base pair, and defines three base edges (Watson-Crick, minor groove, and Major groove) that are used to classify pairing interactions. (D) Base pairs are identified from the coplanarity of base rings and the occurrence of H-bonds. This geometric algorithm can find canonical (Watson-Crick and G-U wobble) as well as non-canonical pairs. Higher-order (three or more) coplanar base associations, termed multiplets, are also detected. (E) Helices are defined by stacking interactions of base pairs, regardless of pairing type (canonical or otherwise) or backbone connectivity (covalently connected or broken). A helix consists of at least two base pairs. A stem is defined as a special type of helix, made up of canonical pairs with a continuous backbone along each strand. Coaxial stacking is defined by the presence of two or more stems within one helix. (F) ‘Closed’ loops of various types (hairpin, bulge, internal, and junction loops) are delineated by stems, and specified by the lengths of the intervening, consecutive nucleotide segments.

commonly used backbone torsion angles, including three sets of virtual η/θ torsions (Li *et al.*, 2019), classifies the backbone into BI/BII conformations and the sugar into C2'/C3'-endo puckers, and assigns the consensus RNA backbone suite names (Richardson *et al.*, 2008). The program automatically identifies A-minor interactions, splayed-apart dinucleotide conformations, atom-base capping interactions, ribose zippers, G-quadruplexes, i-motifs, kissing loops, U-turns, k-turns, etc. DSSR also reports non-pairing interactions (H-bonding or base-stacking) between two nucleotides and contacts involving phosphate groups.

1.2 Integrations into Jmol and PyMOL

Working with Robert Hanson and Thomas Holder, respectively, we initiated the integrations of DSSR into Jmol and PyMOL, two of the most popular molecular viewers. The DSSR-Jmol integration excels in its SQL-like, flexible searching capability of structural features. The DSSR-PyMOL integration, on the other hand, stands out for the appealing cartoon-block schematics it brings (see Figure 2).

The creation of most figures in this manual was facilitated by the DSSR-PyMOL integration. Dozens of cover images of *RNA* in recent years have also been generated using DSSR and PyMOL by the Nucleic Acid Knowledgebase (nakb.org) and more recently by the X3DNA-DSSR resource. Notably, the skmatic.x3dna.org website has been recommended in Faculty Opinions as “simple and effective”. It is classified as “Good for Teaching”.

1.3 Versatile modeling capabilities

DSSR has been augmented with a module for *in silico* base mutations that are context sensitive. Powered by the analysis engine already in DSSR, this modeling module allows users to perform base mutations with unprecedented flexibility and convenience. By default, the mutation preserves both the geometry of the sugar-phosphate backbone and the base reference frame (position and orientation). As a result, re-analyzing the mutated model gives the same base-pair and step parameters as those of the original structure.

From early on, 3DNA (Lu and Olson, 2003, 2008; Li *et al.*, 2019) contains the **fiber** program that can be used to build models of over fifty types of uniform helical structures. These regular models are based primarily on the fiber diffraction work of Arnott (1999). The **fiber** module within DSSR has replaced the 3DNA **fiber** program, enhancing its usability.

The 3DNA **rebuild** program complements the **analyze** routine. These two programs are



Figure 2: Cover images of *RNA* in 2025. Note from the Front Matter: “Cover image provided by X3DNA-DSSR, an NIGMS National Resource for structural bioinformatics of nucleic acids (R24GM153869; skmatics.x3dna.org). Image generated using DSSR and PyMOL (Lu XJ. 2020. *Nucleic Acids Res* **48**: e74).” As a matter of fact, a total of 30 cover images of *RNA* in 2021 (12), 2022 (12) and 2023 (6) were generated by the Nucleic Acid Knowledgebase (nakb.org) (Lawson *et al.*, 2024) using DSSR-PyMOL integration (Lu, 2020).

a defining feature of 3DNA (Lu and Olson, 2003, 2008; Li *et al.*, 2019). The DSSR analyze module has completely surpassed the 3DNA analyze program. Similarly, the rebuild module in DSSR replaces the 3DNA rebuild program, with enriched functionality.

1.4 Quality control of DSSR

DSSR is written in strict ANSI C, as a single command-line program. It is self-contained, with zero runtime dependencies on third-party libraries. The program has been extensively tested using all nucleic-acid-containing structures in the PDB and is continuously developed following user feedback. It is also regularly checked using Valgrind to avoid memory leaks. DSSR is efficient and robust, with sensible defaults for input and intuitive outputs, making it accessible to a broad audience.

Overall, DSSR possesses an unmatched set of features, far more than a typical user would normally need. Some new features in DSSR or variations of well-known ones may come as a surprise even to experts in the field. By connecting dots in DNA/RNA structural bioinformatics, DSSR makes common tasks simple and sophisticated applications feasible.

The design and implementation of DSSR have benefited greatly from our extensive user-support experience, improved programming skills, and accumulated domain knowledge. Considering usability, interoperability, features, and support, DSSR easily stands out among its ‘competitors’.

As always, we greatly appreciate user feedback. So far, all reported bugs have always been promptly responded, and fixed where appropriate. The program has been checked using all nucleic acid-containing entries in the PDB, without any known issues. We strive to make DSSR a pragmatic tool that the DNA/RNA structural bioinformatics community can count on.

When communicating, please provide the specific DSSR version you are using (with `-v` or `-version`). This manual corresponds to the version (`x3dna-dssr -v`):

DSSR: an Integrated Software Tool for
Dissecting the Spatial Structure of RNA
v2.6.0-2025jul24 by xiangjun@x3dna.org

2 Download and installation

DSSR has been licensed by Columbia Technology Ventures (CTV). CTV distributes DSSR Academic in one bundled ZIP file[†] composing of three ZIP files for Linux, macOS, and Windows. Each OS-specific ZIP file[‡] contains the corresponding DSSR binary and the user manual in PDF.

The DSSR binary is named `x3dna-dssr` for Linux and macOS, and `x3dna-dssr.exe` for Windows. On Linux or macOS, one *may* need to run the command `chmod +x x3dna-dssr` to make the DSSR binary executable. On Windows, the `.exe` extension in `x3dna-dssr.exe` already means it is an executable file. The binary executable `x3dna-dssr` will be used throughout this manual. It also works in the Windows terminal or WSL2 (Windows Subsystem for

[†]e.g., `dssr-basic-linuxMacWindows-v2.6.0.zip`

[‡]e.g., `dssr-basic-linux-v2.6.0.zip`

Linux).

Move `x3dna-dssr` into a folder on your command search path (`$PATH`, e.g., `~/bin`) so you can run the program conveniently from anywhere. You can also put `x3dna-dssr` in any folder (e.g., your current working directory), and then specify its path explicitly to execute the program.

Type `x3dna-dssr -h` to verify your installation. Note that DSSR is a command-line program: you need a terminal window to run it. Simply double-click `x3dna-dssr` (e.g., in macOS) does not work: the program will exit immediately[†].

The DSSR binary is tiny (<2MB in size) and completely self-contained. The user manual should work as described and includes thorough examples and explanations. Thus, getting DSSR up and running is straightforward: there is no need for setup or configurations.

DSSR is stable in terms of fundamental functionality and main output (especially with JSON) format. On the other hand, DSSR is being actively maintained and developed, with new features added and known bugs fixed. If you only need primary features, you may not need to bother with (frequent) updates. Nevertheless, users are advised to keep up to date: it is as simple as downloading DSSR again from the CTV website to replace (i.e., overwrite) the old copy.

3 Analysis and annotation

The overview (Section 1) summarizes DSSR’s major functionalities in dry, ‘abstract’ technical terms. Using DSSR effectively, however, is a simple process which is best illustrated with concrete examples.

3.1 Command-line help

As is the norm of Linux/Unix command-line tools, running `x3dna-dssr` with the `-h` (or `--help`) option provides help information. The key component of the ‘`x3dna-dssr -h`’ output is shown below:

```
Usage: x3dna-dssr [options]
```

[†]With the error message: `missing required option: must specify -i=PDBFile/mmCIF`

Each option is specified via `--key[=val]` (or `-key[=val]` or `key[=val]`); i.e., two/one/zero preceding dashes are all accepted), where 'key' can be in either lower, UPPER or MiXed case. Options can be in any order.

Options:

```
--help          Print this command-line help information (-h)
--version       Print version number and exit (-v)
--citation      Print preferred citation(s) and exit
--input=file    Specify a PDB/mmCIF file for analysis (-i=file)
--output=file   Designate the main DSSR output file (-o=file)
--more          Report detailed bp and step/helical parameters
--non-pair      Check non-pairing H-bonds/stacking interactions
--pair-only     Output just base-pairing information [10x faster]
--json          Generate output in JSON format for easy parsing
--nmr           Process an ensemble of NMR structures
--blocview      Generate cartoon/blocks, in extended view, in PyMOL
--frame         Reorient a structure with specified base/pair frame
--get-hbond     Report a list of H-bonds within nucleic acids
                  with --get-hbond=protein to include H-bonds in proteins
```

```
*****
analyze  Output structural parameters following 3DNA v2.4 and generate
          input files for the 'rebuild' module
rebuild   Create customized DNA/RNA models per rigid-body parameters
mutate    Perform in silico base mutations that are context sensitive
fiber     Generate regular models of helical DNA and RNA structures
snap      Characterize features of DNA/RNA-protein interactions
          For further information, please see the DSSR User Manual
*****
```

Examples:

```
x3dna-dssr -i=1msy.pdb
x3dna-dssr -i=1msy.cif --more --non-pair
x3dna-dssr -i=1ehz.pdb -o=1ehz.out # yeast phenylalanine tRNA
x3dna-dssr -i=1jj2.pdb -o=1jj2.out # large ribosomal subunit
x3dna-dssr -i=5afi.cif -o=5afi.out # ribosome (only in .cif)
x3dna-dssr -i=5afi.cif -o=5afi-pairs.out --PAIR-ONLY # 10x faster
x3dna-dssr -i=1msy.pdb --json -o=1msy.json
  # x3dna-dssr -i=1msy.pdb --json | jq .pairs
x3dna-dssr -i=5afi.cif --json --pair-only
x3dna-dssr -i=2n2d.pdb --nmr --json -o=2n2d-dssr.json
x3dna-dssr -i=1msy.pdb --blocview --block-file=wc -o=1msy.pml
  # load '1msy.pml' into PyMOL for interactive visualization
x3dna-dssr -i=1msy.pdb --frame='A.2658-wc-minor' -o=1msy-bp.pdb
  # set 1msy into the minor-groove view of the C2658-G2663 pair
x3dna-dssr -i=1oct.pdb --get-hbond # H-bonds in DNA/RNA (default)
x3dna-dssr -i=1oct.pdb --get-hbond --json -o=hbonds.json
  # The JSON file 'hbonds.json' contains ALL types of H-bonds:
jq '.hbonds[] | select(.residue_pair=="nt:nt")' hbonds.json
jq '.hbonds[] | select(.residue_pair=="nt:aa")' hbonds.json
jq '.hbonds[] | select(.residue_pair=="aa:aa")' hbonds.json

x3dna-dssr snap -h # help for the SNAP module on the interactions
  # of DNA-protein or RNA-protein complexes
```

The message above should be sufficient for most users to get started with DSSR. It is worth noting that the command-line interface is consistent across all operating systems, including Windows.

DSSR uses a consistent and flexible way to process command-line options. Each option

can be specified via a `--key[=value]` pair, `-key[=value]`, or `key[=value]`. In other words, two/one/zero preceding dashes are all acceptable. The key can be in lower, UPPER or MiXed case, and the value is optional for Boolean switches. Moreover, options can be put in any order; if the same key is repeated more than once, the value specified with the last key prevails. Some typical use cases are given below:

```
#1 analyze the PDB entry '1msy', with default output to stdout
x3dna-dssr --input=1msy.pdb

#2 same as #1, with output directed to file '1msy.out'
x3dna-dssr --input=1msy.pdb --output=1msy.out

#3-6, same as #2
x3dna-dssr --output=1msy.out --input=1msy.pdb
x3dna-dssr --OUTPUT=1msy.out --Input=1msy.pdb
x3dna-dssr -output=1msy.out input=1msy.pdb
x3dna-dssr output=1msy.out --input=1msy.pdb

#7 the value '1ehz.pdb' overwrites '1msy.pdb'
x3dna-dssr --input=1msy.pdb input=1ehz.pdb

#8-12 with the switch --more set to true
x3dna-dssr -input=1msy.pdb --more
x3dna-dssr -input=1msy.pdb --more=true
x3dna-dssr -input=1msy.pdb --more=yes
x3dna-dssr -input=1msy.pdb --more=on
x3dna-dssr -input=1msy.pdb --more=1

#13 same as without specifying --more,
#      or with values set to false/no/0
x3dna-dssr -input=1msy.pdb --more=off

#14 shorthand forms for --input and --output
x3dna-dssr -i=1msy.pdb -o=1msy.out

#15 it can also be more verbose
x3dna-dssr --input-pdb-file=1msy.pdb

#16-19 within a key, separator dash(-) and underscore(_) 
#      are treated the same, and can be omitted
x3dna-dssr -i=1msy.pdb -non-pair
x3dna-dssr -i=1msy.pdb -non_pair
x3dna-dssr -i=1msy.pdb -nonpair
x3dna-dssr -i=1msy.pdb -nonPAIR
```

By allowing for two/one/zero dashes to precede each key and dash/underscore/no separators within the key, DSSR provides great versatility in specifying command-line options to fit into a user-preferred style. See the supplemental PDF of Lu (2020) for implementation details.

3.2 Default run on 1msy (27-nt RNA), with detailed explanations

The PDB entry 1msy (Correll *et al.*, 2003) is an x-ray crystal structure of a small RNA fragment with 27 nucleotides, solved at 1.41 Å resolution. Figure 3 shows its 3D structure and corresponding 2D diagram. This structure contains a GUAA (GNRA-type) tetraloop mutant of the sarcin/ricin domain from *E. Coli* 23S rRNA, with bulged-G (GpU dinucleotide platform) motif (Correll *et al.*, 2003; Lu *et al.*, 2010). Overall, this structure is simple, but with features that nicely illustrate fundamental aspects of DSSR.

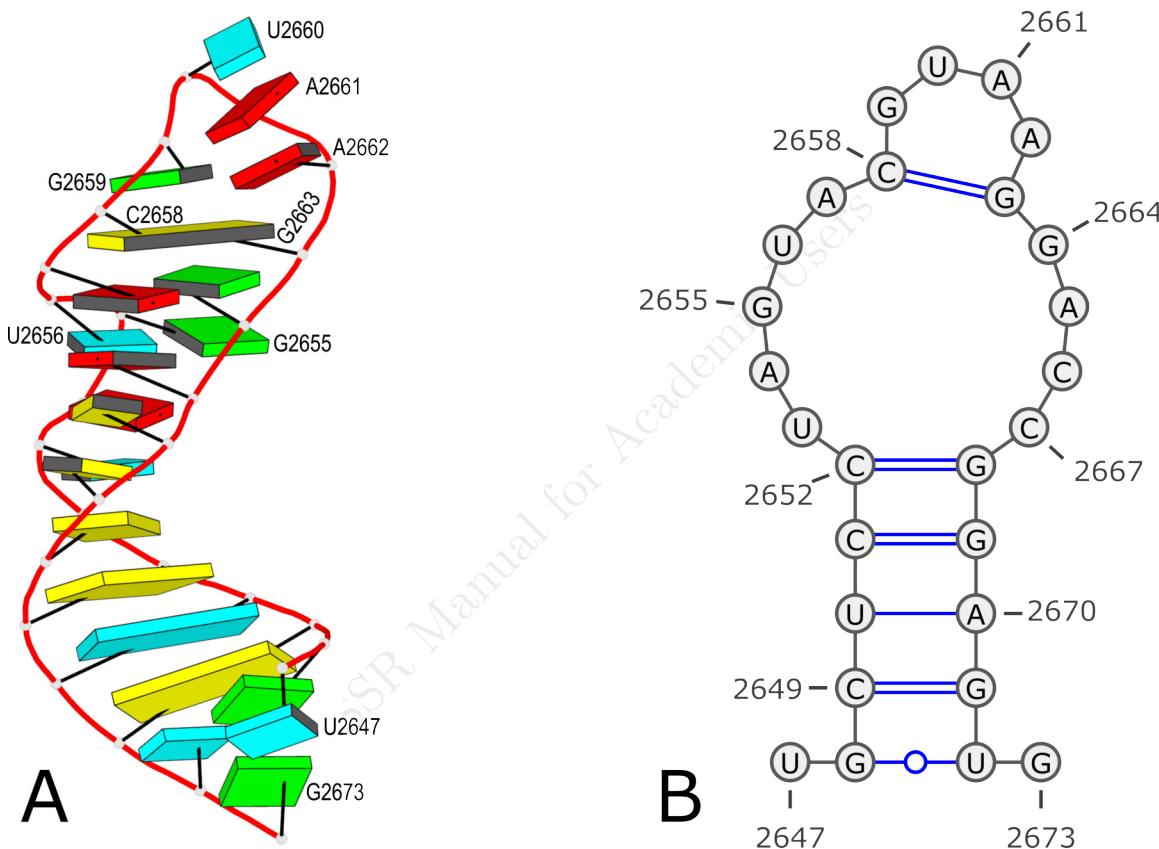


Figure 3: 3D and 2D structures of PDB entry 1msy. (A) 3D schematic auto-created via DSSR-PyMOL integration (Lu, 2020). The labeled residues are referenced later, e.g., the GUAA tetraloop at the top. (B) 2D diagram rendered with VARNA (Darty *et al.*, 2009), using DSSR-derived 2D structural information in .ct format. This figure was annotated using Inkscape (<https://inkscape.org>).

3.2.1 Overview of command runs and outputs

Let the PDB-formatted 3D atomic coordinate file be `1msy.pdb` (or the corresponding mmCIF variant `1msy.cif`), the output file `1msy.out`, and leave all other options in their default settings. Then simply run the commands:

```
x3dna-dssr -i=1msy.pdb -o=1msy.out
x3dna-dssr -i=1msy.cif -o=1msy.out # gives the same results as above
```

It is also possible to pipe (stream) input to `x3dna-dssr`. Here are two examples:

```
cat 1msy.pdb | x3dna-dssr -i=stdin -o=1msy.out
cat 1msy.cif | x3dna-dssr -i=stdin --format=cif -o=1msy.out
# 'stdin' can be replaced by '-' (i.e., -i=-), but '-i=' must be specified
# use '--format=cif' to designate the .cif input format explicitly
```

The screen message (via `stderr`) provides a brief summary of the run, as shown below.

```
total number of nucleotides: 27
total number of base pairs: 13
total number of helices: 1
total number of stems: 1
total number of isolated WC/wobble pairs: 1
total number of multiplets: 1
total number of atom-base capping interactions: 2
total number of splayed-apart dinucleotides: 1
total number of hairpin loops: 1
total number of internal loops: 1
total number of non-loop single-stranded segments: 2

Time used: 00:00:00:00
```

Note that DSSR starts from a `.pdb` or `.cif` file downloaded directly from the PDB. It takes little time to analyze small to mid-size (non-ribosomal) structures.

The 2D diagram (Figure 3B) was rendered with VARNA (Darty *et al.*, 2009), using DSSR-derived 2D structure in connectivity table (`.ct`) format. The `.ct` format is advantageous in that it retains nucleotides numbering information in the input coordinate file. For example, the first U has residue number 2647 (U2647), and the last G is numbered 2673 (G2673). Using dot-bracket notation (`.dbn`) or `.bpseq` format would lead U2647 to be labeled U1 and G2673 as G27, respectively. Here 1 and 27 are the serial numbers of the two terminal bases in the 27-nt long RNA fragment in PDB entry 1msy.

The main output file (`1msy.out`) contains many sections. We will go over them one by one to explain key results and notations used therein. Additionally, DSSR produces the following

list of auxiliary files (named with prefix `dssr-` by default):

```
List of 13 additional files
1 dssr-pairs.pdb -- an ensemble of base pairs
2 dssr-multiplets.pdb -- an ensemble of multiplets
3 dssr-stems.pdb -- an ensemble of stems
4 dssr-helices.pdb -- an ensemble of helices (coaxial stacking)
5 dssr-hairpins.pdb -- an ensemble of hairpin loops
6 dssr-iloops.pdb -- an ensemble of internal loops
7 dssr-2ndstrs.bpseq -- secondary structure in bpseq format
8 dssr-2ndstrs.ct -- secondary structure in connectivity table format
9 dssr-2ndstrs.dbn -- secondary structure in dot-bracket notation
10 dssr-torsions.txt -- backbone torsion angles and suite names
11 dssr-splays.pdb -- an ensemble of splayed-apart units
12 dssr-stacks.pdb -- an ensemble of base stacks
13 dssr-atom2bases.pdb -- an ensemble of atom-base stacking interactions
```

3.2.2 The summary section

Here is the summary section at the top of `1msy.out`:

```
Note: By default, each nucleotide is identified by chainId.name#. So a
common case would be B.A1689, meaning adenosine #1689 on chain B.
One-letter base names for modified nucleotides are put in lower
case (e.g., 'c' for 5MC). For further information about the output
notation, please refer to the DSSR User Manual.
Questions and suggestions are *always* welcome on the 3DNA Forum.
```

```
Command: x3dna-dssr -i=1msy.pdb -o=1msy.out
File name: 1msy.pdb
no. of DNA/RNA chains: 1 [A=27]
no. of nucleotides: 27
no. of atoms: 685
no. of waters: 109
no. of metals: 0
```

The note first explains how a nucleotide (nt) is specified in the output file. Typically, a chain id, residue name, and sequence number are sufficient to identify a nt. For example, A.G19 means guanosine 19 on chain A. In addition to the standard three-letter residue names commonly adopted (as in the PDB), DSSR also uses a one-letter abbreviation symbol. For RNA, the four standard nts in three-letter form—UUA, UUC, UUG and UUU (where U stands for a space)—are shortened to A, C, G, and U, respectively. For DNA, the three-letter (one-letter) nts are UDA (A), UDC (C), UDG (G), and UDT (T), respectively. The one-letter shorthand forms for modified nts, which occur frequently in RNA (e.g., tRNA), are mapped to their canonical counterparts, but put in lower case letters (e.g., c for 5MC). Note that pseudouridine, the most prevalently modified nt in RNA, is denoted P[†] in DSSR and the

[†]Not to be confused with the phosphorus atom in the backbone phosphate group. The distinction should be clear in context.

small case p is reserved for potential modified pseudouridines.

The following sections provide more notations that are used consistently in DSSR output files. It is worth the time and effort to become familiar with them. In practice, users should have little difficulty following the convention, after going over a few examples.

The remaining lines in the summary should be self-explanatory. The ‘Command:’ line provides the running DSSR command with relevant options so that the results can be reproduced. The ‘File name:’ line lists the coordinate file analyzed by DSSR, followed by a list of numbers: DNA/RNA chains (here chain A with 27 nts), total numbers of nts (27), atoms (685), waters (109), and metals (0). In cases with more than one chain or type of metals, the output would be like below (using PDB entry 1jj2 as an example).

```
no. of DNA/RNA chains: 2 [0=2754,9=122]
no. of metals: 210 [Na=86,Mg=117,K=2,Cd=5]
```

Here the first line means that 1jj2 contains two RNA chains: 0 and 9, with 2754 and 122 nts, respectively. The second line indicates the PDB entry includes 210 metal ions: 86 Na, 117 Mg, 2 K, and 5 Cd.

3.2.3 Base pairs

DSSR identifies a total of 13 base pairs (bps), as listed below. They include canonical Watson-Crick (WC) and wobble pairs as well as non-canonical ones (dinucleotide platform, reverse Hoogsteen, sheared G–A, etc). DSSR also produces an auxiliary file named `dssr-pairs.pdb` (Page 17) that contains a MODEL/ENDMDL-delineated ensemble of the identified bps. Here, each bp in the ensemble is oriented in its own reference frame (Olson *et al.*, 2001; Lu and Olson, 2003; Lu *et al.*, 2015), facilitating comparative visualizations.

List of 13 base pairs					
	nt1	nt2	bp	name	Saenger
1	A.U2647	A.G2673	U-G	--	--
2	A.G2648	A.U2672	G-U	Wobble	28-XXVIII
3	A.C2649	A.G2671	C-G	WC	19-XIX
4	A.U2650	A.A2670	U-A	WC	20-XX
5	A.C2651	A.G2669	C-G	WC	19-XIX
6	A.C2652	A.G2668	C-G	WC	19-XIX
7	A.U2653	A.C2667	U-C	--	--
8	A.A2654	A.C2666	A+C	--	--
9	A.G2655	A.U2656	G+U	Platform	--
10	A.U2656	A.A2665	U-A	rHoogsteen	24-XXIV
11	A.A2657	A.G2664	A-G	Sheared	11-XI
12	A.C2658	A.G2663	C-G	WC	19-XIX
13	A.G2659	A.A2662	G-A	Sheared	11-XI

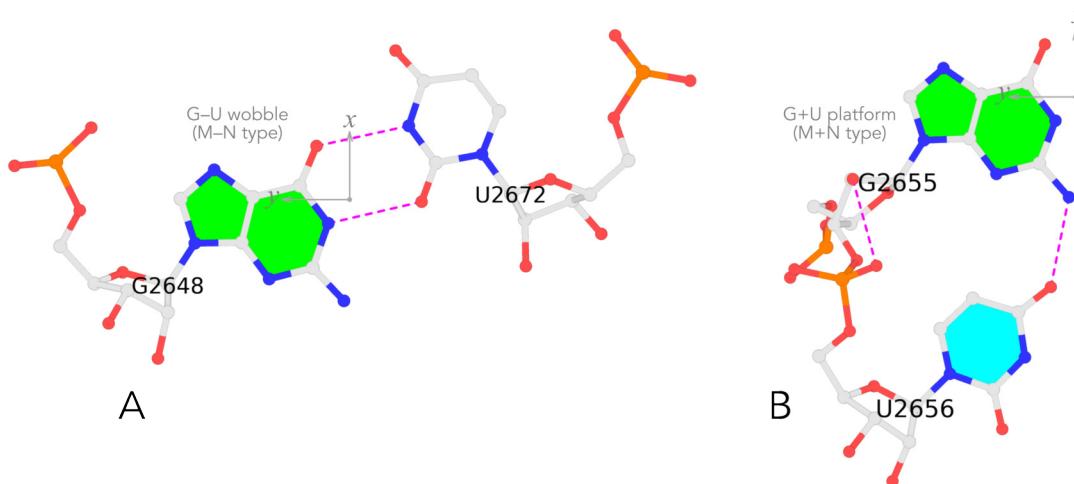


Figure 4: The M–N vs. M+N classification of base pairs (Lu and Olson, 2003; Lu *et al.*, 2015), using the G–U wobble pair and the G+U dinucleotide platform in PDB entry 1msy (Correll *et al.*, 2003) as examples. (A) In the G–U wobble pair, U points in the opposite direction from G. The z-axes of the two bases are anti-parallel, thus their dot product is negative (minus). It is classified as M–N pairing. (B) In the G+U dinucleotide platform, U points in the same direction as G. Their z-axes are parallel, thus their dot product is positive (plus). It is classified as M+N pairing. For easy comparison, the two pairs are oriented in the standard reference frame of the guanine bases. Base rings are filled (green for G and cyan for U) for the positive faces, with z-axis pointing towards the viewer. Note that the M–N type follows the naming convention for WC pairs: A–U/A–T and G–C. This figure was annotated using Inkscape (<https://inkscape.org>).

M+N vs. M–N pairs As the header line shows, each bp is specified by its two constituents (nt1 and nt2 columns, respectively), and an abbreviated type (bp column) of the form $M\pm N$. Here M and N are the one-letter shorthand form of the nts, which can be in upper or lower cases (for modified nts), as noted above. The symbol \pm reflects the two possible relative orientations between bases in a pair: $-$ for canonical WC/wobble bps with opposite faces, and $+$ if one of the two bases is flipped (thus with similar faces). When the $+$ or $-$ relative orientation is irrelevant in context, the M·N notation is used. See below for examples.

Figure 4 provides details on the $M\pm N$ notation that has been used consistently in 3DNA (Lu and Olson, 2003, 2008) and DSSR (Lu *et al.*, 2015) for more than two decades. The $+$ or $-$ relative orientation, combined with the six rigid-body parameters (shear, stretch, stagger, buckle, propeller, and opening; see Figure 5), can unambiguously characterize any pairing geometry, in both DNA and RNA. Among the six parameters, shear, stretch, and opening distinguish different modes of base pairing, whereas stagger, buckle, and propeller

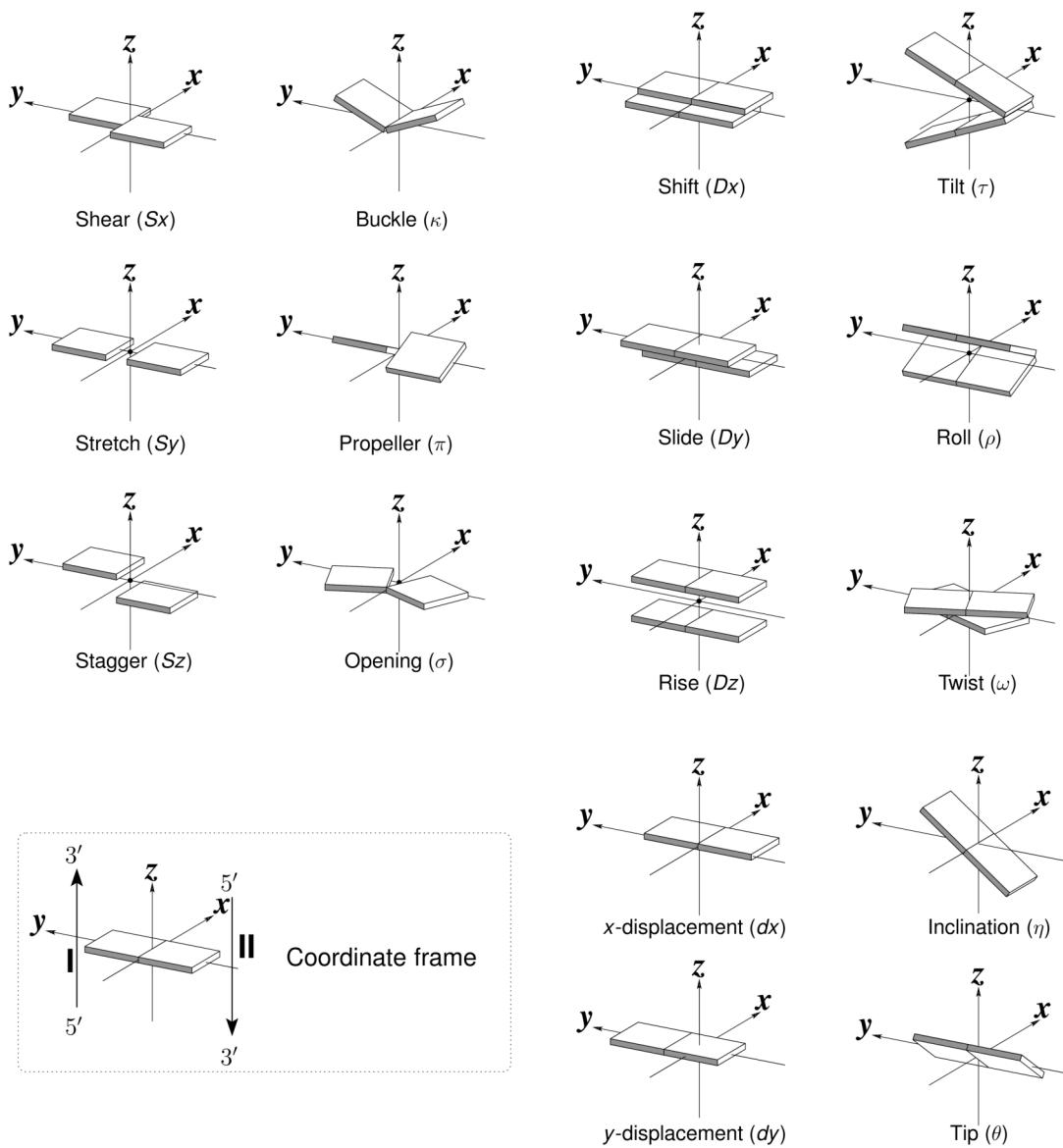


Figure 5: Pictorial definitions of rigid-body parameters used to describe the geometry of base pairs and sequential base-pair steps (Dickerson *et al.*, 1989; Olson *et al.*, 2001). The base-pair reference frame (lower left) is constructed such that the x -axis points away from the (shaded) minor-groove edge of a base or base pair and the y -axis points toward the sequence strand (I). The relative position and orientation of successive base pairs are described with respect to both a dimer reference frame (upper right) and a local helical frame (lower right). Images illustrate positive values of the designated parameters. This figure is taken from Figure 1 of Lu and Olson (2003).

characterize the non-planarity of a base association.

Conversely, given the $M \pm N$ notation with six corresponding bp parameters, 3DNA/DSSR can rigorously reconstruct the spatial disposition of the two interacting bases. This reversibility is one of the unique features of 3DNA/DSSR. However, even with the increasing popularity of 3DNA/DSSR, the value of using $M \pm N$ with six parameters to classify bps has never received the attention it deserves, especially in the RNA structural bioinformatics community.

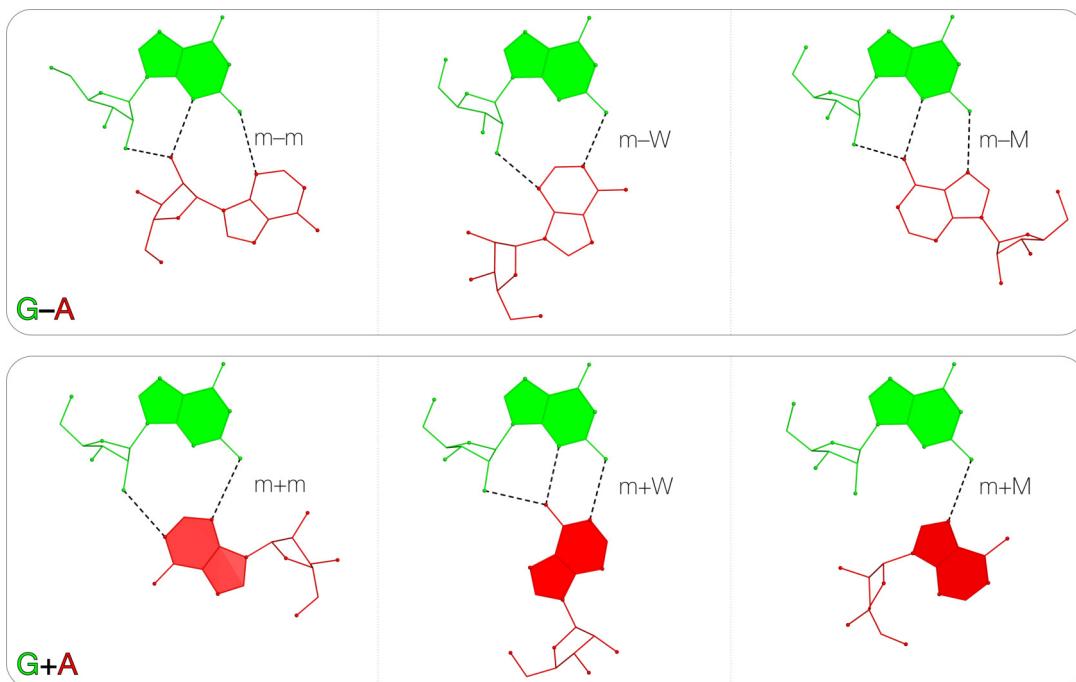


Figure 6: Six types of G·A pairs that involve the minor-groove edge of G. These pairs are automatically uncovered and characterized by DSSR in high-resolution x-ray crystal structures. Top row: three examples in the G–A category with m–m, m–W, and m–M, respectively; Bottom row: three cases in the M+N category with m+m, m+W, and m+M, respectively. This figure was annotated using Inkscape (<https://inkscape.org>).

The strength of the $M \pm N$ approach in 3DNA/DSSR is obvious for rigorous classifications and quantitative comparisons of base pairs. The Olson *et al.* (2019) paper “Effects of non-canonical base pairing on RNA folding: structural context and spatial arrangements of G·A pairs” highlights the advantages enabled by DSSR in uncovering previously unrecognized pairing patterns. Figure 6 shows six types of G·A pairs involving the minor-groove edge of G. They are classified into two groups, three in type G–A and the other three in G+A. The detailed DSSR descriptions facilitate characterization of recurrent geometric motifs and their

structural contexts. The numerical parameters quantify motions of interacting bases and the pathways connecting different spatial forms.

Common names The **name** column gives common names of base pairs, where appropriate. Currently, the list includes WC, reverse WC (rWC), Hoogsteen, reverse Hoogsteen (rHoogsteen), wobble, sheared, imino, Calcutta (Wahl *et al.*, 1996), G+A linker (Olson *et al.*, 2019), and dinucleotide platform (Lu *et al.*, 2010), with unnamed pairs designated --. The PDB entry 1msy contains a GpUpA/GpA miniduplex (Lu *et al.*, 2010), characterized by three named bps (pairs 9 to 11): a platform G+U, a reverse Hoogsteen U–A, and a sheared A–G.

Except for experts in the field, it can be difficult to connect pairing geometries with their familiar names. Can you quickly identify from Figure 6 which one is the well-known sheared G–A pair[†]? Note that the DSSR notation G–A already gives you a hint, limiting the number of possibilities from six to three. With DSSR-assigned names, users can immediately notice common pairs in a given structure. As it turns out, names are handy for dissecting stretches of canonical pairs (e.g., bps 2–6) that form duplex regions, termed ‘stem’ in DSSR (see Page 27).

Saenger classification The **Saenger** column classifies base pairs into 29 types, each with at least two H-bonds between base atoms. While the classic compilation by Saenger (1984) consists of 28 pairs, the DSSR-implementation includes an extra G+C pair added by Burkard *et al.* in Appendix 1 to the 2nd edition of “The RNA World” (Gesteland *et al.*, 1999). Note that in addition to the Roman numerals (I to XXIX) originally used by Saenger, the DSSR output also lists the corresponding Arabic numerals (01 to 29), which may be easier for some users to recognize. If a pair cannot be categorized into one of the 29 known types, the symbol -- is assigned.

Leontis-Westhof (LW) classification The **LW** column presents DSSR implementation of the Leontis and Westhof (2001) classification of base pairs. The LW scheme is based on the three edges of each base with potential for H-bonding interactions (Watson-Crick, Hoogsteen, and Sugar), and the two orientations (*cis* or *trans*) of the interacting bases with respect to the glycosidic bonds. The combinations of edges and orientations ($3 \times 2 \times 2$) “gives rise to 12 basic geometric types with at least two H bonds connecting the bases” (Leontis and Westhof, 2001). This geometry-based approach captures salient features of pairing interactions and strikes a balance between simplicity and expressiveness. The LW scheme is more widely applicable than the Saenger classification, and more intuitive to biologists. As a result, the LW classification

[†]It is the one on the top right of Figure 6, with m–M base edges for interactions.

has become a standard in RNA structural bioinformatics.

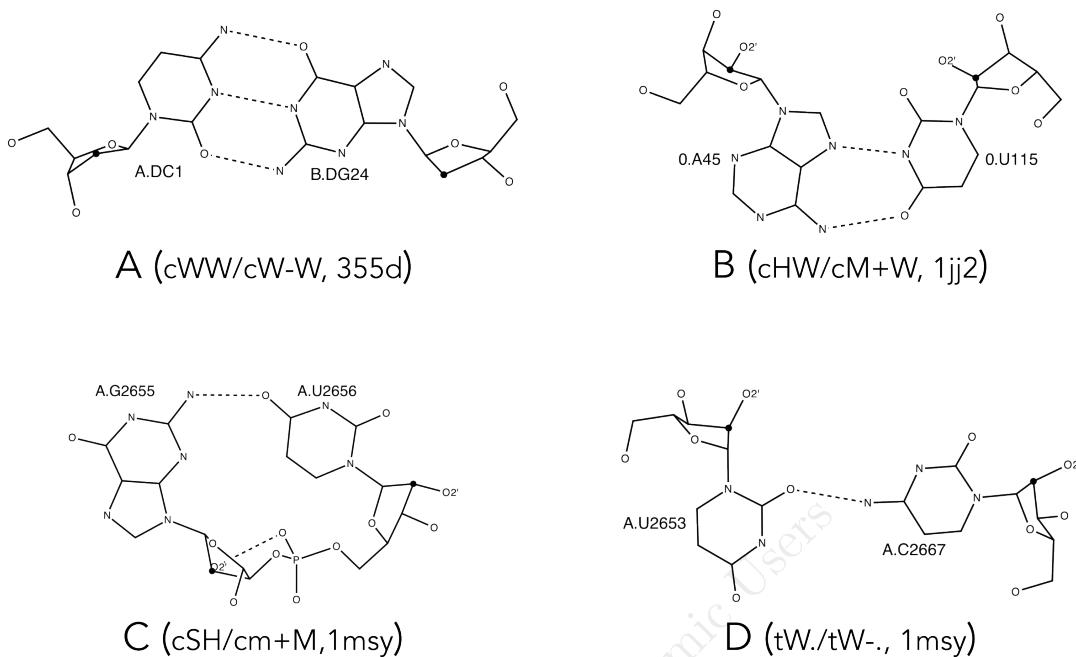


Figure 7: Sample cases where the Leontis-Westhof definition of the three edges does not strictly apply. The results of LW/DSSR classifications are those implemented in the DSSR program. In each image, H-bonds are shown as dashed lines, C_{2'} atoms in black dots, and O_{2'} atoms labeled. (A) A Watson-Crick C–G pair in 355d, a standard B-DNA molecule. The sugar-edge for C or G does not have an O_{2'} atom. (B) The Hoogsteen A+U pair in 1jj2. Here, nucleotide 0.A45 is in the *syn* conformation; thus the O_{2'} atom is pointing away from (instead of towards) the minor-groove edge of the base. (C) The G+U platform (in 1msy) has an out-of-plane ‘backbone edge’. (D) The U–C pair (in 1msy) has only one H-bond at the boundary of WC and Hoogsteen/Major-groove edges. Thus, the interacting edge for A.C2667 cannot be assigned.

The RNA-centric LW classification has inherent limitations (Figure 7). For example, the Sugar edge explicitly includes the 2'-hydroxyl group, rendering it less applicable to DNA structures. Additionally, while the aromatic base can be taken as a rigid body with three fixed edges, the χ (chi) torsion angle characterizes the internal freedom between base and sugar (*anti* vs. *syn*). When χ is in the relatively rare (but not uncommon) *syn* conformation (especially abundant in G-quadruplexes, see Figure 19), the Sugar edge, defined with reference to the common *anti* conformation, seems to no longer exist. The rich variety of RNA pairs extends beyond the 12 basic LW types. There are numerous pairs in RNA with only *one* H-bond or with bifurcated H-bonds, at boundary locations where the LW classification does not

strictly apply. Lemieux and Major (2002) were the first to extend the LW classification. We note the importance of the out-of-plane ‘backbone edge’ formed by an RNA-specific H-bond between O2'(G) and OP2(U) (Lu *et al.*, 2010, see Figure 4B).

DSSR classification The DSSR column presents base-pair classifications introduced by Lu *et al.* (2015). It contains entries such as **cW-W** for canonical pairs and **cm+M** for the G+U dinucleotide platform. Overall, the DSSR scheme follows the pattern of **[ct] [WMm] ± [WMm]** (see Figure 1C):

- **[ct]** stands for *cis/trans*, defined by the relative placement of the two glycosidic bonds with reference to a line connecting the centers of the two bases.
- **[WMm]** stands for the interacting edges defined with reference to an idealized WC pair: W for the Watson-Crick edge, M for the Major-groove edge, m for the minor-groove edge (Figure 1C).
- ± stands for the – vs. + relative base orientations, as noted above (see Page 19 and Figures 4 and 6).

The dot symbol (.) denotes cases where edges or orientations cannot be defined (see pair no. 7 in the list of base pairs on Page 18).

In DSSR, the three interacting edges are strictly base centric; they have nothing to do with the sugar moiety. Only the planar base itself can be, and often is, taken as a rigid body. The terminology of minor and major grooves is widely used in the nucleic-acid community. For example, the well-known A-minor motif (Nissen *et al.*, 2001) is named such because of the m·m interactions between A and a WC pair (see Section 3.4.2). Notably, the standard base reference frame (Olson *et al.*, 2001) has distinct geometric features that enable intuitive assignment of these edges (Figure 1C). Figure 6 shows concrete examples of G–A and G+A pairs in high-resolution crystal structures (Olson *et al.*, 2019); these G·A associations fall naturally into six unique types when the **[WMm]** edges are taken into consideration.

LW vs. DSSR classifications In principle, M (Major groove) in the DSSR classification corresponds to the Hoogsteen-edge (H, LW), and the DSSR m (minor groove) to the Sugar-edge (S, LW) if χ is in the *anti* conformation. In practice, direct DSSR/LW correspondences M/H and m/S are assumed, regardless of *anti/syn* base conformation. Moreover, the *cis/trans* assignment is the same for both notations. Within the **x3dna-dssr** implementation, the LW and DSSR classifications are thus strictly parallel in terms of *cis/trans* orientation and interacting edges.

As noted above, the DSSR scheme has the extra \pm for relative base orientations (see Page 19).

The LW classifications implemented in `x3dna-dssr` may differ from those listed in the RNA 3D Hub website or other resources. These discrepancies normally occur in boundary cases where the assignment of *cis/trans* and interaction edges can be ambiguous. For ‘authentic’ LW classification results, users should consult the original publication of Leontis and Westhof (2001) and use the RNAView (Yang *et al.*, 2003) and FR3D (Sarver *et al.*, 2008) tools instead of DSSR.

3.2.4 Multiplets (higher-order coplanar base associations)

DSSR defines multiplet as three or more bases associated in a coplanar geometry via a network of H-bonding interactions. The program detects one multiplet in 1msy, here a base triplet, with details shown below.

```
List of 1 multiplet
 1 nts=3 GUA A.G2655,A.U2656,A.A2665
```

For each multiplet, DSSR outputs a serial number (here 1), followed by the number of nts (3), the base sequence in one-letter shorthand form (GUA), and then a comma-delimited list of the corresponding nts (A.G2655,A.U2656,A.A2665).

As for base pairs, DSSR also generates a file (named `dssr-multiplets.pdb` by default) that contains a MODEL/ENDMDL-delineated ensemble of identified multiplets. In the PDB NMR-like ensemble file, each multiplet is set to the most extended view with respect to its constituent base rings. The 1msy GUA triplet in this view is shown in Figure 8. Note the extensive network of six H-bonding interactions among the three nts, including the out-of-plane O2'(G) \cdots OP2(U) H-bond. This ‘backbone edge’ H-bond is RNA-specific and contributes to the widespread occurrence of the GpU platform (Lu *et al.*, 2010; Mladek *et al.*, 2012; Ingle *et al.*, 2014).

The abundant A-minor motifs (Nissen *et al.*, 2001; Nguyen *et al.*, 2017) are base triplets, the smallest multiplet. The G-tetrad motif in G-quadruplexes (Burge *et al.*, 2006; Rhodes and Lipps, 2015; Lightfoot *et al.*, 2019), where four Gs are associated via four G+G pairs in a square planar geometry (Figure 19), is another special case of multiplet. In fact, DSSR multiplets are all-encompassing, including pentads, hexads, heptads, octads, etc.

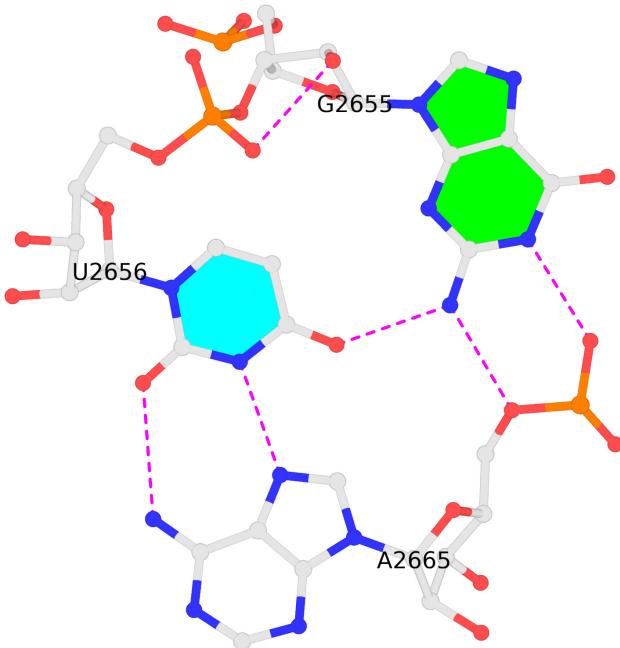


Figure 8: The GUA triplet auto-identified by DSSR in PDB entry 1msy. Here, G2655 and U2656 form a dinucleotide platform (G+U); U2656 and A2665 form a reverse Hoogsteen pair (U–A). The phosphate group of A2665 interacts with the Watson-Crick edge of G2655 with two sequence-specific H-bonds. Note that bases G2655 and U2656 are filled with green and cyan colors, respectively, whilst A2665 is left empty. The contrast between solid colorings and unfilled rings immediately reveals the two different orientations of the bases: G+U–A. See Figures 4 and 6 on M+N vs. M–N pairs. The image was produced via DSSR-PyMOL integration (Lu, 2020).

3.2.5 Helices (stacked pairs of any type or connectivity)

DSSR identifies one helix with 12 base pairs in 1msy, as detailed below. This output section starts with a note explaining the definition of helix, bp type (canonical or otherwise), and the helical form of a dinucleotide step (A-, B-, Z-form, or undefined). In DSSR, helix is defined by base-stacking interactions, *regardless of pairing type and backbone connectivity*. Thus, a helix is composed of at least two stacked pairs, and may contain more than one stem (see below). By referring to Figure 3A, one can immediately see the duplex formed by base-stacking interactions; it starts from the very bottom of the structure and goes all the way up to the sheared G–A pair at the top (part of the GUAA tetraloop).

```

1 List of 1 helix
2   Note: a helix is defined by base-stacking interactions, regardless of bp
3       type and backbone connectivity, and may contain more than one stem.
4   helix#number[stems-contained] bps=number-of-base-pairs in the helix

```

```

5   bp-type: '||' for a canonical WC/wobble pair, '.' otherwise
6   helix-form: classification of a dinucleotide step comprising the bp
7     above the given designation and the bp that follows it. Types
8     include 'A', 'B' or 'Z' for the common A-, B- and Z-form helices,
9     '.' for an unclassified step, and 'x' for a step without a
10    continuous backbone.
11  -----
12  helix#1[1] bps=12
13  strand-1 5'-UGCUCCUAUACG-3'
14  bp-type .|||||.|.
15  strand-2 3'-GUGAGGCCAGGA-5'
16  helix-form ..AAA..x...
17  1 A.U2647      A.G2673      U-G --          --      cWW  cW-W
18  2 A.G2648      A.U2672      G-U Wobble     28-XXVIII cWW  cW-W
19  3 A.C2649      A.G2671      C-G WC         19-XIX   cWW  cW-W
20  4 A.U2650      A.A2670      U-A WC         20-XX    cWW  cW-W
21  5 A.C2651      A.G2669      C-G WC         19-XIX   cWW  cW-W
22  6 A.C2652      A.G2668      C-G WC         19-XIX   cWW  cW-W
23  7 A.U2653      A.C2667      U-C --          --      tW.  tW-.
24  8 A.A2654      A.C2666      A+C --          --      tHH  tM+M
25  9 A.U2656      A.A2665      U-A rHoogsteen 24-XXIV  tWH  tW-M
26  10 A.A2657     A.G2664     A-G Sheared   11-XI    tHS  tM-m
27  11 A.C2658     A.G2663     C-G WC         19-XIX   cWW  cW-W
28  12 A.G2659     A.A2662     G-A Sheared   11-XI    tSH  tm-M

```

Of the 13 base pairs listed on Page 18, only the G+U dinucleotide platform (between G2655 and U2656) is not contained in the helix. U2656 and A2665 form a reverse Hoogsteen pair (U–A, part of the triplet shown in Figure 8) that is included in the helix. Thus, only the so-called bulged G (i.e., G2655) is excluded from the helix.

The helix-form part (line nos. 13–16, starting with **strand-1**, **bp-type**, **strand-2**, and **helix-form**, respectively) gives a quick summary of the stacked pairs: base sequences, bp types (canonical or otherwise), and helical forms of the dinucleotide steps. Needless to say, the A-form helix is the most common type for RNA. However, since DSSR can be equally applied to DNA, the B- and Z-forms are also included in the classification.

3.2.6 Stems (canonical pairs with continuous backbones)

In DSSR, stem is defined as a helix consisting of canonical pairs (WC or wobble) without backbone breaks. These two additional requirements follows the convention widely adopted for RNA secondary structures (Zuker *et al.*, 1999; Mathews *et al.*, 1999; Darty *et al.*, 2009; Lorenz *et al.*, 2011). The PDB entry 1msy contains one stem with five canonical pairs, as shown below (see also Figure 3B).

```

List of 1 stem
Note: a stem is defined as a helix consisting of only canonical WC/wobble
      pairs, with a continuous backbone.
stem#number[#helix-number containing this stem]
Other terms are defined as in the above Helix section.

```

```
-----
stem#1[#1] bps=5
strand-1 5'-GCUCC-3'
bp-type    | || |
strand-2 3'-UGAGG-5'
helix-form .AAA
1 A.G2648      A.U2672      G-U Wobble      28-XXVIII cWW  cW-W
2 A.C2649      A.G2671      C-G WC          19-XIX   cWW  cW-W
3 A.U2650      A.A2670      U-A WC          20-XX    cWW  cW-W
4 A.C2651      A.G2669      C-G WC          19-XIX   cWW  cW-W
5 A.C2652      A.G2668      C-G WC          19-XIX   cWW  cW-W
```

The RNA literature is filled with terms for double helix (helix, stem, arm, paired region, etc.), often used interchangeably and without explicit definition. DSSR employs only the two most commonly used terms, helix and stem, and makes a clear distinction between them. Notice how pair names and the `helix-form` portion in Section 3.2.5 facilitate a quick visual identification of a stem within the helix.

3.2.7 Isolated canonical pairs

If a canonical pair does not belong to any stem, which by definition consists of two pairs, it is an isolated one. In 1msy, C2658 and G2663 form such an isolated WC pair (C–G), as detailed below.

```
List of 1 isolated WC/wobble pair
Note: isolated WC/wobble pairs are assigned negative indices to
differentiate them from the stem numbers, which are positive.
-----
[#1] -1 A.C2658      A.G2663      C-G WC          19-XIX   cWW  cW-W
```

The [#1] at the beginning means this isolated canonical pair is part of helix #1 (see Section 3.2.5). Note also the negative indices (-1) for isolated canonical pairs vs. positive values for stems. The significance of the distinction will become obvious as loops are specified.

3.2.8 Base stacks

Base stack is defined as an ordered list of nucleobases assembled together via stacking interactions, regardless of backbone connectivity or pairing status. By default, stacking interactions within stems are excluded from the listing. DSSR identifies five base stacks in 1msy, as detailed below. The UAA stack (U2660-A2661-A2662, no. 2 in the listing) stands out at the top of Figure 3A, as part of the GUAA tetraloop. Nucleotides not involved in base-stacking interactions, if any, are also listed (as for PDB entry 1ehz, see below).

```
List of 5 base stacks
Note: a stack is an ordered list of nucleotides assembled together via
      base-stacking interactions, regardless of backbone connectivity.
      Stacking interactions within a stem are *not* included.
1 nts=2 GG A.G2648,A.G2673
2 nts=3 UAA A.U2660,A.A2661,A.A2662
3 nts=4 CUAU A.C2652,A.U2653,A.A2654,A.U2656
4 nts=4 GGGG A.G2655,A.G2664,A.G2663,A.G2659
5 nts=6 CAACCG A.C2658,A.A2657,A.A2665,A.C2666,A.C2667,A.G2668
```

3.2.9 Atom-base capping interactions

The backbone phosphate group (more specifically, its exocyclic OP2 atom) can stack on a base to cap a helix, as first observed by Quigley and Rich (1976) in the tRNA^{Phe}. Such atom-base capping interactions are also observed in other structural motifs, including GNRA and UNCG tetraloops (D'Ascenzo *et al.*, 2017). DSSR can automatically identify and characterize these interactions. Moreover, the stacking entities have been generalized to include solvent water and the sugar moiety. An auxiliary file named `dssr-atom2bases.pdb` (Page 17) is also auto-created, with an ensemble of identified cases for easy visualization. Two atom-base capping interactions are identified in 1msy, as shown below.

```
List of 2 atom-base capping interactions
dv: vertical distance of the atom above the nucleotide base
-----
type      atom          nt        dv
1 phosphate  OP2@A.A2661    A.G2659   3.04
2 sugar      O4 '@A.G2664    A.G2663   3.48
```

Here the **type** column is the stacking entity: phosphate, sugar, water, or other. The **atom** and **nt** columns have obvious meanings. The **dv** column presents the vertical distance (in Å) of the stacking atom above the nucleobase. Thus, the first example shows that the OP2 atom of the A2661 phosphate group stacks at 3.04 Å over G2659.

3.2.10 Various loops

Commonly occurring RNA loops (hairpins, bulges, internal loops, junctions) are illustrated in Figure 9. DSSR identifies them all and distinguishes symmetric vs. asymmetric internal loops. In PDB entry 1msy, DSSR identifies one hairpin loop and an asymmetric internal loop, as detailed below.

```
Note: for the various types of loops listed below, numbers within the first
      set of brackets are the number of loop nts, and numbers in the second
      set of brackets are the identities of the stems (positive number) or
      isolated WC/wobble pairs (negative numbers) to which they are linked.
```

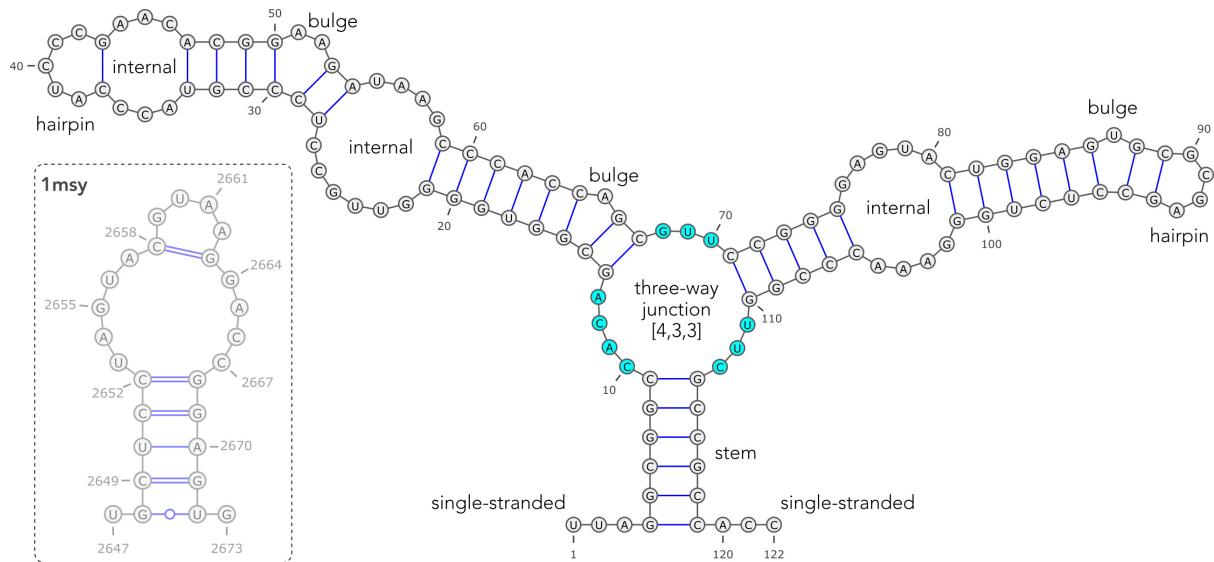


Figure 9: Common RNA secondary structural elements. ‘Closed’ loops of various types (hairpins, bulges, internal loops, and junctions), single-stranded fragments at the 5’ and 3’ ends, and one double-helical stem are labeled. The three-way junction loop is highlighted, and annotated per DSSR (Lu *et al.*, 2015). The 2D diagram is based on the 5S rRNA in PDB entry 1jj2 (Klein *et al.*, 2001), analyzed by DSSR, and rendered via VARNA (Darty *et al.*, 2009). For easy reference, the 2D diagram of 1msy (from Figure 3B) is inserted at the lower left, framed in a dashed box. This figure was annotated using Inkscape (<https://inkscape.org>).

```
*****
List of 1 hairpin loop
1 hairpin loop: nts=6; [4]; linked by [#-1]
summary: [1] 4 [A.2658 A.2663] 1
nts=6 CGUAAG A.C2658 ,A.G2659 ,A.U2660 ,A.A2661 ,A.A2662 ,A.G2663
nts=4 GUAA A.G2659 ,A.U2660 ,A.A2661 ,A.A2662
*****
List of 1 internal loop
1 asymmetric internal loop: nts=13; [5,4]; linked by [#1,#-1]
summary: [2] 5 4 [A.2652 A.2668 A.2658 A.2663] 5 1
nts=13 CUAGUACGGACCG A.C2652 ,A.U2653 ,A.A2654 ,A.G2655 ,A.U2656 ,A.A2657 ,A.C2658 ,A.G2663 ,A.
    ↢ G2664 ,A.A2665 ,A.C2666 ,A.C2667 ,A.G2668
nts=5 UAGUA A.U2653 ,A.A2654 ,A.G2655 ,A.U2656 ,A.A2657
nts=4 GACC A.G2664 ,A.A2665 ,A.C2666 ,A.C2667
```

Users should easily understand the results presented in this section by referring to the note at the top of the section and the 2D diagram (see Figure 3 and also the insert in Figure 9). Nevertheless, the two sets of brackets are worth further explanation. The hairpin loop contains a total of 6 nts, including the closing C–G pair: [4] means 4 nts, i.e., a tetraloop (excluding the 2 nts in the closing pair); [#−1] indicates that the tetraloop is closed by the first isolated

canonical pair (see Section 3.2.7). The asymmetric internal loop contains 13 nts in total: [5,4] means 5 nts along one strand, and 4 nts on the other; [#1,#-1] indicates that the internal loop is delineated by the first stem and the first isolated pair; thus, $5 + 4 + 2 * 2 = 13$ nucleotides in the loop.

In the **summary** line, the first bracket contains the number of stems (including isolated canonical pairs) delineating the loop: [1] for hairpin loops, [2] for bulges and internal loops, and [3] or above for junctions loops. The following set of numbers corresponds to the bridging nucleotides (e.g., 5 4 for the internal loop in 1msy). The second bracket lists the closing pairs in **chainId.residueNumber** format (e.g., [A.2652 A.2668 A.2658 A.2663] for the internal loop in 1msy). The following set of numbers corresponds to the lengths of the stems delineating the loop (e.g., 5 1 for the internal loop in 1msy where length 1 refers to the isolated canonical pair). Finally, a textual description of the loop may be available: e.g., the type of k-turns (see Section 3.4.4) associated with an internal loop. In fact, the **summary** line was introduced and formatted this way to cater to Bayrak *et al.* (2017) for their survey of k-turn motifs in the PDB.

3.2.11 Single-stranded fragments

In DSSR, single-stranded fragment is defined as a consecutive stretch of nts not involved in stems, isolated canonical pairs, or various loops. DSSR identifies two cases in 1msy (Figure 3B), as detailed below.

```
List of 2 non-loop single-stranded segments
 1 nts=1 U A.U2647
 2 nts=1 G A.G2673
```

From Figure 3A, it seems as if DSSR made the wrong assignments here. However, these two occurrences represent special, boundary cases: each fragment consists of only one nt, and the two nts (U2647 and G2673) form a pair that is part of a helix (Section 3.2.5). Since the U2647–G2673 pair is too distorted to be classified as a wobble pair, it does not belong to a stem (Section 3.2.6 and Figure 3B). Thus, U2647 and G2673 fulfill the above definition of single-stranded fragments in DSSR default settings. The situation is clear cut for the 4-nt single-stranded fragment ACCA (A73,C74,C75,A76) in tRNA 1ehz (see below).

3.2.12 2D structure in dot-bracket notation (.dbn) and .ct format

From 3D atomic coordinates, DSSR automatically derives a 2D structure expressed in an extended dot-bracket notation (dbn): pseudoknots are represented by matched pairs of [], {}, <>, etc. and chain breaks are marked by &. The dbn output file (named `dssr-2ndstrs.dbn` by default) can be directly fed into VARNA (Darty *et al.*, 2009) for interactive visualization and creation of 2D diagrams (see Figures 3 and 9).

The dbn is written in a FASTA-like format with three lines: the title line (with > on the first column) is followed by base sequence and the 2D structure, each on a separate line. The dbn for 1msy is as follows.

```
Secondary structures in dot-bracket notation (dbn) as a whole and per chain
>1msy nts=27 [whole]
UGCUCCUAGUACGUAGGACCGGAGUG
.((((.....(....)....)))..
>1msy-A #1 nts=27 0.30(2.47) [chain] RNA
UGCUCCUAGUACGUAGGACCGGAGUG
.((((.....(....)....)))..
```

Since 1msy contains only a single chain (A), the dbn contents per chain and for the whole structure are the same. For a DNA duplex structure, such as PDB entry 355d, the differences are obvious (see below). Here the dbn for the whole DNA structure contains matched parentheses, with the two chains explicitly separated by &. On the other hand, chain A only has fore-facing parentheses whilst chain B only has back-facing parentheses. The dbn notation, per chain, is thus invalid (i.e., incomplete) in this case. More generally, whenever canonical pairs are formed across chains, the dbn per chain would be invalid. For strictly valid dbn per chain, each must be analyzed separately.

```
Secondary structures in dot-bracket notation (dbn) as a whole and per chain
>355d nts=24 [whole]
CGCGAATTCTCGCG&CGCGAATTCTCGCG
((((((((((&))))))))))
>355d-A #1 nts=12 3.21(0.51) [chain] DNA
CGCGAATTCTCGCG
((((((((((
>355d-B #2 nts=12 3.37(0.50) [chain] DNA
CGCGAATTCTCGCG
))))))))))
```

For better connection to the 2D community, DSSR also generates three auxiliary files (see Page 17)—`dssr-2ndstrs.dbn`, `dssr-2ndstrs.bpseq`, and `dssr-2ndstrs.ct`—that contain the 2D structure for the whole entry in dbn, the simple `.bpseq` format, and the connectivity

table (.ct) format, respectively. Compared to dbn and .bpseq, the .ct format is more expressive since it retains sequence numbers per the atomic coordinates file (see Figures 3 and 9). The .ct output file for 1msy is listed below.

```

27 ENERGY = 0.0 [1msy] -- secondary structure derived by DSSR
 1 U      0      2      0  2647 # name=A.U2647
 2 G      1      3     26  2648 # name=A.G2648, pairedNt=A.U2672
 3 C      2      4     25  2649 # name=A.C2649, pairedNt=A.G2671
 4 U      3      5     24  2650 # name=A.U2650, pairedNt=A.A2670
 5 C      4      6     23  2651 # name=A.C2651, pairedNt=A.G2669
 6 C      5      7     22  2652 # name=A.C2652, pairedNt=A.G2668
 7 U      6      8      0  2653 # name=A.U2653
 8 A      7      9      0  2654 # name=A.A2654
 9 G      8     10      0  2655 # name=A.G2655
10 U     9     11      0  2656 # name=A.U2656
11 A    10     12      0  2657 # name=A.A2657
12 C    11     13     17  2658 # name=A.C2658, pairedNt=A.G2663
13 G    12     14      0  2659 # name=A.G2659
14 U    13     15      0  2660 # name=A.U2660
15 A    14     16      0  2661 # name=A.A2661
16 A    15     17      0  2662 # name=A.A2662
17 G    16     18     12  2663 # name=A.G2663, pairedNt=A.C2658
18 G    17     19      0  2664 # name=A.G2664
19 A    18     20      0  2665 # name=A.A2665
20 C    19     21      0  2666 # name=A.C2666
21 C    20     22      0  2667 # name=A.C2667
22 G    21     23      6  2668 # name=A.G2668, pairedNt=A.C2652
23 G    22     24      5  2669 # name=A.G2669, pairedNt=A.C2651
24 A    23     25      4  2670 # name=A.A2670, pairedNt=A.U2650
25 G    24     26      3  2671 # name=A.G2671, pairedNt=A.C2649
26 U    25     27      2  2672 # name=A.U2672, pairedNt=A.G2648
27 G    26      0      0  2673 # name=A.G2673

```

3.2.13 Summary of structural features per nucleotide

This section summarizes structural features that each nt possesses or is a part of. For each identified nt, the output includes one-letter shorthand symbol, dbn, residue identifier, the rmsd of least-squares fitting between the base and its standard counterpart, and a comma-separated list of features: *anti* or *syn* (χ) base orientation, approximate C2' or C3'-endo sugar puckering, BI or BII backbone conformation, modified or not, the 2D structural elements (canonical or non-canonical pair, multiplet, helix, stem, various loops, etc).

```

Summary of structural features of 27 nucleotides
Note: the first five columns are: (1) serial number, (2) one-letter
shorthand name, (3) dbn, (4) id string, (5) rmsd (~zero) of base
ring atoms fitted against those in a standard base reference
frame. The sixth (last) column contains a comma-separated list of
features whose meanings are mostly self-explanatory, except for:
  turn: angle C1'(i-1)--C1'(i)--C1'(i+1) < 90 degrees
  break: no backbone linkage between O3'(i-1) and P(i)
1  U . A.U2647   0.011  anti,~C3'-endo,non-canonical,non-pair-contact,helix-end,ss-non-loop
2  G ( A.G2648   0.012  anti,~C3'-endo,BI,canonical,non-pair-contact,helix,stem-end
3  C ( A.C2649   0.019  anti,~C3'-endo,BI,canonical,non-pair-contact,helix,stem

```

```

4   U ( A.U2650  0.019  anti,-C3'-endo,BI,canonical,non-pair-contact,helix,stem
5   C ( A.C2651  0.024  anti,-C3'-endo,BI,canonical,non-pair-contact,helix,stem
6   C ( A.C2652  0.032  anti,-C3'-endo,BI,canonical,non-pair-contact,helix,stem-end,internal-loop
7   U . A.U2653  0.019  anti,-C3'-endo,non-canonical,non-pair-contact,helix,internal-loop,phosphate
8   A . A.A2654  0.019  anti,-C2'-endo,BII,non-canonical,non-pair-contact,helix,internal-loop
9   G . A.G2655  0.022  turn,anti,-C2'-endo,non-canonical,non-pair-contact,multiplet,internal-loop
10  U . A.U2656  0.020  anti,-C3'-endo,BI,non-canonical,non-pair-contact,helix,multiplet,internal-loop
    ↵ ,phosphate
11  A . A.A2657  0.023  anti,-C3'-endo,BI,non-canonical,non-pair-contact,helix,internal-loop
12  C ( A.C2658  0.013  anti,-C3'-endo,BI,isolated-canonical,non-pair-contact,helix,hairpin-loop,
    ↵ internal-loop
13  G . A.G2659  0.033  u-turn,anti,-C3'-endo,BI,non-canonical,non-pair-contact,helix-end,hairpin-loop
    ↵ ,cap-acceptor,splayed-apart
14  U . A.U2660  0.020  turn,u-turn,anti,-C3'-endo,non-pair-contact,hairpin-loop,splayed-apart
15  A . A.A2661  0.015  u-turn,anti,-C3'-endo,BI,non-pair-contact,hairpin-loop,cap-donor,phosphate
16  A . A.A2662  0.010  u-turn,anti,-C3'-endo,BI,non-canonical,non-pair-contact,helix-end,hairpin-loop
    ↵ ,phosphate
17  G ) A.G2663  0.019  anti,-C3'-endo,BI,isolated-canonical,non-pair-contact,helix,hairpin-loop,
    ↵ internal-loop,cap-acceptor
18  G . A.G2664  0.014  anti,-C3'-endo,BI,non-canonical,non-pair-contact,helix,internal-loop,cap-donor
19  A . A.A2665  0.014  anti,-C3'-endo,BI,non-canonical,non-pair-contact,helix,multiplet,internal-loop
    ↵ ,phosphate
20  C . A.C2666  0.016  anti,-C3'-endo,BI,non-canonical,non-pair-contact,helix,internal-loop,phosphate
21  C . A.C2667  0.029  anti,-C3'-endo,BI,non-canonical,non-pair-contact,helix,internal-loop
22  G ) A.G2668  0.012  anti,-C3'-endo,BI,canonical,non-pair-contact,helix,stem-end,internal-loop
23  G ) A.G2669  0.020  anti,-C3'-endo,BI,canonical,non-pair-contact,helix,stem
24  A ) A.A2670  0.019  anti,-C3'-endo,BI,canonical,non-pair-contact,helix,stem
25  G ) A.G2671  0.023  anti,-C3'-endo,BI,canonical,non-pair-contact,helix,stem
26  U ) A.U2672  0.024  anti,-C3'-endo,BI,canonical,non-pair-contact,helix,stem-end
27  G . A.G2673  0.010  anti,-C3'-endo,non-canonical,non-pair-contact,helix-end,ss-non-loop

```

3.2.14 Backbone conformations

The auto-generated output file `dssr-torsions.txt` contains commonly used backbone parameters as summarized below. They include torsion angles, suite names (Richardson *et al.*, 2008), etc.

Main chain conformational parameters The single-stranded Zp (ssZp, Li *et al.*, 2019) is an extension of the Zp parameter in 3DNA to distinguish different types of duplexes (Lu *et al.*, 2000). The ssZp parameter was inspired by the work of Chen *et al.* (2010). They observed that sugar pucker correlates with the perpendicular distance (Dp) from the 3'-phosphate to the glycosidic bond: if Dp > 2.9 Å, the sugar C3'-endo; otherwise C2'-endo. The splayed-apart conformation of dinucleotides is quantified by an innovative angle parameter in DSSR (see Section 3.3.7). Additionally, backbones are classified into BI/BII forms, χ into *anti/syn* conformations, etc. The collective information (see below) allows for quick identification of interesting backbone features.

```

alpha:  O3'(i-1)-P-05'-C5'
beta:  P-05'-C5'-C4'

```

```

gamma: O5'-C5'-C4'-C3'
delta: C5'-C4'-C3'-O3'
epsilon: C4'-C3'-O3'-P(i+1)
zeta: C3'-O3'-P(i+1)-O5'(i+1)
e-z: epsilon-zeta (BI/BII backbone classification)

chi for pyrimidines(Y): O4'-C1'-N1-C2; purines(R): O4'-C1'-N9-C4
Range [170, -50(310)] is assigned to anti, and [50, 90] to syn

phase-angle: the phase angle of pseudorotation and puckering
sugar-type: ~C2'-endo for C2'-endo like conformation, or
            ~C3'-endo for C3'-endo like conformation
            Note the ONE column offset (for easy visual distinction)

ssZp: single-stranded Zp, defined as the z-coordinate of the 3' phosphorus atom
      (P) expressed in the standard reference frame of the 5' base; the value is
      POSITIVE when P lies on the +z-axis side (base in anti conformation);
      NEGATIVE if P is on the -z-axis side (base in syn conformation)
Dp: perpendicular distance of the 3' P atom to the glycosidic bond
[Ref: Chen et al. (2010): "MolProbity: all-atom structure
validation for macromolecular crystallography."
Acta Crystallogr D Biol Crystallogr, 66(1):12-21]
splay: angle between the bridging P to the two base-origins of a dinucleotide.

```

There are too many parameters to fit into the page width. The following shows a selected portion of nucleotides in 1msy, with parameters split into two sections.

	nt	alpha	beta	gamma	delta	epsilon	zeta	e-z
5	C A.C2651	-62.3	173.8	46.2	84.3	-154.6	-71.3	-83(BI)
6	C A.C2652	-67.5	175.2	58.1	78.0	-154.0	-67.0	-87(BI)
7	U A.U2653	-61.8	167.4	55.9	82.4	-149.4	55.3	155(..)
8	A A.A2654	165.2	133.8	56.7	149.4	-98.3	161.4	100(BII)
9	G A.G2655	-96.2	91.7	179.3	149.3	-167.2	141.4	51(..)
10	U A.U2656	-72.5	157.8	37.9	91.6	-141.0	-65.6	-75(BI)
	nt	chi	phase-angle	sugar-type	ssZp	Dp	splay	
5	C A.C2651	-159.0(anti)	12.9(C3'-endo)	~C3'-endo	4.29	4.58	24.06	
6	C A.C2652	-158.8(anti)	15.5(C3'-endo)	~C3'-endo	4.28	4.53	22.84	
7	U A.U2653	-151.0(anti)	15.7(C3'-endo)	~C3'-endo	4.00	4.62	22.57	
8	A A.A2654	-145.3(anti)	151.0(C2'-endo)	~C2'-endo	0.91	0.92	43.08	
9	G A.G2655	-93.5(anti)	151.5(C2'-endo)	~C2'-endo	2.15	2.14	54.21	
10	U A.U2656	-173.9(anti)	0.4(C3'-endo)	~C3'-endo	4.35	4.42	27.39	

Virtual torsion angles Three sets of virtual torsion angles are calculated by DSSR. The first set is η/θ , pioneered by Olson (1980), that uses the phosphorus (P) and C4' atoms to simplify the sugar-phosphate backbone. The second is η'/θ' , introduced by Keating *et al.* (2011), using the C1' instead of C4' atoms. The C1' atom is covalently linked to the base, and its location is more reliably determined in x-ray crystal structures. The base ‘origin’ (Olson *et al.*, 2001), defined by a least-squares fitting procedure using the whole base, should be even more reliably located. Thus, we defined the third set η''/θ'' (Li *et al.*, 2019), using base origins in place of C4' or C1' atoms.

```

eta:    C4'(i-1)-P(i)-C4'(i)-P(i+1)
theta:  P(i)-C4'(i)-P(i+1)-C4'(i+1)
[Ref: Olson (1980): "Configurational statistics of polynucleotide chains.
An updated virtual bond model to treat effects of base stacking."
Macromolecules, 13(3):721-728]

eta':   C1'(i-1)-P(i)-C1'(i)-P(i+1)
theta': P(i)-C1'(i)-P(i+1)-C1'(i+1)
[Ref: Keating et al. (2011): "A new way to see RNA." Quarterly Reviews
of Biophysics, 44(4):433-466]

eta":   base(i-1)-P(i)-base(i)-P(i+1)
theta": P(i)-base(i)-P(i+1)-base(i+1)

      nt          eta     theta     eta'     theta'     eta"     theta"
1    U A.U2647    ---     ---     ---     ---     ---     ---
2    G A.G2648    172.1   -133.3   -163.5   -131.2   -97.6   -93.1
3    C A.C2649    162.7   -140.1   -178.0   -138.9   -120.5   -104.8
4    U A.U2650    167.0   -148.3   -174.5   -150.1   -122.6   -127.9
5    C A.C2651    165.4   -147.2   -177.6   -145.6   -144.3   -123.9
6    C A.C2652    171.3   -142.0   -176.3   -139.3   -142.0   -91.5
7    U A.U2653    172.2   -18.3    -170.0   -63.3    -107.9   -99.1
8    A A.A2654    46.3    172.0    120.5    135.6    126.4    150.3
9    G A.G2655   -44.2    24.9    -82.3    59.0    -98.6    91.3
10   U A.U2656   170.9   -121.7   163.1   -122.7   161.9   -94.8
11   A A.A2657   162.1   -127.4   -177.7   -123.2   -126.9   -67.0
12   C A.C2658   159.4   -135.3   -176.0   -135.4   -98.2   -106.5
13   G A.G2659   167.6   -117.7   -179.4   -160.6   -134.1   156.9
14   U A.U2660    15.1   -126.1    43.5   -124.9    21.0   -65.2
15   A A.A2661   160.4   -132.0   -169.4   -138.7   -95.8   -93.3
16   A A.A2662   167.0   -83.0    -174.8   -81.7   -117.0   -64.2
17   G A.G2663   172.6   -154.0   -148.2   -148.6   -101.2   -97.0
18   G A.G2664   166.2   -166.9   -168.9   -147.4   -95.5   137.9
19   A A.A2665   -155.6   141.6    175.0   164.3   154.7   -178.6
20   C A.C2666   -178.4   -125.3   -169.0   -123.0   -153.9   -74.8
21   C A.C2667   164.6   -120.7   -172.9   -116.0   -101.6   -71.6
22   G A.G2668   164.9   -150.0   -168.4   -145.8   -97.4   -120.2
23   G A.G2669   171.3   -139.8   -172.8   -139.5   -131.7   -111.7
24   A A.A2670   170.6   -153.2   -173.8   -152.7   -127.2   -132.0
25   G A.G2671   170.4   -134.4   -180.0   -133.1   -147.1   -94.7
26   U A.U2672   172.2   -167.9   -166.6   -170.6   -110.6   -171.6
27   G A.G2673    ---     ---     ---     ---     ---     ---

```

Sugar conformational parameters By default, sugar pucker analysis in DSSR follows the work of Altona and Sundaralingam (1972). The phase angle (in the range of 0° to 360°) of pseudorotation is equally divided into ten regions. The two most frequent sugar puckers are the C3'-endo (0°, 36°), as in canonical RNA and A-form DNA, and the C2'-endo (144°, 180°) as in standard B-form DNA. Where appropriate, sugar pucker is assigned into either ~C3'-endo or ~C2'-endo (see above in Section 3.2.14 under ‘sugar-type’).

```

v0: C4'-O4'-C1'-C2'
v1: O4'-C1'-C2'-C3'
v2: C1'-C2'-C3'-C4'
v3: C2'-C3'-C4'-O4'
v4: C3'-C4'-O4'-C1'

```

tm: the amplitude of pucker
 P: the phase angle of pseudorotation
 [Ref: Altona & Sundaralingam (1972): "Conformational analysis of the sugar ring in nucleosides and nucleotides. A new description using the concept of pseudorotation."
 J Am Chem Soc, 94(23):8205-8212]

	nt	v0	v1	v2	v3	v4	tm	P	Puckering
1	U A.U2647	7.5	-34.5	46.2	-43.8	23.1	46.9	9.9	C3'-endo
2	G A.G2648	9.5	-31.5	39.4	-35.5	16.8	39.6	5.3	C3'-endo
3	C A.C2649	4.0	-28.3	39.9	-38.4	21.9	40.9	12.9	C3'-endo
4	U A.U2650	-2.4	-25.5	41.9	-44.0	29.4	44.9	21.3	C3'-endo
5	C A.C2651	4.8	-32.5	45.7	-44.6	25.0	46.9	12.9	C3'-endo
6	C A.C2652	2.9	-29.7	43.8	-44.0	25.9	45.4	15.5	C3'-endo
7	U A.U2653	1.6	-28.2	42.3	-41.3	25.2	44.0	15.7	C3'-endo
8	A A.A2654	-33.2	44.3	-38.7	20.1	8.5	44.3	151.0	C2'-endo
9	G A.G2655	-37.3	50.1	-43.9	22.9	8.9	50.0	151.5	C2'-endo
10	U A.U2656	12.7	-32.9	39.6	-33.2	13.3	39.6	0.4	C3'-endo
11	A A.A2657	-6.4	-21.7	39.9	-44.5	32.0	44.6	26.5	C3'-endo
12	C A.C2658	-0.0	-28.5	44.6	-44.4	28.4	46.9	17.9	C3'-endo
13	G A.G2659	-9.4	-20.6	40.1	-45.3	35.5	46.1	29.4	C3'-endo
14	U A.U2660	-8.9	-14.6	31.3	-37.4	29.0	37.0	32.3	C3'-endo
15	A A.A2661	7.2	-28.7	38.3	-35.0	17.5	38.6	8.0	C3'-endo
16	A A.A2662	0.0	-21.1	32.9	-33.9	21.4	34.8	18.7	C3'-endo
17	G A.G2663	-7.9	-22.7	42.3	-47.3	35.7	47.8	27.7	C3'-endo
18	G A.G2664	7.4	-33.6	45.9	-43.1	22.1	46.5	9.7	C3'-endo
19	A A.A2665	11.2	-35.6	45.2	-39.6	18.1	45.4	4.5	C3'-endo
20	C A.C2666	2.0	-27.6	41.2	-41.2	24.6	42.8	16.0	C3'-endo
21	C A.C2667	1.9	-30.7	45.7	-46.0	28.2	47.6	16.5	C3'-endo
22	G A.G2668	9.1	-34.4	44.4	-40.9	20.0	44.7	7.3	C3'-endo
23	G A.G2669	4.5	-31.3	45.1	-43.5	24.4	46.3	13.0	C3'-endo
24	A A.A2670	5.3	-32.9	46.7	-45.3	25.1	47.9	12.6	C3'-endo
25	G A.G2671	7.6	-33.2	44.6	-41.4	21.7	45.2	9.2	C3'-endo
26	U A.U2672	-0.3	-23.0	35.9	-37.6	23.6	38.0	19.2	C3'-endo
27	G A.G2673	17.1	-39.6	45.4	-37.1	12.9	45.4	357.2	C2'-exo

Assignment of backbone suite names Backbone suite is defined as the sugar-to-sugar unit (Richardson *et al.*, 2008). It is an alternative to the commonly-used term nucleotide, defined as the phosphate-to-phosphate unit. A total of 53 backbone conformer bins have been designated and expressed in mnemonic 2-letter names (e.g., 1a, 5z, with outliers signified by !!). The DSSR implementation gives similar results to those determined by Suitename from the Richardson laboratory.

```
bin: name of the 12 bins based on [delta(i-1), delta, gamma], where
      delta(i-1) and delta can be either 3 (for C3'-endo sugar) or 2
      (for C2'-endo) and gamma can be p/t/m (for gauche+/trans/gauche-
      conformations, respectively) (2x2x3=12 combinations: 33p, 33t,
      ... 22m); 'inc' refers to incomplete cases (i.e., with missing
      torsions), and 'trig' to triages (i.e., with torsion angle
      outliers)
cluster: 2-char suite name, for one of 53 reported clusters (46
      certain and 7 wannabes), '__' for incomplete cases, and
      '!!!' for outliers
suiteness: measure of conformer-match quality (low to high in range 0 to 1)

[Ref: Richardson et al. (2008): "RNA backbone: consensus all-angle
conformers and modular string nomenclature (an RNA Ontology
```

Consortium contribution)." RNA, 14(3):465-481]

	nt	bin	cluster	suiteness
1	U A.U2647	inc	--	0
2	G A.G2648	33p	1a	0.052
3	C A.C2649	33p	1a	0.665
4	U A.U2650	33p	1a	0.875
5	C A.C2651	33p	1a	0.871
6	C A.C2652	33p	1a	0.919
7	U A.U2653	33p	1a	0.929
8	A A.A2654	32p	5z	0.849
9	G A.G2655	22t	4s	0.730
10	U A.U2656	23p	#a	0.842
11	A A.A2657	33p	1a	0.693
12	C A.C2658	33p	1a	0.884
13	G A.G2659	33p	1a	0.894
14	U A.U2660	33p	1g	0.736
15	A A.A2661	33p	1L	0.688
16	A A.A2662	33p	1a	0.692
17	G A.G2663	33t	1c	0.321
18	G A.G2664	33p	1a	0.878
19	A A.A2665	33t	1e	0.875
20	C A.C2666	33p	1a	0.891
21	C A.C2667	33p	1a	0.887
22	G A.G2668	33p	1a	0.756
23	G A.G2669	33p	1a	0.625
24	A A.A2670	33p	1a	0.914
25	G A.G2671	33p	1a	0.878
26	U A.U2672	33p	1a	0.912
27	G A.G2673	trig	!!	0

Concatenated suite string per chain. To avoid confusion of lower case modified nucleotide name (e.g., 'a') with suite cluster (e.g., '1a'), use --suite-delimiter to add delimiters (matched ')' by default).

```
1 A RNA nts=27 U1aG1aC1aU1aC1aC1aU5zA4sG#
→ aU1aA1aC1aG1gU1LA1aA1cG1aG1eA1aC1aC1aG1aA1aG1aU!!G
```

Note that in assigning suite names, only the six backbone torsion angles (α to ζ) are used. The χ torsion angle, which characterizes the relative sugar-base orientation, is not taken into consideration. Moreover, the 53 defined backbone conformer bins are RNA-centric: even for the classic B-DNA dodecamer structure 355d, 16 out of 22 suites (~73%) are classified as outliers (!!).

3.3 Default run on 1ehz (76-nt tRNA^{Phe}), summary notes

The cloverleaf 2D structure of tRNA has become an iconic image in structural biology. The four stems (acceptor, D, anti-codon, and T) form two halves of the L-shaped 3D structure through coaxial stacking interactions, each consisting of two stems. To our knowledge, DSSR is the only software tool that can neatly delineate the L-shaped 3D vs. the cloverleaf 2D structures of a tRNA molecule, starting from atomic coordinates (Figure 10).

In this section, we will use the crystal structure of yeast phenylalanine tRNA (tRNA^{Phe}) solved at 1.93 Å resolution (PDB entry 1ehz) as an example. The tRNA^{Phe} structure 1ehz is more complicated than the bulged-G/GUAA tetraloop (1msy), and helps illustrate more advanced features than those detailed in Section 3.2.

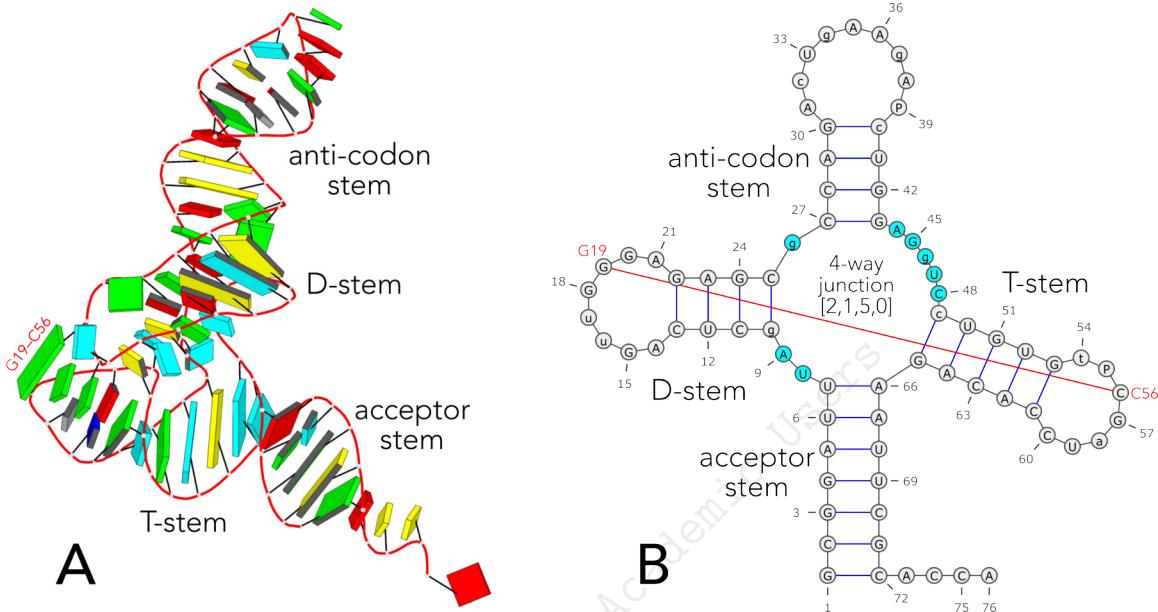


Figure 10: 3D and 2D structures of PDB entry 1ehz (tRNA^{Phe}). (A) 3D schematic auto-created via DSSR-PyMOL integration (Lu, 2020). (B) 2D diagram rendered with VARNA (Darty *et al.*, 2009), using DSSR-derived 2D structural information. This figure was annotated using Inkscape (<https://inkscape.org>).

3.3.1 Brief summary

The screen output of a DSSR run on 1ehz is shown below. Note that DSSR correctly identifies 14 modified nucleotides, 2 helices, 4 stems, 3 hairpin loops, and 1 four-way junction loop, among other things (see Figure 10).

```

total number of nucleotides: 76
modified nucleotides: 14
total number of base pairs: 34
total number of helices: 2
total number of stems: 4
total number of isolated WC/wobble pairs: 1
total number of multiplets: 4
total number of atom-base capping interactions: 4
total number of splayed-apart dinucleotides: 9
consolidated into units: 6
total number of hairpin loops: 3
total number of junctions: 1

```

```

total number of non-loop single-stranded segments: 1
total number of kissing loops: 1

Time used: 00:00:00:00

```

3.3.2 Modified nucleotides

In the tRNA^{Phe} structure 1ehz, 14 out of the 76 nts are modified (of 11 different types), as listed below. In contrast, the PDB entry 1msy detailed in Section 3.2 is composed entirely of canonical A, C, G, and U nucleotides.

List of 11 types of 14 modified nucleotides	nt	count	list
1 1MA-a	1		A.1MA58
2 2MG-g	1		A.2MG10
3 5MC-c	2		A.5MC40 ,A.5MC49
4 5MU-t	1		A.5MU54
5 7MG-g	1		A.7MG46
6 H2U-u	2		A.H2U16 ,A.H2U17
7 M2G-g	1		A.M2G26
8 OMC-c	1		A.OMC32
9 OMG-g	1		A.OMG34
10 PSU-P	2		A.PSU39 ,A.PSU55
11 YYG-g	1		A.YYG37

The above listing provides detailed information about each of the 11 types of modified nts: a 3-letter residue name followed by its 1-letter shorthand symbol in lower case (column nt), its frequency (column count), and a comma-separated list of the occurrences (column list). For example, no. 3 (the 5MC-c line) means 5MC is found twice in 1ehz, residue 40 and 49, respectively. Further information about modified nucleotides can be obtained via the RCSB Ligand Explorer. For example, click the link to check H2U (5,6-dihydrouridine-5'-monophosphate).

3.3.3 The four triplets

DSSR detects four base triplets in 1ehz, as detailed below. Users can use a molecular viewer, such as PyMOL or Jmol, to interactively explore **dssr-multiplets.pdb** (an auxiliary output file). Molecular images shown in Figure S1 of the supplemental PDF of Lu *et al.* (2015) can now easily be created via DSSR-PyMOL integration (Lu, 2020).

List of 4 multiplets
1 nts=3 UAA A.U8,A.A14,A.A21
2 nts=3 AUA A.A9,A.U12,A.A23
3 nts=3 gCG A.2MG10 ,A.C25,A.G45
4 nts=3 CGg A.C13,A.G22,A.7MG46

3.3.4 Coaxial stacks (relationship between helix and stems)

The connection between the two helices and the four stems is available from the main output file. For easy reference, the relevant portions are extracted and listed below. By referring to Figure 10, the meaning of the output should be easy to follow.

```

helix#1[2] bps=15
helix#2[2] bps=15

stem#1[#1] bps=7
stem#2[#2] bps=4
stem#3[#2] bps=4
stem#4[#1] bps=5

List of 2 coaxial stacks
 1 Helix#1 contains 2 stems: [#1,#4]
 2 Helix#2 contains 2 stems: [#3,#2]
```

3.3.5 Three hairpin loops

DSSR identifies three hairpin loops, as detailed below (see Figure 10).

```

List of 3 hairpin loops
1 hairpin loop: nts=10; [8]; linked by [#2]
  summary: [1] 8 [A.13 A.22] 4
    nts=10 CAGuuGGGAG A.C13,A.A14,A.G15,A.H2U16,A.H2U17,A.G18,A.G19,A.G20,A.A21,A.G22
      nts=8 AGuuGGGA A.A14,A.G15,A.H2U16,A.H2U17,A.G18,A.G19,A.G20,A.A21
2 hairpin loop: nts=11; [9]; linked by [#3]
  summary: [1] 9 [A.30 A.40] 4
    nts=11 GAcUgAAgAPc A.G30,A.A31,A.OMC32,A.U33,A.OMG34,A.A35,A.A36,A.YYG37,A.A38,A.PSU39,
      ↗ A.5MC40
      nts=9 AcUgAAgAP A.A31,A.OMC32,A.U33,A.OMG34,A.A35,A.A36,A.YYG37,A.A38,A.PSU39
3 hairpin loop: nts=9; [7]; linked by [#4]
  summary: [1] 7 [A.53 A.61] 5
    nts=9 GtPCGaUCC A.G53,A.5MU54,A.PSU55,A.C56,A.G57,A.1MA58,A.U59,A.C60,A.C61
      nts=7 tPCGaUC A.5MU54,A.PSU55,A.C56,A.G57,A.1MA58,A.U59,A.C60
```

3.3.6 One four-way junction loop

DSSR detects a 4-way junction loop, delineated by the four stems ([#1,#2,#3,#4]) with [2,1,5,0] nucleotides connecting each consecutive pair of stems, respectively (Figure 10). This loop is well-documented in literature, forming the core of the cloverleaf. Note that the junction loop contains four modified nucleotides (2MG10, M2G26, 7MG46, and 5MC49), which are represented by the lower case base symbols in the 2D diagram (Figure 10B).

```

List of 1 junction
1 4-way junction: nts=16; [2,1,5,0]; linked by [#1,#2,#3,#4]
  summary: [4] 2 1 5 0 [A.7 A.66 A.10 A.25 A.27 A.43 A.49 A.65] 7 4 4 5
    nts=16 UUAgCgCGAGgUCcGA A.U7,A.U8,A.A9,A.2MG10,A.C25,A.M2G26,A.C27,A.G43,A.A44,A.G45,A
```

```

→ .7MG46 ,A.U47 ,A.C48 ,A.5MC49 ,A.G65 ,A.A66
nts=2 UA A.U8 ,A.A9
nts=1 g A.M2G26
nts=5 AGgUC A.A44 ,A.G45 ,A.7MG46 ,A.U47 ,A.C48
nts=0

```

3.3.7 Splayed-apart conformations

DSSR detects dinucleotide splayed-apart conformations where the two bases are far away from one another. Based on the intermediate phosphorus (P) atom position and the origins (origin1 and origin2) of the two bases of a N1-p-N2 dinucleotide, the following three geometric parameters are defined:

- angle: the origin1–P–origin2 angle in degrees. By default, a dinucleotide with angle $\geq 85^\circ$ is designated as splayed-apart.
- distance: the origin1–origin2 distance in Å.
- ratio: the ratio of distance origin1–origin2 over the sum of distances origin1–P and P–origin2.

Obviously, the three parameters are highly correlated. The angle criterion is more intuitive, and it has been selected as the criterion for splayed-apart conformation ($\geq 85^\circ$ by default). Nine splayed-apart dinucleotides are identified in 1ehz, as detailed below. Notably, eight out of the nine cases are located in loop regions; the other one (C75-p-A76) is in the single-stranded region at the 3' end.

```

List of 9 splayed-apart dinucleotides
  1 A.U7      A.U8      angle=127    distance=17.2      ratio=0.90
  2 A.H2U16   A.H2U17   angle=146    distance=19.5      ratio=0.96
  3 A.H2U17   A.G18     angle=106    distance=15.8      ratio=0.80
  4 A.G19     A.G20     angle=130    distance=16.8      ratio=0.91
  5 A.7MG46   A.U47     angle=139    distance=19.0      ratio=0.94
  6 A.U47     A.C48     angle=157    distance=19.7      ratio=0.98
  7 A.C48     A.5MC49   angle=148    distance=17.5      ratio=0.96
  8 A.C60     A.C61     angle=91     distance=13.8      ratio=0.71
  9 A.C75     A.A76     angle=160    distance=18.7      ratio=0.98
-----
Summary of 6 splayed-apart units
  1 nts=2 UU A.U7 ,A.U8
  2 nts=3 uuG A.H2U16 ,A.H2U17 ,A.G18
  3 nts=2 GG A.G19 ,A.G20
  4 nts=4 gUCC A.7MG46 ,A.U47 ,A.C48 ,A.5MC49
  5 nts=2 CC A.C60 ,A.C61
  6 nts=2 CA A.C75 ,A.A76

```

Consecutive splayed-apart dinucleotides, if any, are merged into splayed-apart units, each

with two or more continuous nucleotides. Moreover, the atomic coordinates of the units are extracted into an NMR-style ensemble file (named `dssr-splays.pdb` by default) for easy visualization. As shown in the **Summary of 6 splayed-apart units** (see above), the nine splayed-apart dinucleotides are consolidated into six units. The largest unit (no. 4 on the gUCc line) contains four nucleotides (`7MG46`, `U47`, `C48`, `5MC49`; Figure 11), occurring in the 4-way junction loop that connects to the T-stem.

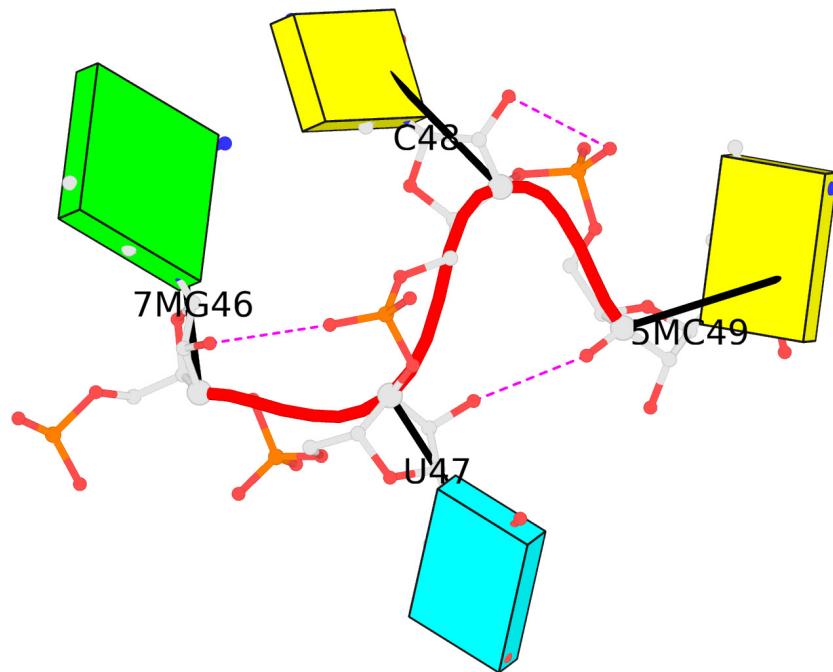


Figure 11: Splayed-apart unit with four nucleotides (`7MG46`, `U47`, `C48`, `5MC49`) in PDB entry `1ehz`. It connects the 4-way junction loop to the T-stem in the tRNA^{Phe} structure. Color code: C, yellow; G, green; U, cyan; H-bond: dashed line in magenta. The image was produced via DSSR-PyMOL integration (Lu, 2020).

3.3.8 Pseudoknot

The tRNA^{Phe} structure `1ehz` contains a pseudoknot, due to the formation of WC pair between G19 and C56 (long red line in Figure 10B, see also the listing below). This G–C pair is conserved in tRNA structures. It is located at the elbow of the L-shaped tertiary structure (Figure 10A) that brings the D-loop and the T Ψ C-loop together, via essentially a kissing loop interaction. The elbow WC pair stacks upon a base triplet of the T-box riboswitch (Zhang and Ferré-D’Amaré, 2013), apparently playing an import role in the formation of the stem-I-tRNA complex (PDB entry `4lck`).

In DSSR, pseudoknots are formed only via canonical pairing interactions, following the convention in the RNA field. Pseudoknots can be of high complexity. In an increasing order, they are characterized using [], {}, <>, Aa, Bb, Cc, etc. tRNA^{Phe} contains the simplest pseudoknot, with only one pseudoknotted pair represented by a matched []. Please refer to Section 3.8 for a more complicate example.

```
>1ehz-A #1 nts=76 0.09(2.86) [chain] RNA
GCGGAUUUAgCUCAGUuGGGAGAGCgCCAGAcUgAAGAPcUGGAGgUCCeUGUGtPCGaUCCACAGAAUUCGCACCA
(((((((((.....[...))).((((.....))))....((((...[...]....))))))))....
```

3.4 Default run on 1jj2 (50S subunit), four motifs

The crystal structure of the *Haloarcula marismortui* large ribosomal subunit (PDB entry 1jj2) serves to illustrate the high performance and rich features of DSSR. The program can analyze complicated RNA-containing structures as easily as smaller ones, such as 1msy and 1ehz discussed in previous sections. The screen output is shown below; it takes less than 20s to run on an M2-based MacBook Pro.

```
total number of nucleotides: 2876
total number of amino acids: 3701
total number of base pairs: 1475
total number of helices: 87
total number of stems: 179
total number of isolated WC/wobble pairs: 53
total number of multiplets: 252
total number of atom-base capping interactions: 239
total number of splayed-apart dinucleotides: 409
    consolidated into units: 257
total number of hairpin loops: 68
total number of bulges: 39
total number of internal loops: 66
total number of junctions: 37
total number of non-loop single-stranded segments: 30
total number of kissing loops: 5
total number of A-minor (types I and II) motifs: 107
total number of eXtended A-minor (type X) motifs: 49
total number of ribose zippers: 46
total number of kink turns: 8

Time used: 00:00:00:10
```

In addition to the large number of pairs, multiplets, loops of various types, DSSR automatically identifies four additional motifs—kissing loops, A-minor motifs, ribose zippers, and k-turns—that will be discussed in detail below.

As a side note, DSSR detects the following base pentaplet (AUAAG, see Figure 12), among

the 252 multiplets. It is held together by five base-base H-bonds and four additional H-bonding interactions involving O2' atoms.

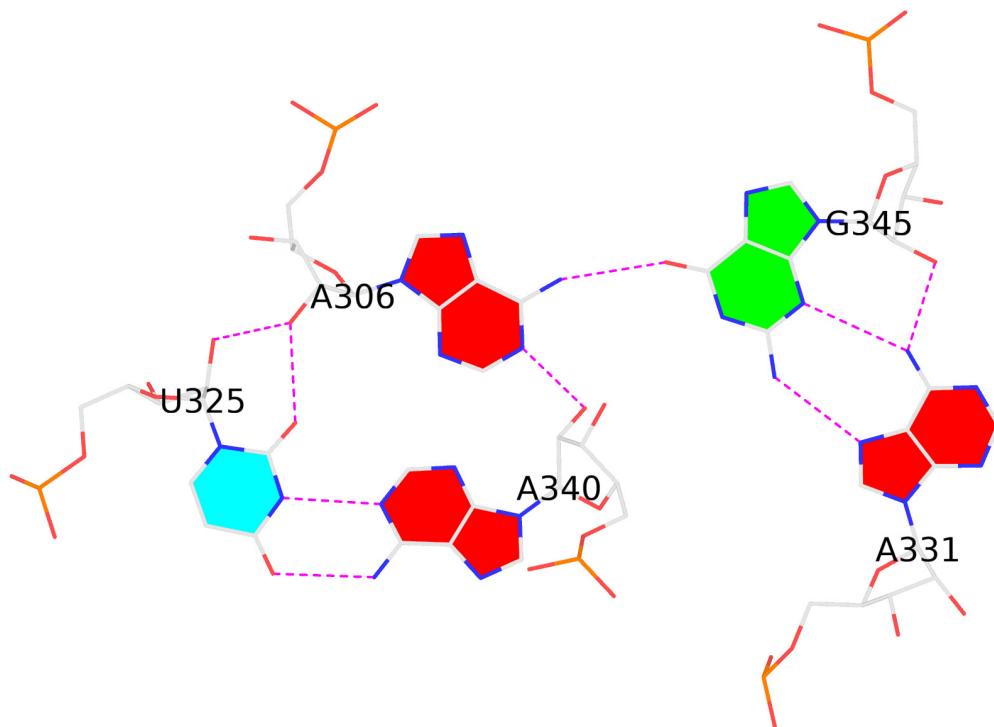


Figure 12: Base pentad (AUAAG) auto-identified by DSSR in PDB entry 1jj2. The five nts (A306, U325, A331, A340, G345) are all within the 23S rRNA. Color code: A, red, G, green; U, cyan; H-bond: dashed line in magenta. The image was produced via DSSR-PyMOL integration (Lu, 2020).

3.4.1 Kissing loops

The kissing-loop motif is characterized by canonical base pairs (isolated or in a stem) between two hairpin loops. DSSR identifies five (5) such motifs in 1jj2, as listed below.

```
List of 5 kissing loop interactions
1 isolated-pair #-42 between hairpin loops #51 and #53
2 isolated-pair #-6 between hairpin loops #6 and #8
3 stem #8 between hairpin loops #1 and #3
4 stem #9 between hairpin loops #1 and #3
5 stem #29 between hairpin loops #14 and #57
```

The 5th kissing loop is illustrated in Figure 13. The stem #29 in the kissing loop motif has six pairs, as detailed below:

stem#29[#15] bps=6					
strand-1	5'-CAUCGA-3'	bp-type		strand-2	3'-GUAGUU-5'
	helix-form AAA..				
1	O.C418	0.G2449	C-G WC	19-XIX	cWW cW-W
2	O.A419	0.U2448	A-U WC	20-XX	cWW cW-W
3	O.U420	0.A2447	U-A WC	20-XX	cWW cW-W
4	O.C421	0.G2446	C-G WC	19-XIX	cWW cW-W
5	O.G422	0.U2445	G-U Wobble	28-XXVIII	cWW cW-W
6	O.A423	0.U2444	A-U WC	20-XX	cWW cW-W

The two kissing hairpin loops, #14 and #57, are as follows:

```
14 hairpin loop: nts=9; [7]; linked by [#28]
  summary: [1] 7 [0.416 0.424] 5
  nts=9 GGCAUCGAC O.G416 ,O.G417 ,O.C418 ,O.A419 ,O.U420 ,O.C421 ,O.G422 ,O.A423 ,O.C424
  nts=7 GCAUCGA O.G417 ,O.C418 ,O.A419 ,O.U420 ,O.C421 ,O.G422 ,O.A423

57 hairpin loop: nts=9; [7]; linked by [#145]
  summary: [1] 7 [0.2442 0.2450] 5
  nts=9 GCUUGAUGC O.G2442 ,O.C2443 ,O.U2444 ,O.U2445 ,O.G2446 ,O.A2447 ,O.U2448 ,O.G2449 ,O.C2450
  nts=7 CUUGAUGC O.C2443 ,O.U2444 ,O.U2445 ,O.G2446 ,O.A2447 ,O.U2448 ,O.G2449
```

3.4.2 A-minor motifs

The interaction of the minor-groove edge of an adenine with the minor-groove side of a canonical pair is defined as A-minor motif (Nissen *et al.*, 2001). This abundant structural motif stabilizes RNA tertiary structures. Depending on the position of the adenine with respect to the interacting pair, the A-minor motif has been divided into four subtypes (I, II, III, and IV). Types I and II, presumably to be adenine-specific, are auto-identified by DSSR (see Figure 14). In types I and II A-minor motifs, the O2' atom of adenine is involved in H-bonding interactions with the pair.

Since each base has three edges (Figure 1), adenine can also employ its Watson-Crick edge or major-groove edge to interact with the minor-groove of a canonical pair. To account for such cases, the extended type X was introduced into DSSR. In contrast to type I and II A-motifs, adenine in type X does not resort to its O2' atom for interactions. These type X interactions were found in the crystal structure of the self-cleaving Pistol ribozyme (Nguyen *et al.*, 2017).

DSSR is unique in using base reference frames and base-pair parameters to characterize A-minor motifs. With `--a-minor=IUPAC_SYMBOL`, you can specify an IUPAC degenerate base symbol other than adenine (the default). For example, `--a-minor=N` means any base, including

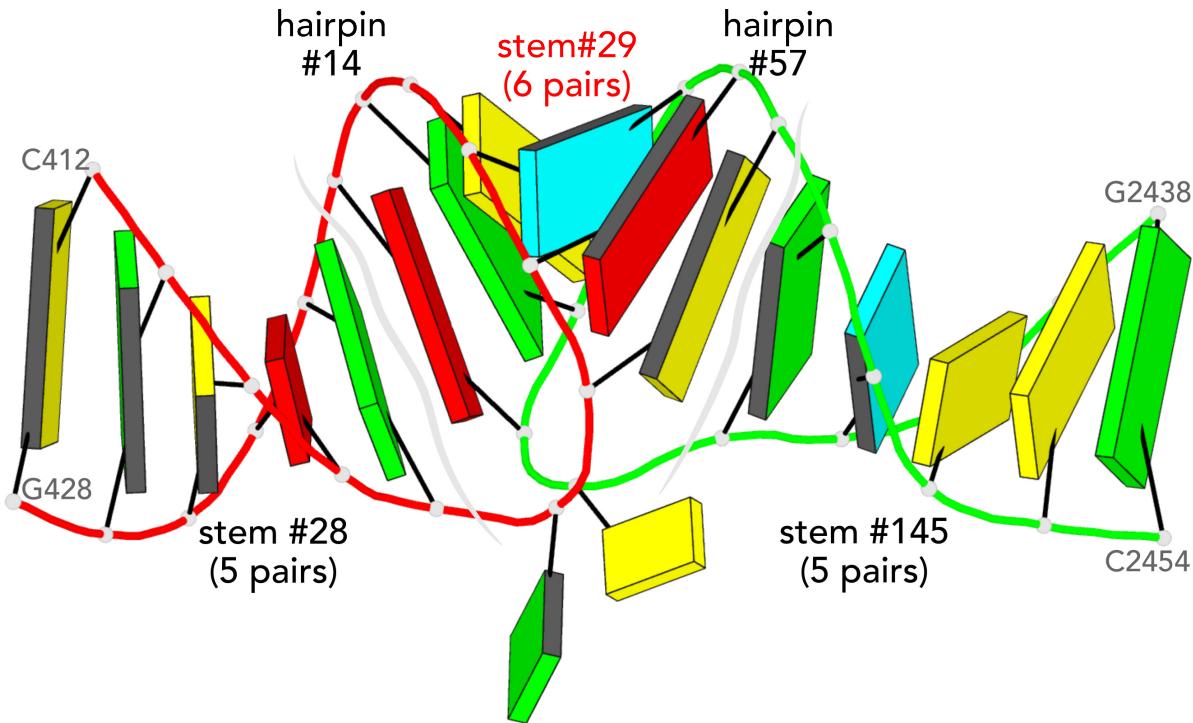


Figure 13: Kissing-loop motif identified in 1jj2. It occurs in the 23S rRNA. The nucleotides comprising stem #28 and hairpin loop #14, from C412 to G428, are shown in cartoon with backbone ribbon colored red. The nucleotides comprising stem #145 and hairpin loop #57, from G2438 to C2454, are similarly rendered but colored green. Color code for blocks: A, red; C, yellow; G, green; U, cyan; WC-pairs, per base in the leading strand. The image was produced via DSSR-PyMOL integration (Lu, 2020). This figure was annotated using Inkscape (<https://inkscape.org>).

A, C, G, T, or U.

DSSR detects a total of 156 A-minor motifs in 1jj2, including type X cases. The identified motifs can be explored interactively in molecular viewers, using the NMR-style ensemble file `dssr-Aminors.pdb`. Three sample entries, one each for types I (no. 4), II (no. 7), and X (no. 11), are listed below:

```
List of 156 A-minor motifs (types I, II, or X)
 4  type=I A|G-C      0.A69|0.G54,0.C65 WC
    +0.G54 H-bonds [2]: "N1-O2'(hydroxyl) [2.69] ,N3-N2(amino) [2.84]"
    -0.C65 H-bonds [2]: "O2'(hydroxyl)-O2'(hydroxyl) [2.62] ,O2'(hydroxyl)-O2(carbonyl)
    ↪ [2.61]"

 7  type=II A|G-C     0.A152|0.G41,0.C440 WC
    +0.G41 H-bonds [0]: ""
    -0.C440 H-bonds [3]: "O2'(hydroxyl)-O3' [3.17] ,O2'(hydroxyl)-O2'(hydroxyl) [2.73] ,N3-O2
    ↪ '(hydroxyl) [2.70]"
```

```

11 type=X A|U-G      0.A166|0.U919,0.G924 Wobble
    -0.U919 H-bonds [0]: ""
    +0.G924 H-bonds [3]: "N6(amino)-O2'(hydroxyl) [2.90], N6(amino)-N3 [3.15], N1-N2(amino)
    ↪ [2.98]"

```

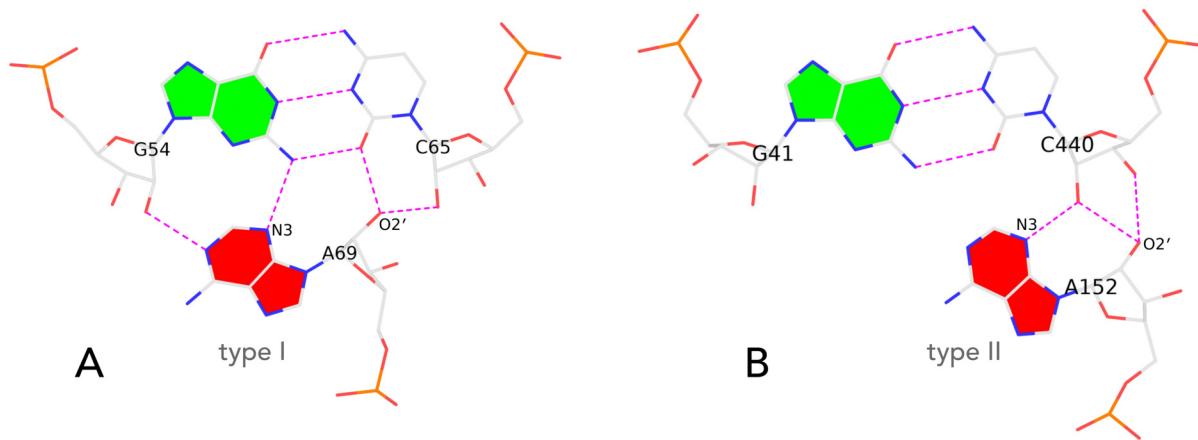


Figure 14: Types I and II A-minor motifs, widely believed to be specific to adenine. (A) Type I, where the O2' and N3 atoms of adenine lie inside the minor-groove edge of the G–C pair; (B) Type II, where the O2' atom of adenine lies outside but N3 remains inside the minor-groove edge of the G–C pair. Filled base rings reveal the similar orientation of adenine (red) and guanine (green). The images were produced via DSSR-PyMOL integration (Lu, 2020). This figure was annotated using Inkscape (<https://inkscape.org>).

For each entry, the type (I, II, or X) is followed by the A-minor motif identity, first in one-letter shorthand notation (e.g., A|G–C) and then the corresponding full specification of the interacting nucleotides (e.g., 0.A69|0.G54,0.C65). The name of the canonical pair (WC or wobble) is listed at the end. The next line lists the relative orientation (+ or –, see Figures 4, 6 and 14) and the H-bonding interactions between adenine and the first nucleotide of the pair. The third line reports similar results between adenine and the complementary base. Thus, for the type I A-minor motif no. 4 listed above, +0.G54 means that G54 and A69 have similar faces. Conversely, -0.C65 in the next line signifies that C65 and A69 have opposite faces.

Note that in type I A-minor motifs, the adenine interacts with both nucleotides of the canonical pair. In type II or X, however, the adenine often interacts with only one of the paired nucleotides.

3.4.3 Ribose zippers

First described in the P4-P6 domain of a group I intron (Cate *et al.*, 1996), ribose zipper is a tertiary interaction that plays an important role in RNA packing. In DSSR, a ribose-zipper motif is defined as two or more (very rare) consecutive H-bonding interactions between ribose 2'-hydroxyl groups from two RNA fragments (see Figure 15).

In 1jj2, DSSR detects a total of 46 ribose zippers, all of the smallest two-by-two type. The first 12 zippers are listed below, and the first one is illustrated in Figure 15.

```
List of 46 ribose zippers
 1 nts=4 UUAG 0.U26,0.U27,0.A1318,0.G1319
 2 nts=4 ACAC 0.A152,0.C153,0.A439,0.C440
 3 nts=4 AACC 0.A160,0.A161,0.C769,0.C770
 4 nts=4 AGAU 0.A189,0.G190,0.A204,0.U205
 5 nts=4 CGAA 0.C208,0.G209,0.A665,0.A666
 6 nts=4 UAAA 0.U233,0.A234,0.A436,0.A437
 7 nts=4 AACC 0.A242,0.A243,0.C376,0.C377
 8 nts=4 AAUG 0.A305,0.A306,0.U325,0.G326
 9 nts=4 GAAC 0.G471,0.A472,0.A773,0.C774
10 nts=4 AACC 0.A520,0.A521,0.C637,0.C638
11 nts=4 AACC 0.A551,0.A552,0.C1334,0.C1335
12 nts=4 AACU 0.A565,0.A566,0.C1263,0.U1264
```

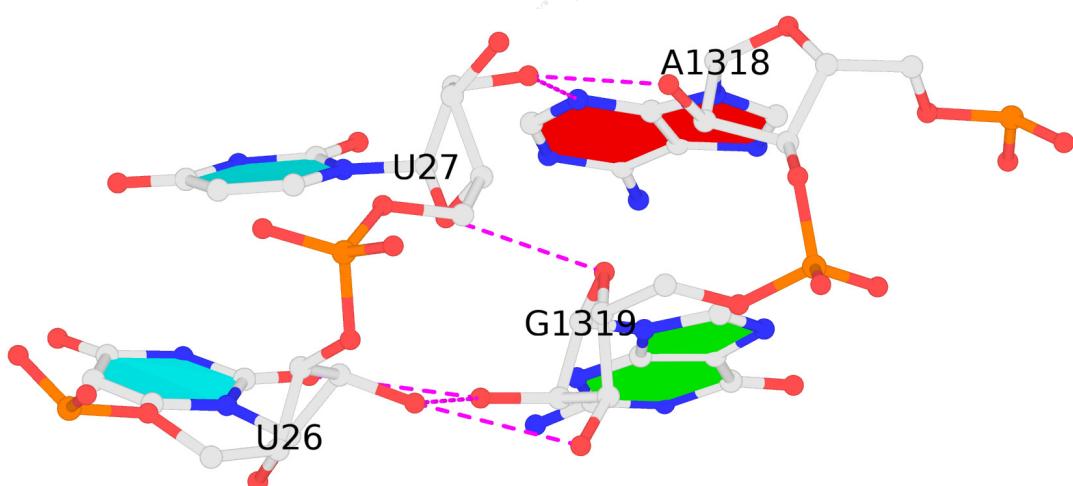


Figure 15: A ribose zipper identified in the 23S rRNA of PDB entry 1jj2. This motif consists of the U26-p-U27 dinucleotide in one strand, and A1318-p-G1319 along the other. The image was produced via DSSR-PyMOL integration (Lu, 2020).

3.4.4 Kink turns

The kink-turn (k-turn) is a widespread structural motif in RNA. It was first characterized in the large ribosomal subunit of *H. marismortui* (Klein *et al.*, 2001). The motif contains a sharp kink in the RNA duplex, introduced by an asymmetric internal loop. The loop is flanked by WC C–G pairs on one side and sheared G–A pairs on the other. A systematic nomenclature has been established for nucleotides in standard k-turns (Lilley, 2012). There is also a dedicated database for k-turns.

By default, DSSR defines the k-turn motif as an asymmetric internal loop with at least one sheared G–A pair and a large bending angle in the helical axis, among other criteria. The program detects three types of k-turns: normal, reverse, or else (for possible but unnamed cases). The type normal actually includes simple (standard or non-standard) and complex k-turns defined by Wang *et al.* (2014), as long as they belong to asymmetric internal loops.

DSSR finds a total of eight (8) k-turns in 1jj2. The first one is a normal k-turn, commonly known as *H. marismortui* Kt-7. It is listed below and depicted in Figure 16.

```
1 Normal k-turn; iloop#43; between stems [#11,#10]; bending-angle=54
C#11[CG 0.C93,0.G81] [GA 0.G97,0.A80] NC#10[CG 0.C100,0.G77]
strand1 nts=15; GGCAGAACCAUGG 0.G91,0.G92,0.C93,0.G94,0.A95,0.A96,0.G97,0.A98,0.A99
    ↵ ,0.C100,0.C101,0.A102,0.U103,0.G104,0.G105
strand2 nts=12; CCAUGGGGAGCC 0.C72,0.C73,0.A74,0.U75,0.G76,0.G77,0.G78,0.G79,0.A80,0.
    ↵ G81,0.C82,0.C83
```

- The first line starts with a serial number (1), followed by the type of the k-turn (normal). The motif is derived from internal loop #43, which is delineated by stems #11 and #10. The bending angle between the two stems is 54°.
- The second line shows that the canonical helix (C) consists of stem #11, with a C–G pair (between C93 and G81) closing the internal loop at one end. The non-canonical helix (NC) contains stem #10, with a C–G pair (between C100 and G77) closing the internal loop at the other end. The crucial sheared G–A pair is formed between G97 and A80. See Lilley (2012) for the nomenclature of C and NC helices in k-turns.
- The following two lines list nucleotides of the two strands composing the k-turn. The bulged strand (**strand1**) is longer and is listed first.

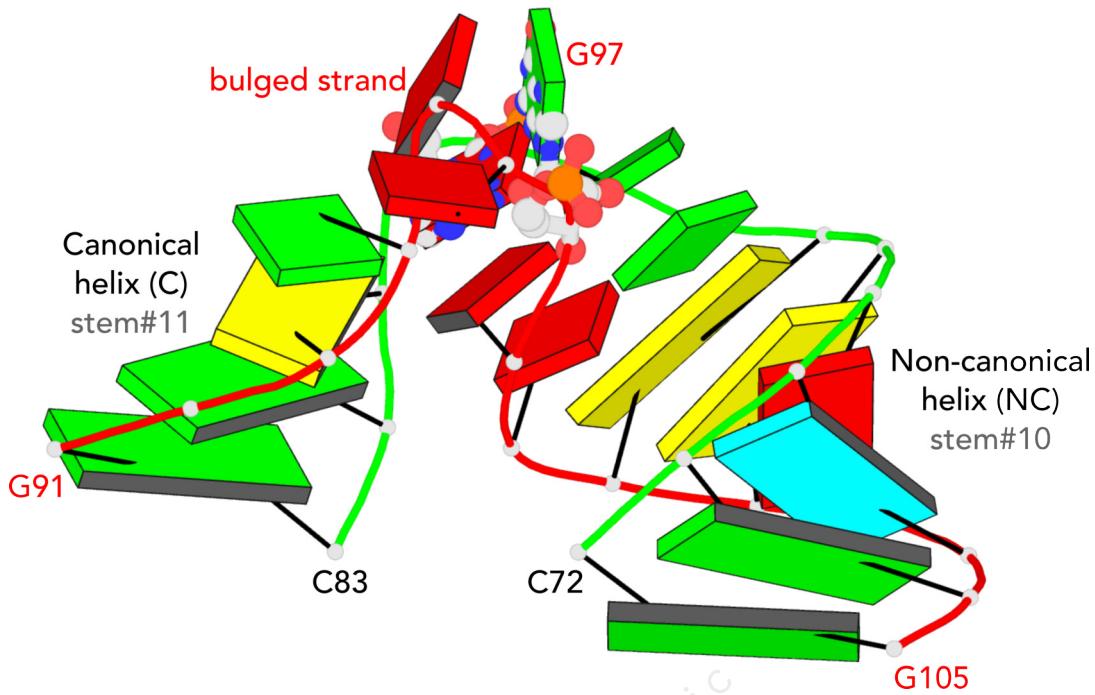


Figure 16: A standard k-turn (commonly known as *H. marismortui* Kt-7) identified in 1jj2. The bulged strand (G91 to G105) is colored red, and the other strand (C72 to C83) green. The crucial sheared G–A pair is highlighted in ball-and-stick representation. The canonical (C) and non-canonical (NC) helices are noted. The image was produced via DSSR-PyMOL integration (Lu, 2020). This figure was annotated using Inkscape (<https://inkscape.org>).

3.5 The U-turn motifs

When the `--u-turn` option[†] is specified, DSSR detects a variety of U-turns (Figure 17), including the two classical types (Gutell *et al.*, 2000). The first is the UNR-type originally characterized by Quigley and Rich (1976) in yeast tRNA^{Phe}. The second is the GNRA-type later established by Jucker and Pardi (1995) in GNRA tetraloops.

DSSR identifies two UNR-type U-turns in tRNA^{Phe} 1ehz, as detailed below. Note the listing includes an additional nucleotide at each end of a U-turn.

```
# x3dna-dssr -i=1ehz.pdb --u-turn -o=1ehz-Uturns.out

List of 2 U-turns
  1  A.U33-A.A36 H-bonds [1]: "N3(imino)-OP2[2.80]" nts=6 cUgAAg A.OMC32,A.U33,A.OMG34,A.A35
    ↪ ,A.A36,A.YYG37
```

[†]Unlike other common motifs (e.g., k-turn) which are auto-detected by DSSR, the `--u-turn` option must be specified explicitly for the identification and characterization of U-turn motifs. In fact, the option allows for quite a few variations as well.

```
2 A.PSU55-A.1MA58 H-bonds [1]: "N3(imino)-OP2[2.77]" nts=6 tPCGaU A.5MU54 ,A.PSU55 ,A.C56 ,A
  ↪ .G57 ,A.1MA58 ,A.U59
```

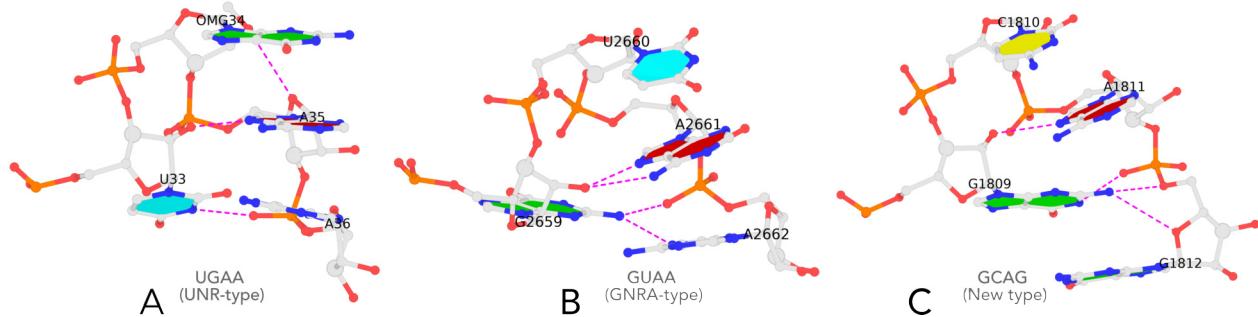


Figure 17: Three different types of U-turns. (A) The classic UNR-type in tRNA^{Phe} 1ehz. (B) The GNRA-type in 1msy. (C) A novel GCAG-type in 1jj2 (the 50S large ribosomal subunit). H-bonds are shown as dashed lines in magenta. The images were produced via DSSR-PyMOL integration (Lu, 2020). This figure was annotated using Inkscape (<https://inkscape.org>).

As its name implies, U-turn is characterized by a chain reversal of the RNA backbone (normally within a few nucleotides). Among other factors, the U-turn is stabilized by two key H-bonding interactions involving O2'/OP2 and base atoms (Figure 17).

Thirty-four (34) U-turns are found in 1jj2, the 50S large ribosomal subunit of *H. marismortui*. In addition to the well documented UNR- and GNRA-type U-turns, DSSR also finds other variants. An example is shown in Figure 17C, where the U-turn is formed via a GCAG fragment instead of a GNRA tetraloop. In the new U-turn, N1@G1809 makes an H-bond with OP2@G1812. The N2@G1809 atom is H-bonded to O5'@G1812, further stabilizing the U-turn. Moreover, unlike in GNRA-type U-turns where the G and A form a sheared G–A pair, the two Gs (G1809 and G1812) in the GCAG-type U-turn are not paired.

An examination of the chemical structures of nitrogenous bases shows other possible ways that can connect RNA base donors to the phosphate. DSSR allows for the exploration of many variations via the --u-turn option.

3.6 Identification and characterization of G-quadruplexes

G-quadruplex (G4) is a common type of higher-order nucleic acid structure formed from G-rich sequences. The sequence pattern is typically G₃₊N₁₋₇G₃₊N₁₋₇G₃₊N₁₋₇G₃₊. For example, the human telomere DNA has a repetitive sequence of GGGTTA. The building block of G4 is a tetrad of guanines (G-tetrad) arranged in a cyclic planar manner (Figure 18B), held

together by eight H-bonds via four consecutive G+G pairs (Figure 19A-B). G4 is formed by the stacking of G-tetrads and stabilized by cations (Figure 19C-D). G-quadruplexes are known to play important roles in the regulation of gene expression, the maintenance of genome stability, and serve as potential therapeutic targets (Rhodes and Lipps, 2015).

3.6.1 Existing issues

Experimentally solved 3D structures of G4s are deposited in the PDB (Burley *et al.*, 2018). These strutures provide important insights into the numerous functions that G4s can play at the atomic level. G4s in the PDB have revealed a plethora of topologies (Burge *et al.*, 2006; Lightfoot *et al.*, 2019): they can be intra- or intermolecular, formed by DNA or RNA; the four strands can be parallel, anti-parallel, or mixed in numerous combinations; loops connecting them can be lateral (edgewise), diagonal, or propeller (double-chain reversal), and of varying lengths (including zero); the helical twist can right-handed or left-handed, like in DNA; the stacks can consist of only two layers of G-tetrads instead of 3+, as often used in the pattern for sequence searches (see above); and the strands may contain bulges (Meier *et al.*, 2018).

However, limited annotations of G4s in the PDB unavoidably lead to false positive and false negative search results based on keywords. The lack of a widely accepted annotation tool also makes results reported in the literature difficult, if not impossible, to compare. Schematic representations that capture the essential features of G4s in a simple yet revealing manner are missing. It is easy to get lost in detailed descriptions and complicated illustrations while reading publications on G4s. Clearly, the increasing number of G4s available in the PDB calls for a pragmatic software tool that can identify this important class of structures automatically, characterize them consistently, and visualize them intelligibly.

3.6.2 DSSR solves known problems

DSSR detects and annotates G4s, starting from atomic coordinates in PDB or mmCIF format. It identifies G-tetrads and arranges them into G4 helices, composed of G4 stems (canonical G4-quadruplexes) via coaxial stacking interactions (Figure 19C). Note that the terms G-tetrads, G4 helices, and G4 stems in G-quadruplexes parallel those in duplexes: base pairs, helices, and stems, respectively. The similarities carry further to the concept of coaxial stacking and the calculation of rigid-body parameters (including helical twist and rise). DSSR is consistent; it provides systemic solutions to a set of related topics (pairs, duplexes, and quadruplexes).

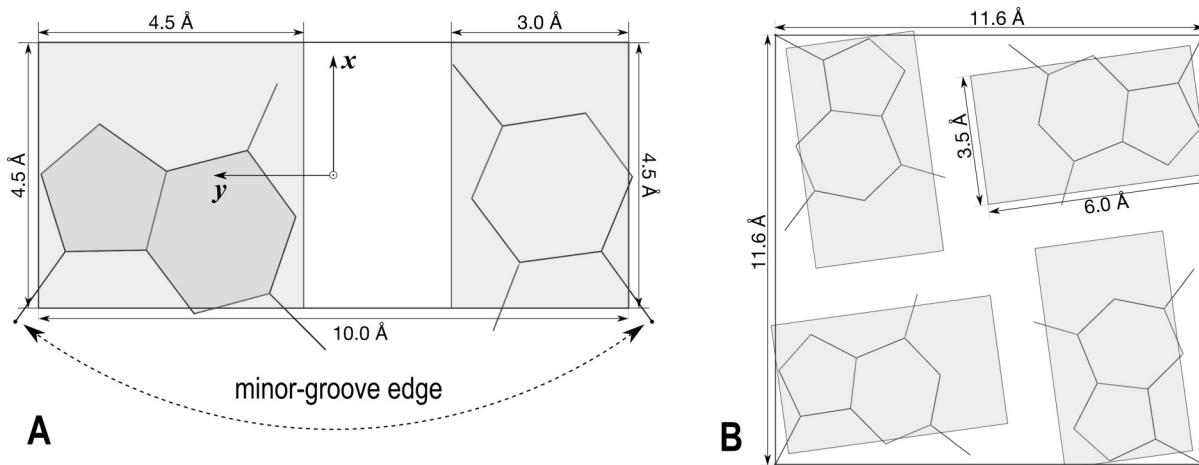


Figure 18: Definition of various types of DSSR blocks, illustrated using idealized, planar WC-pair and G-tetrad geometries. (A) Purine and pyrimidine base blocks and the WC-pair block in default dimensions. (B) The slim purine block and the square G-tetrad block for simplified visualizations of G-quadruplexes. This figure is taken from Lu (2020).

G4 stems are categorized in terms of loops connecting the four strands, by common names (chair, basket, etc.), or the revised Webba da Silva structural descriptors (Dvorkin *et al.*, 2018) (see Figure 20). DSSR accounts for bulges, identifies V-loops (Zhang *et al.*, 2001), characterizes G4s using rigid-body parameters, and quantifies stacking with overlapping areas. The program introduces innovative schematic representations (Lu, 2020), highlighting G-tetrads in PyMOL with unprecedented clarity (Figures 18 to 20).

It is worth noting that the assigned descriptors for PDB entries 2gku by 3(-p-1_n-1_w) and 2lod by 3(-pd+1_n) in Figure 1 of Dvorkin *et al.* (2018) are wrong (see Figure 20). DSSR revised the notations by using upper case letters (instead of lower case) L for lateral, P for propeller, and D for diagonal loops, and lower case letters (instead of subscripts) n for narrow and w for wide grooves. The Dvorkin *et al.* (2018) 2gku assignment is incorrect because it swaps the narrow and wide grooves: 3(-P-Lw-Ln) in DSSR vs. 3(-p-1_n-1_w). For 2lod, Dvorkin *et al.* (2018) mistake narrow for wide groove: 3(-PD+Lw) in DSSR vs. 3(-pd+1_n). These two cases clearly illustrate the power of a robust software tool like DSSR: applying it consistently can correct errors even from leading experts who proposed the descriptors in the first place[†].

DSSR further expands the descriptors by introducing the diagonal-propeller loop (X)

[†]We communicated with Dr. Mateus Webba da Silva who graciously acknowledged the mis-assignments for PDB entries 2gku and 2lod. More importantly, we have since collaborated with him on G-quadruplexes.

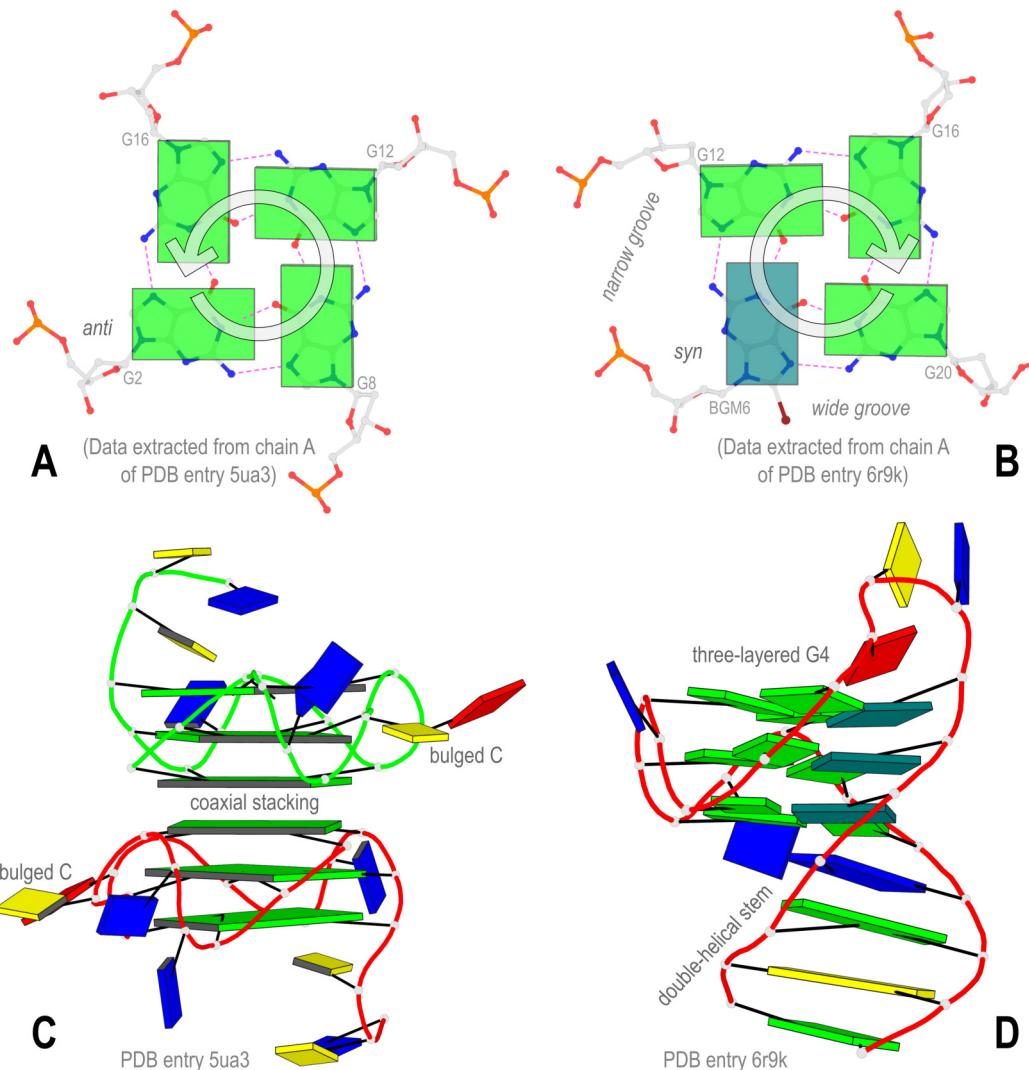


Figure 19: Innovative block representations introduced in DSSR to simplify the visualization of G-quadruplexes. (A) The *anti* guanine (lower-left) leads to a counter-clockwise orientation of H-bonding interactions (dashed lines in magenta) from the WC edge to the major-groove (Hoogsteen) edge. (B) The *syn* guanine (colored teal) reverses the direction of the H-bonds to clockwise and also creates a wide groove and a narrow groove. (C) A parallel G-quadruplex (Meier *et al.*, 2018) illustrated with square blocks for G-tetrads, highlighting a six-layered G4-helix composed of two three-layered G4-stems via coaxial stacking. The two bulged cytosines are marked. (D) Another G4 with a (1+3) hybrid conformation (Karg *et al.*, 2019), emphasizing the duplex-quadruplex transition, and the three guanines in *syn* conformation (teal). This figure is taken from Lu (2020).

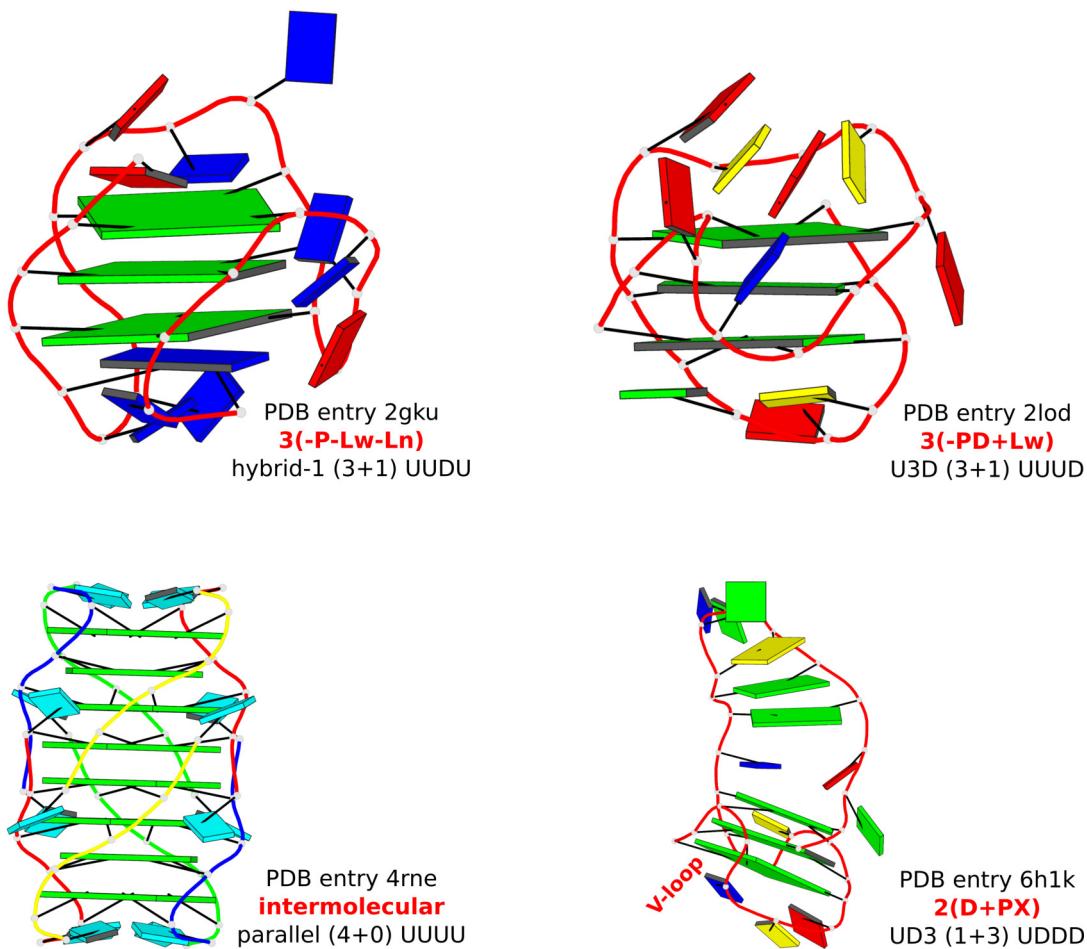


Figure 20: DSSR-derived topological descriptors of G-quadruplexes. Four canonical G4s from the PDB are used as examples: 2gku (top left), 2lod (top right), 4rne (bottom left), and 6h1k (bottom right). The images were produced via DSSR-PyMOL integration (Lu, 2020). This figure was annotated using Inkscape (<https://inkscape.org>).

to account for the special, but common V-shaped G4s (Zhang *et al.*, 2001) (see Figure 20). With the option `--g4-onz`, DSSR also classifies G-tetrads in intramolecular G4s into ONZ topological types following Popenda *et al.* (2020).

In short, starting from 3D atomic coordinates, such as those from the PDB or MD simulations, DSSR provides a streamlined solution to the identification, characterization, and visualization of G-quadruplexes. The program can be used to analyze one G4 structure to highlight its specific features, or compare many G4 structures to draw general principles.

3.6.3 Survey of G-quadruplexes in the PDB

DSSR-annotated G4s from the PDB are available at <http://G4.x3dna.org>. The curated results are easily searchable and regularly updated. The website is a uniquely helpful resource for the G4 community.

3.7 Detection and characterization of i-motifs

C+C pairs (type tW+W and Saenger XV, see Section 3.2.3), where one cytosine is protonated, can form an intercalated four-stranded structure (Figure 21). Unlike G-quadruplexes (Section 3.6), however, the intercalated i-motif actually consists of two parallel duplexes in opposite directions. The i-motif has potential roles in gene regulation and can serve as building blocks in nanostructure design (Abou Assi *et al.*, 2018).

DSSR can automatically identify and characterize i-motifs. It dissects each i-motif into four strands, up to three loops, and two double helices, among other things. Using PDB entry 1a83 (Han *et al.*, 1998) as an example, the DSSR output is shown below. With the annotations in Figure 21A, users should be able to easily understand the results.

```
List of 1 i-motif
 1 helix#1 nts=8 cCCCCCC A.MCY1,A.DC2,A.DC6,A.DC7,A.DC12,A.DC13,A.DC17,A.DC18
    strand-1(||3) nts=2 cC A.MCY1,A.DC2
    strand-2(||4) nts=2 CC A.DC6,A.DC7
    strand-3(||1) nts=2 CC A.DC12,A.DC13
    strand-4(||2) nts=2 CC A.DC17,A.DC18
  loop#1(1...2) nts=3 TTT A.DT3,A.DT4,A.DT5
  loop#2(2...3) nts=4 TTTA A.DT8,A.DT9,A.DT10,A.DA11
  loop#3(3...4) nts=3 TTT A.DT14,A.DT15,A.DT16
  Duplex(1||3), bps=2
    1 A.MCY1          A.DC12      C+C --      15-XV      tWW  tW+W
    2 A.DC2           A.DC13      C+C --      15-XV      tWW  tW+W
  ++++++=====
  Duplex(2||4), bps=2
    1 A.DC6           A.DC17      C+C --      15-XV      tWW  tW+W
    2 A.DC7           A.DC18      C+C --      15-XV      tWW  tW+W
```

When the `--more` option is specified (see Section 3.10), the base-pair morphology parameters are also available (e.g., shear and opening; see Section 3.10.3). The intercalation is quantified by the large rise and small twist. In Figure 21A, the large buckle and propeller of the top C+C pair (between C2 and C13) is obvious. See below:

Duplex(1 3), bps=2						
1 A.MCY1	A.DC12	C+C --		15-XV	tWW	tW+W
bp1-par: [2.02	0.38	-0.13	1.93	-6.72	178.38]	
step-par: [-1.15	1.23	5.56	-5.70	-5.62	16.62]	
heli-par: [9.07	-1.73	4.99	-18.14	18.39	18.44]	

bp2-pars:	[2.09	0.55	-0.19	34.21	18.25	183.56]
C1'-based:			rise=5.56		twist=16.26	
C1'-based:			h-rise=4.97		h-twist=18.11	
2 A.DC2	A.DC13		C+C	--	15-XV	tWW tW+W

For comparison, PDB entry 2n89 (Assi *et al.*, 2016), with an interstrand i-motif consisting of 10 C+C pairs, is illustrated in Figure 21B.

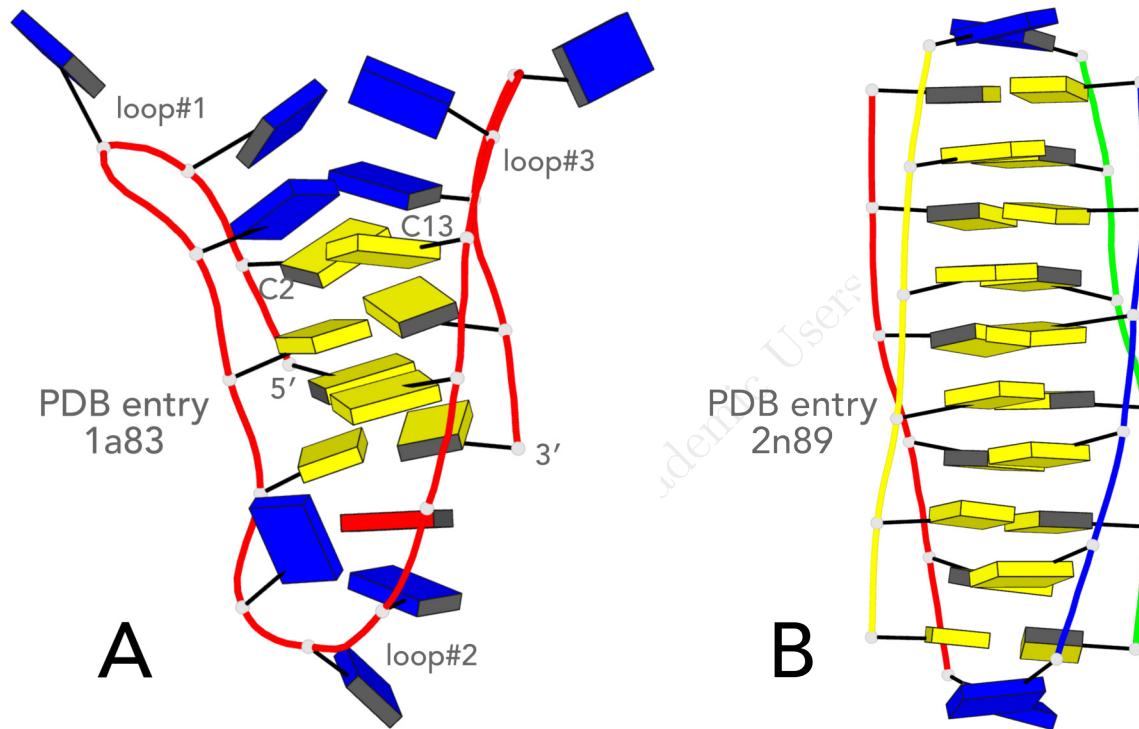


Figure 21: Two i-motifs in schematic representations. (A) PDB entry 1a83, an intrastrand i-motif with three loops. (B) PDB entry 2n89, an interstrand i-motif, no loops. Color code: C, yellow; T, blue; A, red. The images were produced via DSSR-PyMOL integration (Lu, 2020). This figure was annotated using Inkscape (<https://inkscape.org>).

3.8 Identification and removal of pseudoknots

RNA pseudoknot is formed by crossed canonical pairings between hairpin-loop nucleotides and those outside the enclosing stem (Figure 22). Pseudoknots are abundant in RNA structures and they are known to play essential functional roles (Theimer *et al.*, 2005). Pseudoknotted structures possess a challenge to many (e.g., dynamic programming based) RNA computational tools. DSSR provides a pragmatic means to characterize RNA pseudoknots of arbitrary complexity from 3D structures, like those deposited in the PDB. It also has the

capability to remove pseudoknots to produce a fully nested 2D structure in dbn notation.

3.8.1 Higher-order pseudoknots

The first biological unit of PDB entry 1ddy (Sussman *et al.*, 2000) is used as an example of higher-order pseudoknots. The PDB coordinates file is named 1ddy.pdb1, which contains an RNA chain (designated A) with 35 nucleotides. It is more complicated than the tRNA^{Phe} 1ehz (Section 3.3.8), yet simple enough to help make the points.

DSSR identifies a 2-order pseudoknot, as shown in the listing below and illustrated in Figure 22. Pairs of the first order are distinguished by matched [], and the second order by matched {}. They are colored blue and red, respectively, in Figure 22 for easy visualization.

```
# x3dna-dssr -i=1ddy.pdb1 -o=1ddy.out

This structure contains 2-order pseudoknot
  o You may want to run DSSR again with the '--nested' option which removes
    pseudoknots to get a fully nested secondary structure representation.

*****
Secondary structures in dot-bracket notation (dbn) as a whole and per chain
>1ddy nts=35 [whole]
GGAACCGGUGCGCAUAACCACCUUCAGUGCGAGCAA
.....((.{{[[....[[]))...].}.}])..
>1ddy-A #1 nts=35 0.46(2.84) [chain] RNA
GGAACCGGUGCGCAUAACCACCUUCAGUGCGAGCAA
.....((.{{[[....[[]))...].}.}])..
```

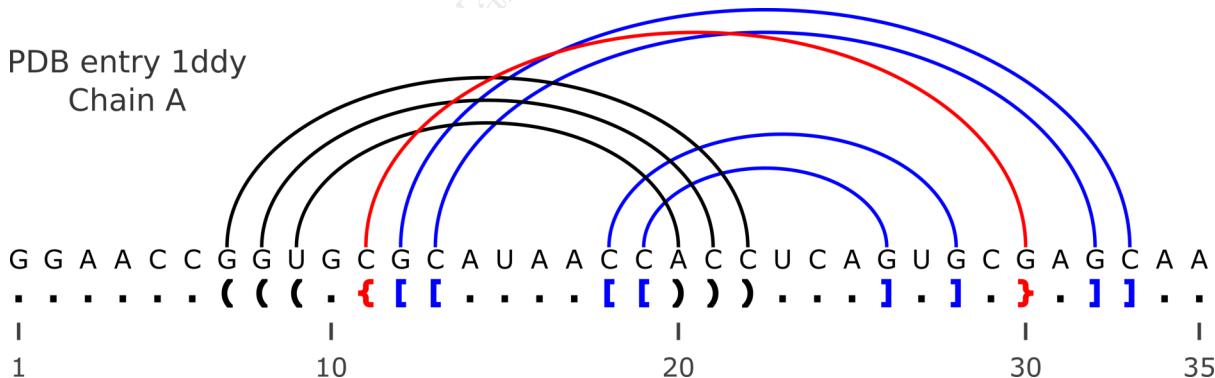


Figure 22: Second-order pseudoknots identified in PDB entry 1ddy (Sussman *et al.*, 2000). The linear 2D diagram was rendered using VARNA (Darty *et al.*, 2009), based on DSSR-derived dot-bracket notation. This figure was annotated using Inkscape (<https://inkscape.org>).

3.8.2 Pseudoknot removal

A closely related problem is pseudoknot removal, a topic nicely summarized in Smit *et al.* (2008). The `--nested` option does the trick by deriving a fully nested 2D structure in dbn. The results are shown in the listing below. In Figure 22, the nested 2D structure corresponds to removal of the crossing blue and red arcs, leaving only three black arcs above the matched parentheses.

```
# x3dna-dssr -i=1ddy.pdb1 --nested -o=1ddy-nested.out

This structure contains 2-order pseudoknot
  o You have chosen to remove the pseudoknots with the '--nested' option so
    only a fully nested secondary structure representation remains.

*****
Secondary structures in dot-bracket notation (dbn) as a whole and per chain
>1ddy nts=35 [whole]
GGAACCGGUGCGCAUAACCACCUUCAGUGCGAGCAA
.....(((.....))).....
>1ddy-A #1 nts=35 0.46(2.84) [chain] RNA
GGAACCGGUGCGCAUAACCACCUUCAGUGCGAGCAA
.....(((.....))).....
```

3.9 The `--pair-list` options

By default, DSSR automatically identifies all base pairs that fulfill simple geometric criteria (Lu *et al.*, 2015, Figure 1). While the criteria have been extensively tested and work well in real-world applications, the resultant base pairs may not be a perfect match in every situation. No size fits all; users may find certain pairs missing and some spurious in the default DSSR output. That is where the various `--pair-list` options come in. The intended usages are illustrated below, taking 1msy as an example.

The `--pair-list` option by default By simply specifying the `--pair-list` option, DSSR will output an additional auxiliary file, named `dssr-pairs.txt` by default. This file may serve as a starting point for customizations. For example, the option makes it possible to examine the features of base pairs expected on the basis of the complementarity of chain sequence, even if the base pairs are broken. The default option identifies the base pairs that are actually formed. After modifications, however, the default file needs to be renamed to something else (e.g., `my-pairs.txt`) to avoid being overwritten. Moreover, the default file name `dssr-pairs.txt` should not be used as the argument to `--pair-list-outfile` or `--pair-list-inpfile` (see below).

```
x3dna-dssr -i=1msy.pdb -o=1msy.out --pair-list # --> 'dssr-pairs.txt'
```

The file `dssr-pairs.txt` contains a list of pairs in a simple, intuitive format. The first line contains the numbers of base pairs. The followings lines list these pairs, one per line, with two serial numbers specifying the bases in a pair. Empty lines (those containing only white spaces) are ignored. The `#` and following characters in a line serve as comments.

```
# 13 (no. of base pairs)
 1    27  # o    1 A.U2647      A.G2673      U-G --
 2    26  # o    2 A.G2648      A.U2672      G-U Wobble
 3    25  # o    3 A.C2649      A.G2671      C-G WC
 4    24  # o    4 A.U2650      A.A2670      U-A WC
 5    23  # o    5 A.C2651      A.G2669      C-G WC
 6    22  # o    6 A.C2652      A.G2668      C-G WC
 7    21  # o    7 A.U2653      A.C2667      U-C --
 8    20  # o    8 A.A2654      A.C2666      A+C --
 9    10  # .    9 A.G2655      A.U2656      G+U Platform
10   19  # o   10 A.U2656      A.A2665      U-A rHoogsteen
11   18  # o   11 A.A2657      A.G2664      A-G Sheared
12   17  # o   12 A.C2658      A.G2663      C-G WC
13   16  # o   13 A.G2659      A.A2662      G-A Sheared
```

The `--pair-list-outfile` option The name of the output pairs list can be specified via the `--pair-list-outfile=NAME` option. One should not set NAME to the default file name `dssr-pairs.txt` (see above). This option has been designed to go with its counterpart `--pair-list-inpfile=NAME` with a user-supplied list of pairs (see below).

```
x3dna-dssr -i=1msy.pdb -o=1msy.out --pair-list-outfile=my-pairs.txt
```

The output file can be easily edited (i.e., by removing or adding pairs) as appropriate and then serves as input into DSSR (see below). For illustration purpose, let's remove the two beginning pairs (1 27 and 2 26), the G+U platform (9 10), and some comments in columns 4, 8, and 9 in the file `my-pairs.txt`.

```
10
 3    25  # 3      A.C2649 A.G2671
 4    24  # 4      A.U2650 A.A2670
 5    23  # 5      A.C2651 A.G2669
 6    22  # 6      A.C2652 A.G2668
 7    21  # 7      A.U2653 A.C2667
 8    20  # 8      A.A2654 A.C2666
10   19  # 10     A.U2656 A.A2665
11   18  # 11     A.A2657 A.G2664
12   17  # 12     A.C2658 A.G2663
13   16  # 13     A.G2659 A.A2662
```

The --pair-list-inpfile option The modified list of base pairs can be used for the customized analysis of the input structure. The user-supplied list of pairs is specified via the `--pair-list-inpfile=NAME` option. As noted above, NAME should not be the default file `dssr-pairs.txt`. In the input file, the number-of-pairs line (10 at the top) is optional. It can be removed, or commented out (e.g., `# 10 customized pairs`). With this option, DSSR then employs the customized pairs instead of identifying them *de novo*. DSSR then proceeds as usual to derive multiplets, helices, stems, and loops, etc.

```
x3dna-dssr -i=1msy.pdb --pair-list-inpfile=my-pairs.txt
```

By combining the writing and reading modes, users have full control over the pairs to be analyzed in DSSR.

3.10 The --more option

The `--more` option triggers additional parameters to be generated, mainly for the following sections: base pairs, helices, and stems. Since helices/stems share the same format for the added parameters, only an example from stems is shown. The following results are based on PDB entry 1msy with the DSSR command:

```
x3dna-dssr -i=1msy.pdb --more -o=1msy-more.out
```

3.10.1 Extra characterizations of base pairs

The G–U wobble pair formed by G2648 and U2672 (the 2nd pair on Page 18) is used as an example of the additional features enabled with the `--more` option.

```

1   2 A.G2648          A.U2672          G-U Wobble          28-XXVIII cWW  cW-W
2   [-167.8(anti) ~C3'-endo lambda=42.1] [-152.8(anti) ~C3'-endo lambda=68.6]
3   d(C1'-C1')=10.44 d(N1-N9)=8.84 d(C6-C8)=9.70 tor(C1'-N1-N9-C1')=-8.1
4   H-bonds [2]: "O6(carbonyl)-N3(imino)[2.78], N1(imino)-O2(carbonyl)[2.83]"
5   interBase-angle=9 Simple-bpParams: Shear=-2.44 Stretch=-0.01 Buckle=2.7 Propeller
6   ↘ =-8.6
    bp-pars: [-2.37 -0.60 0.11 4.67 -7.75 -2.95]
```

- Line no. 1: specification of the pair, as shown previously on Page 18.
- Line no. 2: the first bracket `[-167.8(anti) C3'-endo lambda=42.1]` corresponds to A.G2648. It contains three items: `-167.8(anti)` is the χ torsion angle formed by

$\text{O}4'-\text{C}1'-\text{N}9-\text{C}4$, $\text{C}3'$ -endo is the sugar pucker (as is the norm for RNA), and λ (lambda) is the angle $\text{N}9-\text{C}1'-\text{C}1'$ (A.U2672). The second bracket [-152.8(anti) $\text{C}3'$ -endo lambda=68.6] corresponds to A.U2672, with similar meanings for parameters, except that χ is defined by $\text{O}4'-\text{C}1'-\text{N}1-\text{C}2$, and λ is the angle $\text{N}1-\text{C}1'-\text{C}1'$ (A.G2648).

- Line no. 3: lengths of three virtual bonds ($\text{C}1'-\text{C}1'$, $\text{N}1-\text{N}9$, $\text{C}6-\text{C}8$), and the virtual torsion angle ($\text{C}1'-\text{N}1-\text{N}9-\text{C}1'$). Note here N1/N9 are general terms, referring to either N1 of pyrimidines or N9 of purines as appropriate. Similar conventions apply for the labeling of C6/C8.
- Line no. 4: detailed H-bonding information (atom names, types, and H-bond distances in square brackets).
- Line no. 5: inter-base-angle (9°), and a set of ‘simple’ parameters (Li *et al.*, 2019) which are easier to understand than the rigorous rigid-body parameters (listed below) for non-canonical pairs, especially when opening is $\sim 180^\circ$.
- Line no. 6: the six rigid-body base-pair parameters in the order of shear, stretch, stagger, buckle, propeller, and opening (see Figure 5).

3.10.2 Orientation of helices/stems

The additional output contains information about the best-fitted linear helical axis of a helix/stem, derived using a combination of equivalent $\text{C}1'$ and $\text{RN}9/\text{YN}1$ atom pairs along each strand (Lu *et al.*, 1997a). The results for the 1msy stem listed on Page 27 are shown below.

```

1   helical-rise: 2.60(0.18)
2   helical-radius: 9.12(0.79)
3   helical-axis: -0.776    -0.167    -0.608
4     point-one: 24.637    21.051    22.830
5     point-two: 16.686    19.344    16.602

```

- Line no. 1 (**helical-rise**): with numbers 2.60(0.18), represents the average helical rise (2.60) and its standard deviation (sd, 0.18) in Å. For a perfectly regular DNA/RNA duplex, the sd would be zero. RNA models generated with the **fiber** module (Lu and Olson, 2003; Li *et al.*, 2019) (see Section 5.2), for example, are characterized by numbers 2.55(0.00). In DSSR, the sd value is used to determine if a helix/stem is strongly curved, with a default cutoff of 0.6 Å. If the sd for a helix/stem is over the cutoff (as for the 1msy helix listed on Page 26), a * is appended at the end, serving as a hint that

the best-fitted linear helical axis may not be meaningful.

- Line no. 2 (**helical-radius**): with numbers 9.12(0.79), gives the average and sd of the perpendicular distances from phosphorus atoms (of both strands) to the helical axis. Typically, the mean radius is around 9.2 Å for RNA, and A- or B-form DNA. Textbooks normally list the diameter of the DNA duplex as ~20 Å.
- Line no. 3 (**helical-axis**): with three numbers -0.776 -0.167 -0.608, provides the normalized helical axis vector expressed in the original coordinate frame. This vector can be used to calculate DNA bending angles (e.g., in DNA-protein complexes), or to quantify the relative orientation between any two fairly straight helices/stems (Lu and Olson, 2008).
- Lines no. 4–5 (**point-one** and **point-two**): the following two lines are the end points (in the original coordinate frame) of the helical axis of the helix/stem. These two points can be added to the original PDB coordinates file for visualization of the helical axis or for rendering helical regions as cylinders (Lu and Olson, 2008; Lu *et al.*, 2015).

3.10.3 Base-pair morphology parameters for helices/stems

The 2nd dinucleotide step consisting of pairs C–G (C2649 with G2671) and U–A (U2650 with A2670) in the 1msy stem (Section 3.2.6) is used to illustrate the additional geometrical features for each base-pair step. Six extra lines are available (nos. 2–7 in the listing) with the **--more** option. For the definition of base-pair and step parameters, see Figure 5 and the 3DNA papers (Lu and Olson, 2003, 2008).

1	2 A . C2649	A . G2671	C-G WC	19-XIX	cWW	cW-W	
2	bp1-pars:	[0.09	-0.17	0.02	9.60	-15.93	-2.31]
3	step-pars:	[0.79	-1.42	3.25	-0.33	7.26	33.48]
4	heli-pars:	[-3.49	-1.39	2.88	12.42	0.57	34.24]
5	bp2-pars:	[0.11	-0.11	0.25	4.92	-16.45	6.33]
6	C1'-based:					twist=33.33	
7	C1'-based:					h-twist=34.09	
8	3 A . U2650	A . A2670	U-A WC	20-XX	cWW	cW-W	

- Line no. 2: the six rigid-body parameters in the order of shear, stretch, stagger, buckle, propeller, and opening for the C–G (C2649 with G2671) pair.
- Line no. 3: the six step parameters in the order of shift, slide, rise, tilt, roll, and twist for the dinucleotide step between pairs C–G (C2649 with G2671) and U–A (U2650 with A2670).

- Line no. 4: the six helical parameters in the order of *x*-displacement, *y*-displacement, helical rise, inclination, tip, and helical twist for the aforementioned dinucleotide step.
- Line no. 5: the six rigid-body parameters in the order of shear, stretch, stagger, buckle, propeller, and opening for the U–A (U2650 with A2670) pair.
- Line no. 6: rise and twist derived using the two consecutive C1'–C1' vectors, each defined by a pair in the dinucleotide step. These two parameters are related to the middle-step frame used to calculate the six step parameters (line no. 3). When non-WC pairs are involved, they normally make more intuitive sense than the corresponding rise and twist values reported on line no. 3.
- Line no. 7: helical rise and helical twist derived using the two consecutive C1'–C1' vectors, each defined by a pair in the dinucleotide step. These two parameters are related to the middle-helical frame used to calculate the six helical parameters listed on line no. 4.

3.11 The --idstr option

DSSR is designed with simplicity in mind. It employs sensible defaults to streamline the most common use cases. Powerful users may want to build upon DSSR by parsing its output files for other applications. The `--idstr=TYPE` option allows for the specification of nucleotides in different styles: `short`, `long`, `unit-id`, `jmol`, or the default.

Here are some examples, illustrated with the reverse Hoogsteen U–A pair (between U2656 and A2665) in PDB entry 1msy.

U2656	A2665	U-A rHoogsteen	24-XXIV	tWH	tW-M	# short
A.U2656	A.A2665	U-A rHoogsteen	24-XXIV	tWH	tW-M	# default
..A.U.2656.	..A.A.2665.	U-A rHoogsteen	24-XXIV	tWH	tW-M	# long
1 A U 2656	1 A A 2665	U-A rHoogsteen	24-XXIV	tWH	tW-M	# unit-id
[U]2656:A	[A]2665:A	U-A rHoogsteen	24-XXIV	tWH	tW-M	# jmol
///A/2656	///A/2665	U-A rHoogsteen	24-XXIV	tWH	tW-M	# pymol

The `short` format is easy to understand. It follows the convention in literature and specifies a residue by its name and sequence number (e.g., U2656). The `default` format prefixes `short` by chain id and a dot to separate them. The `long` form consists of six components in strict order of `model number`, `segment id`, `chain id`, `residue name`, `residue number`, and `insertion code`. The different fields are separated by dots.

To better connect DSSR to other tools, we have added three additional types for id string. The `unit-id` type is based on the work of Leontis and Zirbel *et al.*. Jmol and PyMOL are two of the most popular molecular viewers. The DSSR-Jmol integration (Hanson and Lu, 2017) actually employs `unit-id` since it is better structured (e.g., `|1|A|U|2656||||`) than the `jmol` style (e.g., `[U]2656:A`). The `pymol` style is widely recognized. It also contains a field for `segid` (segment id) whereas the `unit-id` and `jmol` forms do not have.

3.12 The --json option

The `--json` option represents DSSR results in the standard JSON data exchange format. The single JSON output file contains numerous DSSR-derived structural features, including those in the default main output, backbone conformations (Section 3.2.14), hydrogen bonds (Section 6.4), etc. The JSON interface makes it straightforward to integrate DSSR into other bioinformatics tools, or conveniently explore DSSR results via the command line as demonstrated below.

Using tRNA 1ehz as an example (Section 3.3), let's go over some simple yet powerful use cases. The following commands take advantages of jq, a lightweight and versatile command-line JSON processor.

```
x3dna-dssr -i=1ehz.pdb --json -o=1ehz-dssr.json
jq . 1ehz-dssr.json # reformatted for pretty output
x3dna-dssr -i=1ehz.pdb --json | jq . # the above two steps combined
```

With file `1ehz-dssr.json` in hand, one can easily extract various DSSR-derived structural features, in flexible ways:

```
jq .pairs 1ehz-dssr.json # list of 34 pairs
jq .multiplets 1ehz-dssr.json # list of 4 base triplets
jq .hbonds 1ehz-dssr.json # list of 113 hydrogen bonds
jq .helices 1ehz-dssr.json # list of 2 helices
jq .stems 1ehz-dssr.json # list of 4 stems

# a comprehensive list of parameters of the 76 nucleotides
jq .nts 1ehz-dssr.json

# list of 14 modified nucleotides
jq '.nts[] | select(.is_modified)' 1ehz-dssr.json

# select nucleotide id, delta torsion, sugar puckering, and ssZp
jq '.nts[] | {nt_id, delta, puckering, ssZp}' 1ehz-dssr.json

# same selections as above, but in 'comma-separated-values' (csv) format
jq -r '.nts[] | [.nt_id, .delta, .puckering, .ssZp] | @csv' 1ehz-dssr.json
```

Here are the results of running jq on selecting multiplets:

```
# jq .multiplets 1ehz-dssr.json
[{"index": 1, "num_nts": 3, "nts_short": "UAA", "nts_long": "A.U8,A.A14,A.A21", "planarity": 0.252}, {"index": 2, "num_nts": 3, "nts_short": "AUA", "nts_long": "A.A9,A.U12,A.A23", "planarity": 0.203}, {"index": 3, "num_nts": 3, "nts_short": "gCG", "nts_long": "A.2MG10,A.C25,A.G45", "planarity": 0.465}, {"index": 4, "num_nts": 3, "nts_short": "CGg", "nts_long": "A.C13,A.G22,A.7MG46", "planarity": 0.149}]
```

The JSON interface acts as a USB-like adaptor that connects DSSR to other bioinformatics resources in a structured way, with a clearly defined boundary. It also allows us great flexibility to refine human-readable output text without worrying about breaking third-party parsers or backward compatibility.

3.13 The `--nmr` option

The DSSR `--nmr` (or `--md`) option automates the analysis of an ensemble, such as NMR structures in the PDB or snapshots from MD simulations. The input coordinates file must be in either the legacy PDB format where each model is delineated by MODEL/ENDMDL tags, or the mmCIF format where each ATOM/HETATM record has an associated model number.

Here, the PDB entry 2n2d (Brčić and Plavec, 2015) is used as an example. It is an NMR ensemble of 10 different conformations of a DNA G-quadruplex with GGGGCC repeat. Two sample usages, with and without `--json`, are shown as follows.

```
# with --json, it is easy to parse the output.
x3dna-dssr -i=2n2d.pdb --nmr --json -o=2n2d-models.json
jq '.models[].parameters.num_Gtetrads' 2n2d-models.json

# default, results for each model in <model id="1">...</model> etc.
x3dna-dssr -i=2n2d.pdb --nmr -o=2n2d-models.out
```

The top-level skeleton of the JSON output is shown below. Each member of the `models` array contains three items: an auto-incremental `index` (from 1 to the number of models), the actual `model` number, and the `parameters` object which corresponds to the JSON output if the model is analyzed alone. Normally, `index` and `model` match each other, as is the case for `2n2d`. If models do not start from one or the numbers are not continuous, they will no longer share the same value (see below).

```
{
  "input_file": "2n2d.pdb",
  "num_models": 10,
  "models": [
    {
      "index": 1,
      "model": 1,
      "parameters": { ... }
    },
    ...
    {
      "index": 10,
      "model": 10,
      "parameters": { ... }
    }
  ]
}
```

The `--nmr` option allows for flexible ways to specify the models to be analyzed. Three examples are shown below. Note that in these cases, the values of `index` and `model` are no longer the same.

```
x3dna-dssr -i=2n2d.pdb --nmr=2+5      # 2 models: [2, 5]
x3dna-dssr -i=2n2d.pdb --nmr=2:3:10   # 3 models: [2, 5, 8]
x3dna-dssr -i=2n2d.pdb --nmr=3+5+6:9 # 6 models: [3, 5, 6, 7, 8, 9]
```

The `--json` option makes it easy to parse the output of multiple models pragmatically. In addition to NMR structures, trajectories from MD simulations can also be processed. Popular MD packages (AMBER, GROMACS, CHARMM, etc.) all have their own specialized binary formats for trajectories. By design, DSSR does not work on these binary files. They must be converted to the standard PDB or mmCIF format to be analyzed by DSSR. The combination

of `--nmr` and `--json` makes DSSR directly accessible to the MD community.

3.14 The `--pair-only` option

DSSR derives far more nucleic-acid structural features than a typical user may normally need. The option `--pair-only` instructs DSSR to generate only base-pairing information that is most fundamental for DNA/RNA structural analysis and annotation. It can also be combined with the `--more` and/or `--json` options. Speedwide, when the `--pair-only` option is enabled, DSSR runs approximately 10 times faster than the default.

3.15 The `--non-pair` option (interactions other than pairing)

With the `--non-pair` option, DSSR identifies H-bonding and base-stacking interactions between two nucleotides that do not form a pair. In PDB entry 1msy, DSSR identifies 30 non-pairing interactions. Three sample cases (nos. 16–18 in the full listing) are shown below. The results have been reformatted to fit the page width.

```

16 A.C2658    A.G2659    stacking: 0.4(0.1)--pm(>>,forward) interBase-angle=10 connected
                                         min-baseDist=3.34
17 A.G2659    A.A2661    interBase-angle=31 H-bonds[2]:
                           "O2'(hydroxyl)-N7[2.60], O2'(hydroxyl)-N6(amino)[3.26]"
                                         min-baseDist=3.97
18 A.G2659    A.G2663    stacking: 3.9(1.2)--mm(<>,outward) interBase-angle=4
                                         min-baseDist=3.35

```

G2659 is a crucial nucleotide in the GUAA (GNRA-type) tetraloop (Figure 3). It is involved in several interactions. In addition to the sheared G–A pair (with A2662) reported in Section 3.2.3, G2659 also participates in three non-pairing interactions. No. 16: G2659 stacks slightly on C2658, with an overlap area of $0.4(0.1)\text{ \AA}^2$. The term `connected` means that the two nucleotides are linked by a phosphodiester bond. No. 17: G2659 forms bifurcated H-bonds with A2661, involving the O2' atom of G2659 with N7 and N6 of A2661 simultaneously. No. 18: G2659 stacks heavily on G2663, with an overlap area of $3.9(1.2)\text{ \AA}^2$.

In all case, `min-baseDist` (in Å) means the minimum distance between base atoms. The inter-base angle (`interBase-angle`, in degrees) is also reported; closer to zero means the two bases are nearly parallel.

DSSR quantifies base-stacking by the overlap area (in \AA^2) between the two interacting bases. The base atoms are projected onto the mean plane to derive the overlapping polygon for area calculations. In the output, values in parentheses measure the overlap of base ring

atoms only, and those outside parentheses include exocyclic atoms (Lu and Olson, 2003; Lu *et al.*, 2015).

Base-stackings are classified into one of the following four categories: `pm(>,forward)`, `mp(<,backward)`, `mm(<,outward)`, and `pp(>,<,inward)`. Here `p` and `m` represent the plus and minus faces of the base ring, defined by the direction of the *z*-axis of the standard base reference frame (Figures 1 and 18). The symbols (`>`, `<`, `<`, and `><`) follow Parisien *et al.* (2009), except `pm(>)` is called forward instead of upward, and `mp(<)` backward instead of downward.

3.16 The --po4 option (phosphate interactions)

The phosphate group is negatively charged. It acts as H-bond acceptor or forms metal coordination complex. The exocyclic OP2 (and OP1, to a less extent) atoms are especially prominent in forming H-bonding interactions (Lu *et al.*, 2010), stacking over base rings (Section 3.2.9), and creating various types of U-turns (see Figure 17), etc.

Running DSSR on 1ehz with the `--po4` option leads to the following results.

```
# x3dna-dssr -i=1ehz.pdb --po4 -o=1ehz-po4.out

List of 18 phosphate interactions
 1 A.U7      OP1-hbonds [1]: "MG@A.MG580 [2.60]"
 2 A.A9      OP2-hbonds [1]: "N4@A.C13 [3.01]"
 3 A.A14     OP2-hbonds [1]: "MG@A.MG580 [1.93]"
 4 A.H2U16   OP2-cap: "A.H2U16"
 5 A.G18     OP1-hbonds [1]: "O2'@A.H2U17 [2.97]"
 6 A.G19     OP1-hbonds [2]: "N4@A.C60 [3.27] , MN@A.MN530 [2.19]"
 7 A.G20     OP1-hbonds [1]: "MG@A.MG540 [2.07]"
 8 A.A21     OP2-hbonds [1]: "MG@A.MG540 [2.11]"
 9 A.A23     OP2-hbonds [1]: "N6@A.A9 [3.12]"
10 A.A35    OP2-cap: "A.U33"
11 A.A36    OP2-hbonds [1]: "N3@A.U33 [2.80]"
12 A.YYG37   OP2-hbonds [1]: "MG@A.MG590 [2.53]"
13 A.C48    OP2-hbonds [1]: "O2'@A.7MG46 [3.55]"
14 A.5MC49   OP1-hbonds [1]: "O2'@A.C48 [3.13]" OP2-hbonds [1]: "O2'@A.U7 [2.68]"
15 A.U50    OP1-hbonds [1]: "O2'@A.U47 [2.71]"
16 A.G57    OP2-cap: "A.PSU55"
17 A.1MA58   OP2-hbonds [1]: "N3@A.PSU55 [2.77]"
18 A.C60    OP1-hbonds [1]: "N4@A.C61 [3.12]" OP2-hbonds [1]: "O2'@A.1MA58 [2.42]"
```

Here, entry no. 1 means that the OP1 atom of A.U7 (`OP1@A.U7`) is in coordination with metal A.MG580 (2.60 Å). Entry no. 2 says that `OP2@A.A9` is H-bonded with the N4@A.C13 (3.01 Å). Entry no. 16 signifies that `OP2@A.G57` stacks on base A.PSU55. As no. 6 shows, `OP1@A.G19` forms an H-bond with N4@A.C60 (3.27 Å) and coordinates with metal A.MN530 (2.19 Å) at the same time. OP1 and OP2 atoms from a given nucleotide can both participate

in H-bonding interactions simultaneously, as demonstrated in no. 14 for A.5MC49 and no. 18 for A.C60.

3.17 Miscellaneous options

3.17.1 The `--symmetry` option

By default, DSSR reads in the first model and thus analyze only a representative structure of an NMR ensemble. Biological units of x-ray crystal structures may contain multiple models formatted as an NMR-like ensemble in the PDB. This is confusing since the different models in a biological unit are symmetry related instead of being fully independent as in an NMR ensemble. In such cases, the `--symmetry` (or `--symm`) option is required for DSSR to process the entire structure of a biological unit with multiple models.

As an example, the asymmetric unit of PDB entry 4ms9 (Sheng *et al.*, 2014) is single-stranded (`4ms9.pdb`), while the biological unit is a double helix (`4ms9.pdb1`). Running DSSR on `4ms9.pdb` or `4ms9.pdb1` with default settings, it does not find any pairs. The `--symm` option needs to be used together with `4ms9.pdb1` (but not `4ms9.pdb`) to identify the 10 base pairs:

```
x3dna-dssr -i=4ms9.pdb1 --symm
```

Please note that DSSR reads atomic coordinates as provided in the input file. It does not expand an asymmetric unit into biological unit based on crystallographic symmetry information that may exist in the PDB or mmCIF file. Users must supply the biological unit file (`4ms9.pdb1`) and specify `--symm` for DSSR to work as expected.

3.17.2 The `--isolated-pair` option

By default, isolated canonical pairs are taken as a special case of stems in delineating various closed loops. They can be excluded, via the `--isolated-pair=not-in-loop` option, from playing such a role. This feature is useful to avoid spurious loops, especially junctions in complicated RNA structures.

As shown in Figure 3B, taking the isolated WC C–G pair (between C2658 and G2663) into account reveals the reported GUAA tetraloop (Correll *et al.*, 2003) and a [5,4] asymmetric internal loop. Otherwise, the tetraloop and the internal loop delineated by the C–G pair will be merged, leading to an enlarged hairpin loop consisting of 17 nucleotides (from C2652 to

G2668).

3.17.3 The **--nt-mapping** option

DSSR contains a built-in list of the most commonly used identifiers for canonical nucleotides, including A, C, G, and U for RNA, and DA, DC, DG, and DT for DNA. It uses a heuristic procedure to identify modified nucleotides and map them to their canonical counterparts (Lu *et al.*, 2015). Over the years, the method has proven to work well in real-world applications, as noted above for 1ehz (Section 3.3.2). It is one of the defining features that make DSSR just work.

DSSR has the **--nt-mapping** option that allows users to control the mapping process. For example, DSSR automatically maps 5MU (5-methyluridine 5'-monophosphate) to t (i.e., modified thymine) because of the 5-methyl group. With the option **--nt-mapping='5MU:u'**, DSSR would take 5MU as a modified uracil. This option allows for multiple mappings separated by comma.

The mapping of 5MU to u or t is obviously arbitrary. DSSR is robust against the ambiguity in designating a modified nucleotide to its nearest canonical counterpart. For example, mapping 5MU to u or t has minimal influence on DSSR-derived base-pair parameters and other structural features.

3.17.4 The **--prefix** option

By default, DSSR auxiliary output files are prefixed with `dssr`, as in `dssr-pairs.pdb`. These fixed files are overwritten on following DSSR runs in the same directory to reduce pollution and save space. With the **--prefix=text** option, the auxiliary output files will be prefixed by `text`. For example, with the following command, the auxiliary output files will be named `1ehz-pairs.pdb`, etc.

```
x3dna-dssr -i=1ehz.pdb -o=1ehz.out --prefix=1ehz
```

3.17.5 The **--auxfile** option

In applications such as the analyses of MD trajectories, one may be only interested in the DSSR main output file (in human-readable text or machine-friendly JSON format). This is where the **--auxfile=no** option comes in. It instructs DSSR not to generate any auxiliary files (Page 17).

3.17.6 The `--dbn-break` option

By default, DSSR employs the symbol & to separate multiple chains or chain breaks in dot-bracket notation, for compatibility with VARNA (Darty *et al.*, 2009). With `--dbn-break`, any character from the list '&.:,|+' can be used instead.

Using PDB entry 355d (Shui *et al.*, 1998, a B-DNA dodecamer) as an example, here are the results when + is used or the separator is removed. Without a separator character, the two DNA chains are merged into a seemingly ‘continuous’ strand. Moreover, the dbn implies a hairpin loop without bridging nucleotides, which would be physically impossible.

```
# x3dna-dssr -i=355d.pdb --dbn-break=+
>355d nts=24 [whole]
CGCGAATT CGCG+CGCGAATT CGCG
(((((((((+) )))))))))))

# x3dna-dssr -i=355d.pdb --dbn-break=no
>355d nts=24 [whole]
CGCGAATT CGCGCGCGAATT CGCG
((((((((( )))))))))
```

3.17.7 The `--sugar-pucker` option

Following 3DNA (Lu and Olson, 2003), DSSR employs the Altona and Sundaralingam (1972) definitions for sugar conformational analyses. Specifying `--sugar-pucker=westhof83` leads to the newer Westhof and Sundaralingam (1983) formulae to be used. These two methods give slightly different results for puckering amplitudes and pseudorotation phase angles. The DSSR `--sugar-pucker=westhof83` option provides identical numerical values as those from Curves+ (Lavery *et al.*, 2009).

3.17.8 The `--torsion360` option

By default, DSSR outputs torsion angles (see Section 3.2.14) in the range of -180° to 180° . The `--torsion360` option leads to torsion angles reported in the range of 0 to 360° .

3.17.9 The `--raw-xyz` option

By default, DSSR-derived auxiliary files (such as `dssr-pairs.pdb`) are transformed for better visualization (see Figure 8) and easy comparison. At times, it may be desirable to keep the original orientation; see Figures 2–3 of the DSSR paper (Lu *et al.*, 2015). The `--raw-xyz` option does the trick: it retains the raw atomic coordinates of selected residues.

3.17.10 The --cleanup option

This option removes auxiliary output files generated by DSSR. It can be used with the `--prefix` option to delete auxiliary files with customized names.

3.18 The analyze module (following 3DNA)

The DSSR `analyze` module serves as a drop-in replacement of the 3DNA `analyze` program. It reproduces key structural parameters from 3DNA, which is still maintained but without further development other than bug fixes. The module should be specified strictly as: `x3dna-dssr analyze`. The `analyze` keyword must be immediately after `x3dna-dssr`, without leading dashes.

Unlike the 3DNA `analyze` program which is limited to PDB-formatted files, this module can take mmCIF files as input. It also provides new features, including base-pair classifications. There are quite a few features that can be specified via options (see Section 3.1). By default (or with `--duplex`), the module analyzes double helical structures. With `--cehs`, it produces authentic CEHS parameters (El Hassan and Calladine, 1995). With `--single-strand` (or `--ss`), it takes the whole input structure as single-stranded: duplexes and G-quadruplexes can also be analyzed with this generic option.

The parameters are calculated and organized following the 3DNA `analyze` program. Moreover, the module produces rigid-body parameters files that can be fed into the `rebuild` module for customized model building (see Section 5.3).

3.18.1 Analysis of double helices

Using PDB entry 355d (Shui *et al.*, 1998) as an example, here is the output of the command:

```
x3dna-dssr analyze -i=355d.pdb
```

```
num_pairs=12 # no. of base pairs
*****
Local base-pair parameters
  bp      Shear    Stretch   Stagger   Buckle   Propeller   Opening
  1 C-G     0.28    -0.14     0.07     6.93    -17.31     -0.61
  2 G-C    -0.24    -0.18     0.49     9.34    -14.30     -2.08
  3 C-G     0.24    -0.17     0.16    -4.43     -5.40      0.43
  4 G-C    -0.26    -0.11     0.01    10.81    -9.45      1.01
  5 A-T    -0.04    -0.11     0.01     4.72    -15.31      1.60
  6 A-T     0.05    -0.05     0.07     0.44    -15.00      6.23
  7 T-A    -0.04    -0.12     0.17    -0.26    -16.74      3.93
  8 T-A    -0.11    -0.12    -0.00    -1.56    -16.36      5.12
  9 C-G     0.21    -0.13     0.00   -12.41    -10.27     -1.22
```

10 G-C	-0.11	-0.05	0.24	4.21	-9.60	3.21
11 C-G	0.16	-0.13	0.21	0.28	-17.42	-1.75
12 G-C	-0.24	-0.07	0.25	4.67	-4.95	-1.62

Local base-pair step parameters

bp	Shift	Slide	Rise	Tilt	Roll	Twist
1 C-G	0.09	0.04	3.20	-3.22	8.52	32.73
2 G-C	0.50	0.67	3.69	2.85	-9.06	43.88
3 C-G	-0.14	0.59	3.00	0.97	11.30	25.11
4 G-C	-0.45	-0.14	3.39	-1.59	1.37	37.50
5 A-T	0.17	-0.33	3.30	-0.33	0.46	37.52
6 A-T	-0.01	-0.60	3.22	-0.31	-2.67	32.40
7 T-A	-0.08	-0.40	3.22	1.68	-0.97	33.74
8 T-A	-0.27	-0.23	3.47	0.68	-1.69	42.14
9 C-G	0.70	0.78	3.07	-3.66	4.18	26.58
10 G-C	-1.31	0.36	3.37	-2.85	-9.37	41.60
11 C-G	-0.31	0.21	3.17	-0.68	6.69	33.31
12 G-C	----	----	----	----	----	----

Local base-pair helical parameters

bp	X-disp	Y-disp	h-Rise	Incl.	Tip	h-Twist
1 C-G	-1.27	-0.65	3.09	14.77	5.57	33.94
2 G-C	1.79	-0.36	3.52	-11.95	-3.76	44.84
3 C-G	-1.40	0.52	2.98	24.46	-2.09	27.52
4 G-C	-0.40	0.49	3.40	2.13	2.46	37.56
5 A-T	-0.57	-0.31	3.29	0.71	0.51	37.52
6 A-T	-0.61	-0.03	3.26	-4.78	0.56	32.51
7 T-A	-0.53	0.41	3.22	-1.68	-2.89	33.80
8 T-A	-0.13	0.45	3.47	-2.34	-0.95	42.17
9 C-G	0.66	-2.36	3.03	8.97	7.84	27.14
10 G-C	1.47	1.50	3.30	-12.97	3.95	42.69
11 C-G	-0.69	0.42	3.16	11.53	1.17	33.96
12 G-C	----	----	----	----	----	----

The 'simple' parameters are intuitive for non-Watson-Crick base pairs

and associated base-pair steps (where the above corresponding 3DNA
parameters often appear cryptic). Note that they are for structural
description only, not to be fed into the DSSR rebuild module.

Simple base-pair parameters based on RC8--YC6 vectors

bp	Shear	Stretch	Stagger	Buckle	Propeller	Opening
1 C-G	0.28	-0.13	0.07	7.30	-17.16	-0.60
2 G-C	-0.24	-0.18	0.49	9.00	-14.52	-2.07
3 C-G	0.25	-0.17	0.16	-4.32	-5.50	0.43
4 G-C	-0.26	-0.11	0.01	10.56	-9.72	1.01
5 A-T	-0.04	-0.11	0.01	4.72	-15.31	1.59
6 A-T	0.05	-0.05	0.07	0.61	-15.00	6.18
7 T-A	-0.04	-0.12	0.17	-0.41	-16.74	3.89
8 T-A	-0.11	-0.12	-0.00	-1.82	-16.34	5.07
9 C-G	0.21	-0.12	0.00	-12.17	-10.55	-1.23
10 G-C	-0.11	-0.05	0.24	4.13	-9.63	3.20
11 C-G	0.16	-0.13	0.21	0.47	-17.41	-1.73
12 G-C	-0.25	-0.06	0.25	4.56	-5.05	-1.62

Simple step parameters based on consecutive C1'-C1' vectors

bp	Shift	Slide	Rise	Tilt	Roll	Twist
1 C-G	0.09	0.04	3.20	-3.18	8.53	35.82
2 G-C	0.50	0.67	3.69	2.82	-9.06	40.61
3 C-G	-0.14	0.59	3.00	1.01	11.30	28.45
4 G-C	-0.46	-0.13	3.39	-1.56	1.40	36.11
5 A-T	0.17	-0.33	3.30	-0.33	0.46	36.73

6 A-T	-0.01	-0.60	3.22	-0.31	-2.68	32.91
7 T-A	-0.08	-0.40	3.22	1.67	-0.98	34.10
8 T-A	-0.26	-0.23	3.47	0.70	-1.68	39.85
9 C-G	0.69	0.78	3.07	-3.69	4.15	29.16
10 G-C	-1.31	0.36	3.37	-2.84	-9.37	39.83
11 C-G	-0.31	0.21	3.17	-0.64	6.70	35.85
12 G-C	----	----	----	----	----	----

Simple helical parameters based on consecutive C1'-C1' vectors

bp	X-disp	Y-disp	h-Rise	Incl.	Tip	h-Twist
1 C-G	-1.07	-0.56	3.11	13.60	5.07	36.93
2 G-C	2.00	-0.37	3.50	-12.85	-4.00	41.66
3 C-G	-1.01	0.45	3.01	21.91	-1.95	30.58
4 G-C	-0.41	0.51	3.40	2.26	2.51	36.16
5 A-T	-0.58	-0.32	3.29	0.73	0.52	36.74
6 A-T	-0.61	-0.03	3.26	-4.71	0.54	33.01
7 T-A	-0.52	0.40	3.22	-1.68	-2.85	34.15
8 T-A	-0.13	0.47	3.47	-2.46	-1.03	39.89
9 C-G	0.70	-2.09	3.04	8.15	7.24	29.68
10 G-C	1.57	1.55	3.29	-13.51	4.10	40.97
11 C-G	-0.57	0.41	3.17	10.76	1.03	36.46
12 G-C	----	----	----	----	----	----

Classification of dinucleotide steps in right-handed nucleic acid structures as A-like, B-like, TA-like, or other. For definition of the A-B index (ABI), see Waters et al. (2016). "Transitions of Double-Stranded DNA Between the A- and B-Forms." J. Phys. Chem. B, v120, p8449.

bp	Xp	Yp	Zp	XpH	YpH	ZpH	Form	ABI
1 C-G	-2.24	8.78	0.37	-3.51	8.41	2.60	B	0.72
2 G-C	-2.27	8.89	0.20	-0.59	8.76	-1.53	B	0.83
3 C-G	-3.22	9.14	-0.51	-4.60	8.57	3.24	B	1.01
4 G-C	-3.17	8.84	-0.33	-3.50	8.85	-0.03	B	0.95
5 A-T	-3.43	8.95	-0.40	-3.97	8.95	-0.30	B	0.95
6 A-T	-3.45	9.06	-0.20	-4.03	9.01	-0.92	B	0.90
7 T-A	-3.53	9.07	-0.65	-4.04	9.05	-0.91	B	0.98
8 T-A	-2.74	8.65	0.14	-2.85	8.65	-0.19	B	0.87
9 C-G	-2.88	8.85	-0.43	-2.37	8.81	1.05	B	1.00
10 G-C	-2.59	9.03	-0.04	-1.21	8.82	-1.93	B	0.90
11 C-G	-2.96	8.99	-0.97	-3.62	9.01	0.77	B	----
12 G-C	----	----	----	----	----	----	----	----

Minor and major groove widths: direct P-P distances, and 'Refined' P-P distances which take into account the directions of sugar-phosphate backbones. Subtract 5.8 Angstrom from the values to account for the vdw radii of the phosphate groups, and for comparison with Curves+.

Ref: the Appendix titled "Calculation of Major- and Minor-groove Widths" in the paper "Two Distinct Modes of Protein-induced Bending in DNA." by El Hassan and Calladine (1998) in J. Mol. Biol., v282, p331.

bp	Minor Groove		Major Groove	
	P-P	Refined	P-P	Refined
1 C-G	----	----	----	----
2 G-C	----	----	----	----
3 C-G	14.33	----	18.20	----
4 G-C	12.02	11.94	17.34	17.28
5 A-T	9.68	9.62	19.11	18.96
6 A-T	9.08	9.07	16.41	15.99
7 T-A	9.09	9.08	17.77	17.73
8 T-A	9.61	9.55	18.62	18.59
9 C-G	10.44	----	18.95	----
10 G-C	----	----	----	----

```

11 C-G      ----      ----      ----      ----
12 G-C      ----      ----      ----      ----

*****
Global linear helical axis defined by equivalent C1' and RN9/YN1 atom pairs
helical-rise: 3.30(0.52)
helical-radius: 9.42(0.82)
helical-axis: -0.127   -0.275   -0.953
  point-one: 17.536   25.713   25.665
  point-two: 12.911   15.677   -9.080

```

With the `--more` option, the DSSR base-pair classifications are also available. An example is shown below for the local base-pair parameters of PDB entry 1msy (Correll *et al.*, 2003).

Local base-pair parameters														
bp	Shear	Stretch	Stagger	Buckle	Propeller	Opening	# nt1	nt2	Saenger	LW	DSSR	name		
1 U-G	-1.59	0.32	-0.26	1.95	-3.60	-43.73	# A.U2647	A.G2673	--	cWW	cW-W	--		
2 G-U	-2.37	-0.60	0.11	4.67	-7.75	-2.95	# A.G2648	A.U2672	28-XXVIII	cWW	cW-W	Wobble		
3 C-G	0.09	-0.17	0.02	9.60	-15.93	-2.31	# A.C2649	A.G2671	19-XIX	cWW	cW-W	WC		
4 U-A	0.11	-0.11	0.25	4.92	-16.45	6.33	# A.U2650	A.A2670	20-XX	cWW	cW-W	WC		
5 C-G	0.19	-0.10	-0.07	1.46	-17.75	3.27	# A.C2651	A.G2669	19-XIX	cWW	cW-W	WC		
6 C-G	0.28	-0.11	0.26	-4.32	-8.63	0.53	# A.C2652	A.G2668	19-XIX	cWW	cW-W	WC		
7 U-C	5.75	-2.19	-0.25	-0.40	-12.06	-15.31	# A.U2653	A.C2667	--	tW.	tW-	--		
8 A+C	-5.79	7.40	0.73	-26.65	0.96	-136.94	# A.A2654	A.C2666	--	tHH	tM+M	--		
9 U-A	4.04	-1.62	-0.50	8.42	-17.42	-106.05	# A.U2656	A.A2665	24-XXIV	tWH	tW-M	rHoogsteen		
10 A-G	-6.87	-4.21	-0.06	-1.13	2.56	-0.46	# A.A2657	A.G2664	11-XI	tHS	tM-m	Sheared		
11 C-G	0.12	-0.11	-0.29	6.74	-3.96	1.14	# A.C2658	A.G2663	19-XIX	cWW	cW-W	WC		
12 G-A	6.79	-4.82	1.19	-0.98	-18.47	-11.76	# A.G2659	A.A2662	11-XI	tSH	tM-M	Sheared		

The above command also produces the following two auxiliary files:

- File `dssr-pairFrames.txt` contains a list of base-pair reference frames.
- File `dssr-pairs.txt` presents a list of base pairs used in the analysis. The default name of the output file can be changed via `--duplex-outfile`. The list can be manually edited and then fed back to DSSR via `--duplex-inpfile`. Thus, the `analyze` module has the variations of `--pair-list` documented above (see Section 3.9). Just as the `find_pair` and `analyze` combo in 3DNA, the DSSR `analyze` module is flexible: it allows users to have full control in characterizing desired pairs.

With option `--rebuild-parameters` (or just `rebuild`), two additional files would be generated: `dssr-dsStepPars.txt` and `dssr-dsHeliPars.txt`. They can be used with the DSSR `rebuild` module (see Section 5.3) to create customized duplex models, using local step or helical parameters, respectively (see Section 3.18.1). The contents of `dssr-dsStepPars.txt` is shown below.

```

# 12 (no. of base pairs)
#bp      Shear      Stretch     Stagger     Buckle     Propeller     Opening      Shift      Slide      Rise      Tilt      Roll      Twist
C-G      0.2760    -0.1398    0.0730     6.9298    -17.3081    -0.6058     0.0866    0.0385     3.1999    -3.2156    8.5199    32.7307

```

G-C	-0.2363	-0.1819	0.4909	9.3411	-14.3021	-2.0765	0.4964	0.6682	3.6907	2.8468	-9.0554	43.8789
C-G	0.2444	-0.1712	0.1587	-4.4328	-5.4046	0.4341	-0.1383	0.5933	2.9998	0.9666	11.2997	25.1144
G-C	-0.2551	-0.1148	0.0080	10.8059	-9.4487	1.0121	-0.4531	-0.1393	3.3878	-1.5850	1.3726	37.4997
A-T	-0.0371	-0.1071	0.0108	4.7238	-15.3137	1.6013	0.1713	-0.3250	3.2981	-0.3304	0.4588	37.5204
A-T	0.0514	-0.0495	0.0653	0.4421	-15.0014	6.2330	-0.0113	-0.6010	3.2186	-0.3111	-2.6749	32.4028
T-A	-0.0366	-0.1165	0.1734	-0.2639	-16.7436	3.9280	-0.0817	-0.3973	3.2156	1.6811	-0.9740	33.7441
T-A	-0.1077	-0.1204	-0.0040	-1.5612	-16.3643	5.1200	-0.2673	-0.2256	3.4655	0.6845	-1.6863	42.1361
C-G	0.2092	-0.1271	0.0030	-12.4104	-10.2744	-1.2229	0.7001	0.7758	3.0685	-3.6563	4.1800	26.5812
G-C	-0.1064	-0.0529	0.2385	4.2054	-9.6032	3.2064	-1.3106	0.3601	3.3708	-2.8525	-9.3681	41.6009
C-G	0.1561	-0.1302	0.2078	0.2832	-17.4174	-1.7549	-0.3093	0.2112	3.1742	-0.6786	6.6923	33.3104
G-C	-0.2444	-0.0691	0.2539	4.6712	-4.9538	-1.6200	999999	999999	999999	999999	999999	999999

3.18.2 Analysis of single-stranded structures

With `--single-strand` (or `--ss`), DSSR would take the input coordinate file as a ‘pseudo’ single-strand, no matter how many chains it contains or if any chain has broken backbones. PDB entry 355d is the classic B-DNA dodecamer duplex (Shui *et al.*, 1998), with base sequence CGCGAATTCTCGCG in both chains A and B. The following result would be produced with the command: `x3dna-dssr analyze --single-strand -i=355d.pdb`

```
num_nts=24
```

```
*****
local base-step parameters
base Shift Slide Rise Tilt Roll Twist
 1 C 0.35 0.26 3.39 2.21 12.27 31.38 # A.DC1
 2 G 1.11 0.69 3.42 0.16 -4.01 45.02 # A.DG2
 3 C -0.20 0.71 2.92 10.04 9.96 24.80 # A.DC3
 4 G 0.06 0.06 3.41 -0.47 1.10 37.82 # A.DG4
 5 A 0.66 -0.25 3.27 2.56 1.35 39.67 # A.DA5
 6 A 0.38 -0.64 3.25 3.61 -3.65 31.03 # A.DA6
 7 T 0.34 -0.46 3.10 5.79 -1.29 33.80 # A.DT7
 8 T 0.35 -0.45 3.43 0.38 -1.36 38.80 # A.DT8
 9 C 0.79 0.71 3.17 6.75 3.50 28.83 # A.DC9
10 G -0.77 0.45 3.46 0.06 -12.21 39.61 # A.DG10
11 C -0.18 0.24 3.23 4.42 13.44 33.15 # A.DC11
12 G 0.12 -0.25 -0.23 173.17 2.45 -93.36 # A.DG12
13 C 0.43 0.18 3.12 5.97 0.10 32.97 # B.DC13
14 G 1.83 0.27 3.20 5.73 -6.54 43.34 # B.DG14
15 C -0.62 0.83 2.97 14.04 4.67 23.60 # B.DC15
16 G 0.89 -0.03 3.39 -0.98 -1.99 45.53 # B.DG16
17 A 0.53 -0.37 3.27 2.47 -0.99 33.10 # B.DA17
18 A 0.39 -0.60 3.14 4.30 -2.06 33.23 # B.DA18
19 T 0.31 -0.43 3.25 3.15 -0.63 34.97 # B.DT19
20 T 0.96 -0.35 3.28 2.70 1.63 37.08 # B.DT20
21 C 0.09 0.47 3.09 8.14 12.53 25.02 # B.DC21
22 G 0.16 0.65 3.87 -5.43 -14.14 42.99 # B.DG22
23 C 0.12 -0.15 2.99 8.75 4.90 33.12 # B.DC23
24 G ---- ---- ---- ---- ---- ---- # B.DG24

*****
local base-step helical parameters
base X-disp Y-disp h-Rise Incl. Tip h-Twist
 1 C -1.66 -0.22 3.27 21.64 -3.89 33.71 # A.DC1
 2 G 1.27 -1.43 3.36 -5.23 -0.21 45.19 # A.DG2
 3 C -0.66 2.52 2.72 21.14 -21.31 28.50 # A.DC3
```

4 G	-0.05	-0.16	3.41	1.69	0.73	37.84	# A.DG4
5 A	-0.52	-0.68	3.30	1.99	-3.76	39.77	# A.DA5
6 A	-0.49	-0.01	3.32	-6.77	-6.69	31.44	# A.DA6
7 T	-0.59	0.30	3.13	-2.19	-9.86	34.31	# A.DT7
8 T	-0.50	-0.48	3.45	-2.04	-0.57	38.82	# A.DT8
9 C	0.64	-0.12	3.33	6.89	-13.26	29.79	# A.DC9
10 G	2.00	1.10	3.19	-17.53	-0.08	41.38	# A.DG10
11 C	-1.53	0.92	3.05	22.33	-7.34	35.97	# A.DC11
12 G	0.13	-0.03	-0.24	-1.24	87.51	-175.33	# A.DG12
13 C	0.30	0.20	3.15	0.18	-10.42	33.49	# B.DC13
14 G	0.97	-1.90	3.33	-8.75	-7.66	44.16	# B.DG14
15 C	0.74	4.27	2.36	10.24	-30.75	27.80	# B.DC15
16 G	0.14	-1.24	3.36	-2.57	1.26	45.58	# B.DG16
17 A	-0.48	-0.51	3.30	-1.73	-4.33	33.20	# B.DA17
18 A	-0.71	0.02	3.19	-3.57	-7.47	33.56	# B.DA18
19 T	-0.63	-0.04	3.27	-1.04	-5.22	35.12	# B.DT19
20 T	-0.77	-1.14	3.32	2.55	-4.23	37.21	# B.DT20
21 C	-1.77	1.59	2.89	26.16	-16.99	29.09	# B.DC21
22 G	2.28	-0.76	3.45	-18.62	7.15	45.46	# B.DG22
23 C	-0.94	1.03	2.88	8.37	-14.92	34.56	# B.DC23
24 G	----	----	----	----	----	----	# B.DG24

DSSR treats A.DG12/B.DC13 as a ‘pseudo’ step, and calculates local step and helical parameters for it just as for any covalently bonded steps. This A.DG12/B.DC13 step possesses ‘weird’ local base-step parameters: `rise=-0.23`, `tilt=173.17` and `twist=-93.36`. Nevertheless, all these parameters are required for a rigorous rebuilding of the base geometry (Lu and Olson, 2003; Li *et al.*, 2019). See Section 5.3. The above command also produces two auxiliary files: `dssr-baseFrames.txt` consisting of base reference frames, and `dssr-nts.txt` listing nucleotides identified.

With options `--ss --rebuild`, files `dssr-ssStepPars.txt` and `dssr-ssHeliPars.txt` would be generated. They can be used with the DSSR `rebuild` module (see Section 5.3) to create customized models, using local step or helical parameters, respectively (cf. the main output listing on Page 78). The contents of `dssr-ssStepPars.txt` for PDB entry 355d is shown below.

# 24 (no. of nucleotides)							
#	Shift	Slide	Rise	Tilt	Roll	Twist	
C	0.3454	0.2569	3.3869	2.2059	12.2655	31.3789	# 1 A.DC1
G	1.1089	0.6932	3.4241	0.1624	-4.0109	45.0168	# 2 A.DG2
C	-0.1963	0.7102	2.9169	10.0404	9.9618	24.8025	# 3 A.DC3
G	0.0641	0.0633	3.4100	-0.4701	1.0985	37.8198	# 4 A.DG4
A	0.6641	-0.2491	3.2712	2.5585	1.3498	39.6730	# 5 A.DA5
A	0.3760	-0.6388	3.2502	3.6106	-3.6509	31.0272	# 6 A.DA6
T	0.3367	-0.4604	3.1019	5.7879	-1.2854	33.8048	# 7 A.DT7
T	0.3511	-0.4461	3.4299	0.3762	-1.3568	38.7955	# 8 A.DT8
C	0.7921	0.7089	3.1700	6.7457	3.5041	28.8298	# 9 A.DC9
G	-0.7738	0.4495	3.4609	0.0555	-12.2148	39.6093	# 10 A.DG10
C	-0.1795	0.2431	3.2278	4.4154	13.4407	33.1518	# 11 A.DC11
G	0.1210	-0.2522	-0.2252	173.1738	2.4484	-93.3637	# 12 A.DG12
C	0.4332	0.1839	3.1219	5.9737	0.1038	32.9669	# 13 B.DC13
G	1.8331	0.2671	3.2010	5.7276	-6.5423	43.3366	# 14 B.DG14

C	-0.6204	0.8339	2.9705	14.0411	4.6730	23.6037	#	15	B.DC15
G	0.8896	-0.0279	3.3852	-0.9789	-1.9929	45.5329	#	16	B.DG16
A	0.5276	-0.3678	3.2659	2.4689	-0.9887	33.0978	#	17	B.DA17
A	0.3856	-0.5980	3.1354	4.2977	-2.0561	33.2338	#	18	B.DA18
T	0.3099	-0.4338	3.2481	3.1477	-0.6259	34.9740	#	19	B.DT19
T	0.9581	-0.3486	3.2817	2.6977	1.6269	37.0782	#	20	B.DT20
C	0.0860	0.4713	3.0910	8.1394	12.5346	25.0236	#	21	B.DC21
G	0.1565	0.6513	3.8651	-5.4327	-14.1402	42.9926	#	22	B.DG22
C	0.1242	-0.1466	2.9931	8.7466	4.9035	33.1161	#	23	B.DC23
G	999999	999999	999999	999999	999999	999999	#	24	B.DG24

4 Visualization features

By design, DSSR can be easily incorporated into other structural bioinformatics pipelines/resources (Sagendorf *et al.*, 2020; Lawson *et al.*, 2024; Feng *et al.*, 2024). Over the years, I have collaborated with Robert Hanson and Thomas Holder to connect DSSR to Jmol and PyMOL, respectively. These integrations exemplify the critical roles that a domain-specific analysis engine may play in general-purpose molecular visualization tools.

4.1 DSSR-Jmol integration

The DSSR-Jmol paper (Hanson and Lu, 2017), “DSSR-enhanced visualization of nucleic acid structures in Jmol”, introduces a flexible, powerful SQL-like query language that employs the standard JSON interface between the two programs. Users can now select DSSR-derived structural features (such as base pairs, double helices, and various loops) as easily as they can select protein α -helices and β -sheets. Moreover, fine-grained characteristics of RNA structural features can be queried. Here are some simple examples:

```
SELECT WITHIN(dssr, "nts WHERE is_modified = true") # modified nucleotides
SELECT pairs # all pairs
Select WITHIN(dssr, "pairs WHERE name = 'Hoogsteen'") # Hoogsteen pairs
SELECT WITHIN(dssr, "pairs WHERE name != 'WC'") # non-Watson-Crick pairs
SELECT junctions # all junctions loops
select within(dssr, "junctions WHERE num_stems = 4") # four-way junction loops
```

The “DSSR-Jmol integration” section on the 3DNA Forum contains scripts and data files for reproducing reported results, including the graphical abstract and the cover image (see Figure 23).

Overall, the DSSR-Jmol integration makes salient features of DSSR readily accessible via Jmol/JSmol, as demonstrated at the website <http://jmol.x3dna.org>. This work fills a gap

in RNA structural bioinformatics. It enables deep analyses and SQL-like queries of RNA structural characteristics, interactively.

4.2 DSSR-PyMOL integration

The DSSR-PyMOL paper (Lu, 2020), “DSSR-enabled innovative schematics of 3D nucleic acid structures with PyMOL”, presents schematic block representations in diverse styles. These DSSR blocks can be seamlessly integrated into PyMOL and complement its other popular visualization options. In addition to portraying individual base blocks, DSSR can draw WC pairs as long blocks and highlight the minor-groove edges. Notably, DSSR can dramatically simplify the depiction of G-quadruplexes by automatically detecting G-tetrads and treating them as large square blocks (see Figures 18 and 19). The DSSR-enabled innovative schematics with PyMOL are aesthetically pleasing and highly informative; the base identity, pairing geometry, stacking interactions, double-helical stems, and G-quadruplexes are immediately obvious (see Figures 3, 18 and 19 and the cover image).

```
http://skmatic.x3dna.org/pdb/2lx1 # with internal loop 5'GAGU-3'UGAG
http://skmatic.x3dna.org/pdb/2grb # an RNA quadruplex containing inosine-tetrad
http://skmatic.x3dna.org/pdb/6vu1 # an RNA hairpin
http://skmatic.x3dna.org/pdb_entries # 12 randomly picked entries
```

These features can be accessed via four interfaces: the command-line interface (CLI), the DSSR plugin for PyMOL, the web application (skmatic.x3dna.org), and the web application programming interface (API). Taken together, those interfaces cover virtually all conceivable use cases. The easiest way to get started and benefit quickly from DSSR-PyMOL is via the web application (skmatic.x3dna.org), which also provides pre-calculated schematics and meta information of nucleic-acid-containing structures in the PDB. Some examples are provided in the above listing.

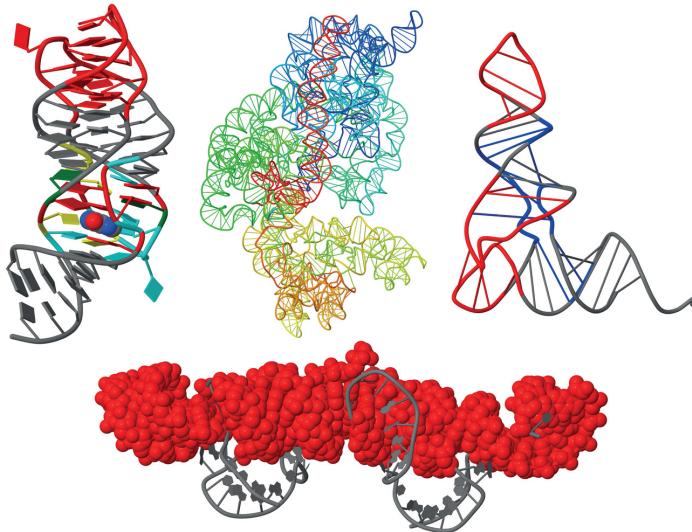
The supplemental PDF of Lu (2020) serves as a practical guide, with complete and reproducible examples. It provides detailed descriptions for five options: (i) `--block-file`, (ii) `--block-color`, (iii) `--block-depth`, (iv) `--cartoon-block`, and (v) `--blocview`. Moreover, the `--block-file` option contains more features than those documented therein, including `major`, `up`, `down`, `equal`, `thin`, and `thick`.

The `major` keyword (`--block-file=major`) highlights the major-groove edge, in contrast to `minor`. It is pair-sensitive when used with `wc`, i.e., `--block-file=major-wc`. The `up` and `down` keywords refer to the top/bottom (or positive/negative) faces of rectangular blocks.

PRINT ISSN: 0305-1048
ONLINE ISSN: 1362-4962

Nucleic Acids Research

VOLUME 45 WEB SERVER ISSUE JULY 3, 2017
<https://academic.oup.com/nar>



OXFORD
UNIVERSITY PRESS

Open Access

No barriers to access – all articles freely available online



Figure 23: 3D interactive visualization of selected RNA structural features enabled by the DSSR-Jmol integration (<http://jmol.x3dna.org>). Clockwise from upper left: Structure of the xpt-pbuX guanine riboswitch in complex with hypoxanthine (PDB entry 4fe5) in ‘base blocks’ representation. The three-way junction loop encompassing the metabolite (in space-filling representation) is color-coded by base identity: A, red; C, yellow; G, green; U, cyan. The loop-loop interaction at the top is highlighted in red. Structure of the *T. thermophilus* 30S ribosomal subunit in complex with antibiotics (PDB entry 1fjg) in step diagram. The 16S rRNA is color-coded in spectrum with the 5'-end in blue and the 3'-end in red (upper middle). Structure of the classic L-shaped yeast tRNA^{Phe} (PDB entry 1ehz) in step diagram, with the three hairpin loops highlighted in red and the four-way junction loop in blue (upper right corner). Structure of the Pistol self-cleaving ribozyme (PDB entry 5ktj), showcasing (in red) the horizontal helix in space-filling representation. The helix is composed of six short stems stabilized via coaxial stacking interactions (bottom, see also Figure 19C).

Their effects are most easily noticed for WC pairs (the M–N type), where the two bases are in opposite orientations. They can also be used with `fill` in molecular representations, i.e., `--block-file=fill-up`. In fact, Figures 4 and 8 were generated using this feature. The `equal` attribute produces purine and pyrimidine blocks of the same size, half that of the WC-pair block (Figure 18A). Finally, `thin` and `thick` set the thickness of blocks to 0.2 Å and 0.8 Å, respectively. They serve as shorthands of using the `--block-depth` option.

The website <http://skmatic.x3dna.org> has been recommended by the Faculty Opinions as “simple and effective”, and classified as “Good for Teaching”.

4.3 Visualization-related options

4.3.1 The `--hbfile-pymol` option

The `--hbfile-pymol` option is intended to draw molecular images using PyMOL commands, with DSSR-detected H-bonding information. Figures 1, 3, 4, and 7 in the supplemental PDF of Lu *et al.* (2015) were generated using this approach. The listing below provides detailed steps to create Figure 24, using various DSSR options.

```
x3dna-dssr -i=1ehz.pdb -o=1ehz.out    # generate dssr-multiplets.pdb
x3dna-dssr -i=dssr-multiplets.pdb --select-model=2 -o=1ehz-AUA.pdb  # extract model no.2

x3dna-dssr -i=1ehz-AUA.pdb --hbfile-pymol -o=1ehz-AUA.pml  # "load 1ehz-AUA.pml" in PyMOL

x3dna-dssr -i=1ehz-AUA.pdb --hbfile-jmol -o=1ehz-AUA.scr  # "script 1ehz-AUA.scr" in Jmol
```

4.3.2 The `--hbfile-jmol` option

In parallel to `--hbfile-pymol`, the `--hbfile-jmol` option can be used to draw molecular images in Jmol, with DSSR-derived H-bond information (Figure 24B). See the listing in Section 4.3.1.

4.3.3 The `--helical-axis` option

The `--helical-axis` option introduces an auxiliary file, named `dssr-helicalAxes.pdb` by default. It contains least-squares fitted linear helical axes for all helices that DSSR detects. The file is in PDB format that can be fed into a molecular viewer (e.g., PyMOL or Jmol) for visualization.

Using tRNA^{Phe} 1ehz as an example, the DSSR command and the resultant PDB file are

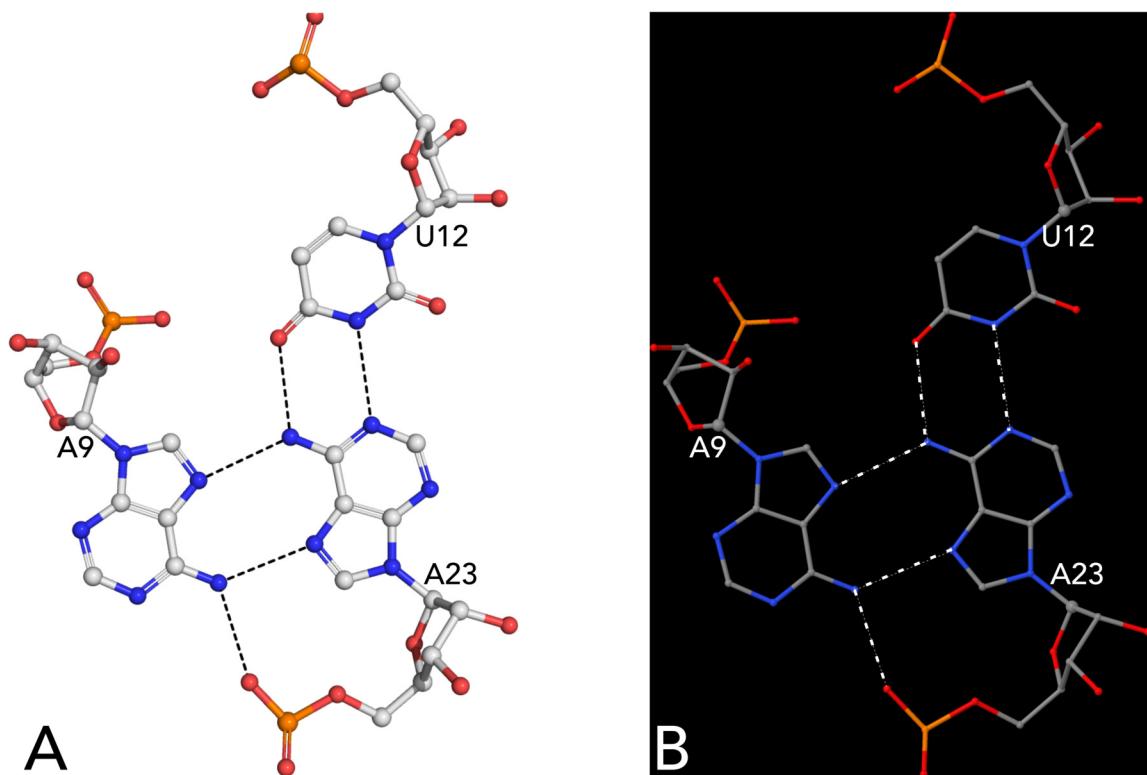


Figure 24: Molecular images with DSSR-detected H-bonds. The AUA base triplet (A9, U12, and A23) was extracted from PDB entry 1ehz (tRNA^{Phe}). (A) Image rendered in PyMOL, using the `--hbfile-pymol` option. (B) Image rendered in Jmol, using the `--hbfile-jmol` option. This figure was annotated using Inkscape (<https://inkscape.org>).

shown below.

```
# x3dna-dssr -i=1ehz.pdb --helical-axis

REMARK-DSSR: helix#1
ATOM      1  P1      G A    1       52.699   41.360   47.877   1.00  99.85      H1     P
REMARK-DSSR: helix#1
ATOM      2  P2      C A   56      78.559   72.327   36.597   1.00  37.81      H1     P
REMARK-DSSR: helix#2
ATOM      3  P1      A A   36      68.999   35.486   -4.554   1.00  81.67      H2     P
REMARK-DSSR: helix#2
HETATM    4  P2      H2U A   16      72.025   54.504   33.741   1.00  64.01      H2     P
CONECT    1      2
CONECT    2      1
CONECT    3      4
CONECT    4      3
```

4.3.4 The `--simple-junction` option

This option leads to the creation of an auxiliary file, named `dssr-simplifiedJcts.pdb` by default. It contains a simplified representation of junction loops that DSSR detects. The file is in PDB format consisting of C1' atoms of consecutive nucleotides in a junction.

Using tRNA^{Phe} 1ehz as an example, the DSSR command and the resultant PDB file are shown below.

```
# x3dna-dssr -i=1ehz.pdb --simple-junction

MODEL      1
REMARK  model=1 nts=16
REMARK  4-way junction: nts=16; [2,1,5,0]; linked by [#1,#2,#3,#4]
ATOM    1  C1'   U A   7     63.015  49.475  35.442  1.00 37.23      C
ATOM    2  C1'   U A   8     68.407  51.830  29.255  1.00 30.28      C
ATOM    3  C1'   A A   9     65.985  50.969  21.571  1.00 28.79      C
HETATM   4  C1'  2MG A  10    60.152  47.173  19.012  1.00 44.62      C
ATOM    5  C1'   C A   25    67.844  39.818  17.685  1.00 51.93      C
HETATM   6  C1'  M2G A  26    64.147  42.470  14.901  1.00 46.92      C
ATOM    7  C1'   C A   27    63.159  46.064  10.787  1.00 48.68      C
ATOM    8  C1'   G A   43    71.295  52.730  8.219   1.00 46.94      C
ATOM    9  C1'   A A   44    67.208  53.996  10.887  1.00 54.14      C
ATOM   10  C1'   G A   45    65.456  53.277  15.954  1.00 45.24      C
HETATM  11  C1'  7MG A  46    65.949  56.384  22.900  1.00 39.69      C
ATOM   12  C1'   U A   47    60.798  60.075  28.167  1.00 50.55      C
ATOM   13  C1'   C A   48    68.291  55.799  30.245  1.00 27.98      C
HETATM  14  C1'  5MC A  49    60.568  54.849  34.825  1.00 33.10      C
ATOM   15  C1'   G A   65    57.142  59.262  44.032  1.00 42.23      C
ATOM   16  C1'   A A   66    55.595  55.117  40.311  1.00 40.50      C
CONECT  1    16   2
CONECT  2    1    3
CONECT  3    2    4
CONECT  4    3    5
CONECT  5    4    6
CONECT  6    5    7
CONECT  7    6    8
CONECT  8    7    9
CONECT  9    8   10
CONECT 10   9   11
CONECT 11   10   12
CONECT 12   11   13
CONECT 13   12   14
CONECT 14   13   15
CONECT 15   14   16
CONECT 16   15   1
ENDDML
```

5 Modeling capabilities

DSSR comes with three modeling modules: (i) `mutate` for *in silico* base mutations, (ii) `fiber` for generating regular helical models, and (iii) `rebuild` for creating customized

structures. They replace the 3DNA `mutate_bases`, `fiber`, and `rebuild` programs, with considerably improved usability.

The modeling functionality is activated via `x3dna-dssr mutate|fiber|rebuild`. The module name must be specified directly after the DSSR executable, without leading dashes. The modeling modules are introduced into DSSR to make it feature complete and convenient to use, thus replacing 3DNA completely.

5.1 *In silico* mutations (`mutate`)

5.1.1 Introduction

3DNA contains the `mutate_bases` program that performs *in silico* base mutations (Li *et al.*, 2019). Notably, the mutation process preserves both the geometry of the backbone and the base reference frame (Olson *et al.*, 2001). As a result, re-analyzing the mutated variant gives the same base-pair parameters as the original structure.

The `mutate_bases` program was created upon request from the 3DNA user community. Over time, it has become popular. Searching `mutate_bases` on the 3DNA Forum would lead to hits from more than 40 threads with over 100 posts. Notably, the program has been (explicitly) cited in literature, including the following three notable cases:

- AlQuraishi and McAdams (2013) in *Proteins*: “For each base position, *in silico* structural mutants are generated using 3DNA to mutate the basepair to include all four possibilities.”
- Howe *et al.* (2015) in *Nature*: “Homology modelling. A homology model of the *E. coli* FMN aptamer was constructed using program `mutate_bases` of the 3DNA package using the *F. nucleatum* impX riboswitch aptamer X-ray structure as the template and the FMN aptamer alignment of *E. coli*, *F. nucleatum*, *P. aeruginosa* and *A. baumannii*.”
- Wang *et al.* (2017) in *Cell Chemical Biology*: “Homology Modeling: The homology models of both *S. aureus* SA1 and SA2 FMN aptamers were constructed using program `mutate_bases` (Lu and Olson, 2003) of the 3DNA package, with the *F. nucleatum* riboswitch aptamer X-ray structure as the template.”

While the 3DNA `mutate_bases` program has been successful, it lacks some desirable features. Shown below is the gist of the post on the 3DNA Forum that started the thread “`Mutate_Bases`: Option to mutate all residues of the same type to another”, dated April 20,

2012.

“I am playing with mutating 1RNA.pdb (contains only A/U) into a structure that contains only C/G (G for A and C for U). This is not vital but it would be nice if there was an option to convert ALL of the A’s to G’s (or whatever is needed) and all of the U’s to C’s. The current functionality is perfectly fine for a few point mutations but this added capability would be helpful and make the process less tedious having to create a file to list out all of the individual mutations. I can also see a logical negation to be useful as well. For example, mutate all of the A’s to G’s except for residue 5 and residue 7.”

The above requests, and some more advanced features, have been in mind for a long while. However, they cannot be easily implemented in the (dated) 3DNA codebase. That is where DSSR comes to the rescue. DSSR includes a new module for *in silico* base mutations that are context sensitive and extremely flexible. It serves as a drop-in replacement of the 3DNA `mutate_bases` program, with additional advanced features.

This template-based homology modeling feature is enabled by the DSSR `mutate` sub-command, which must be specified right after `x3dna-dssr` and without any leading dashes (i.e., `x3nda-dssr mutate`). The whole command goes like this:

```
x3dna-dssr mutate -i=coordinate-file --enum|--list|--entry [-o=output-file]
```

The sections below document the three options for specifying mutations: `--enum`, `--list`, and `--entry`. The combination of `--enum` and `--list` allows for full user-control over the mutation process. The `--entry` option, on the other hand, offers great flexibility in common use cases. Taken together, these options cover a wide range of real-world applications.

5.1.2 The `--enum` option

The `--enum` option enumerates all nucleotides in a given coordinate file. The output can be easily modified as appropriate, and then fed into `x3nda-dssr mutate` via the `--list` option (see below).

Using PDB entry 1msy (Figure 3) as an example, here is the command:

```
x3dna-dssr mutate -i=1msy.pdb --enum -o=1msy-nts.txt
```

The contents of the resultant file `1msy-nts.txt` are listed below:

```
# For each entry to be mutated, remove the leading #, and add to=base

# To mutate A.A9 to G in PDB entry 1ehz (tRNA):
#       # chain=A name=A num=9  #9  A.A9
#       ---> chain=A name=A num=9 to=G

# To mutate B.DT19 to DC in PDB entry 355d (B-DNA):
#       # chain=B name=DT num=19  #19  B.DT19, pairedNt=A.DA6
#       ---> chain=B name=DT num=19 to=DC

# Empty or comment (starting with #) lines are ignored.

# chain=A name=U num=2647  #1  A.U2647
# chain=A name=G num=2648  #2  A.G2648, pairedNt=A.U2672
# chain=A name=C num=2649  #3  A.C2649, pairedNt=A.G2671
# chain=A name=U num=2650  #4  A.U2650, pairedNt=A.A2670
# chain=A name=C num=2651  #5  A.C2651, pairedNt=A.G2669
# chain=A name=C num=2652  #6  A.C2652, pairedNt=A.G2668
# chain=A name=U num=2653  #7  A.U2653
# chain=A name=A num=2654  #8  A.A2654
# chain=A name=G num=2655  #9  A.G2655
# chain=A name=U num=2656  #10  A.U2656
# chain=A name=A num=2657  #11  A.A2657
# chain=A name=C num=2658  #12  A.C2658, pairedNt=A.G2663
# chain=A name=G num=2659  #13  A.G2659
# chain=A name=U num=2660  #14  A.U2660
# chain=A name=A num=2661  #15  A.A2661
# chain=A name=A num=2662  #16  A.A2662
# chain=A name=G num=2663  #17  A.G2663, pairedNt=A.C2658
# chain=A name=G num=2664  #18  A.G2664
# chain=A name=A num=2665  #19  A.A2665
# chain=A name=C num=2666  #20  A.C2666
# chain=A name=C num=2667  #21  A.C2667
# chain=A name=G num=2668  #22  A.G2668, pairedNt=A.C2652
# chain=A name=G num=2669  #23  A.G2669, pairedNt=A.C2651
# chain=A name=A num=2670  #24  A.A2670, pairedNt=A.U2650
# chain=A name=G num=2671  #25  A.G2671, pairedNt=A.C2649
# chain=A name=U num=2672  #26  A.U2672, pairedNt=A.G2648
# chain=A name=G num=2673  #27  A.G2673
```

For illustration purpose, let's mutate the isolated WC C–G pair (between C2658 and G2663 in 1msy to a U–A pair (see Section 3.2.7 and Figure 3). File `1msy-nts.txt` is edited and simplified to the contents listed below. Since empty and comment lines are ignored, only the 2nd and 5th lines are significant.

```
1 # chain=A name=C num=2658  #12  A.C2658, pairedNt=A.G2663
2 chain=A name=C num=2658 to=U      # C2658 ---> U
3
4 # chain=A name=G num=2663  #17  A.G2663, pairedNt=A.C2658
5 chain=A name=G num=2663 to=A      # G2663 ---> A
```

The mutations list above is saved to a file named `1msy-CG2UA.txt`. This file will be used in the next section, with option `--list`, to carry out base mutations.

5.1.3 The --list option

The `--list` option specifies a file that contains a list of mutations, one entry per line. Using the above `1msy-CG2UA.txt` file as an example, the usage is as below.

```
x3dna-dssr mutate -i=1msy.pdb --list=1msy-CG2UA.txt -o=1msy-CG2UA.pdb
```

Note that file `1msy-nts.txt`, the direct output from `--enum`, is an valid input to `--list`. It does not specify any base mutations since the file contains only empty and comment lines.

By combining `--enum` and `--list`, users can pinpoint the bases to be mutated. The procedure is easy to understand in concept and simple to follow in practice. It covers all use cases.

5.1.4 The --entry option

The `--entry` option offers advanced features with great flexibility. It can be used with the `--cond=filter` option to perform mutations that are sensitive to a structural context.

A mutation entry is defined in the following format. Here the key=value fields are separated by spaces. The fields in square brackets (e.g., `[chain=chain-id]`) are optional, and can be omitted.

```
[chain=chain-id] [name=base-name] [num=residue-number] [skip] to=base-name
```

Examples in the listing below are based on PDB entry 1msy (Figure 3), with accompanying comments. Run these commands, together with a molecular viewer (e.g., PyMOL), to check the results.

```
# mutate all bases to U's
x3dna-dssr mutate -i=1msy.pdb --entry='to=U' -o=1msy-all-Us.pdb

# mutate all bases in the HAIRPIN LOOP to U's
x3dna-dssr mutate -i=1msy.pdb --entry='to=U' cond=hairpin -o=1msy-hairpin-Us.pdb
# same as above, but with loop nucleotides explicitly specified
x3dna-dssr mutate -i=1msy.pdb --entry='num=2659:2662 to=U' -o=1msy-hairpinNts-Us.pdb

# mutate Watson-Crick (WC) pairs from C-G to G-C; entries are separated by semi-colon
x3dna-dssr mutate -i=1msy.pdb --entry='name=C to=G; name=G to=C' --cond=WC -o=1msy-allWC-
→ CG2GC.pdb

# mutate C2658 to U and G2663 to A; same results as with: --list=1msy-CG2UA.txt
x3dna-dssr mutate -i=1msy.pdb --entry='num=2658 to=U; num=2663 to=A' -o=1msy-entry-CG2UA.pdb
```

Coming back to PDB entry 1rna (Dock-Bregeon *et al.*, 1989), here are the commands to fulfill the user requests quoted on Page 87.

```
# mutate all A-->G, and U-->C
x3dna-dssr mutate -i=1rna.pdb --entry='name=A to=G; name=U to=C' -o=1rna-A2G-U2C.pdb
#   the above can be simplified via M:N for M-->N mutations
x3dna-dssr mutate -i=1rna.pdb --entry='A:G; U:C' -o=1rna-A2G-U2C-abbr.pdb

# mutate all A-->G, except residues 5 and 7
x3dna-dssr mutate -i=1rna.pdb --entry='A:G; num=5+7 skip' -o=1rna-A2G-x5x7.pdb
```

5.1.5 The **--mutate-type** option

DSSR v2.5.3 introduced a significant enhancement to the **mutate** sub-command: in addition to base mutations, the **--mutate-type** option now supports mutation of the backbone. Furthermore, the target can be any fragment, regardless of length or composition, rather than just a single nucleotide. When combined with the **rebuild** module, this feature significantly enhances DSSR's ability to model nucleic acid structures.

5.2 Regular helical models (**fiber**)

5.2.1 Introduction

From early on, 3DNA (Lu and Olson, 2003, 2008) contains the **fiber** program that can be used to build models of over fifty types of uniform helical structures. These regular models are based primarily on the fiber diffraction work of Arnott (1999). Overall, they include double-stranded (ds) DNA in various helical types (including A-, B-, C, and Z-form), dsRNA, DNA-RNA hybrids, DNA or RNA triplexes, one single-stranded (ss) RNA, and one RNA quadruplex. Among the list, commonly used A-, B- and C-DNA models (e.g., those derived from calf thymus) can be built from generic base sequence (A, C, G, and T). However, the majority of these models are from pre-defined base sequence. For example, Z-DNA from repeated GC·GC sequence, and the parallel polyI:polyI:polyI:polyI RNA quadruplex from hydrogen-bonded hypoxanthine (I) tetrads.

Over the years, the 3DNA **fiber** program has been improved with an expanded list of models and new features, including: (i) ssRNA with generic base sequence (A, C, G, and U), (ii) the historic Pauling triplex model (Pauling and Corey, 1953), an unusual structure with exposed bases, and (iii) output in mmCIF format for huge models. The **fiber** module in DSSR has surpassed the 3DNA **fiber** program, with vastly improved usability. Its usage is

described in the listing below, with each option documented in the following sections.

```
x3dna-dssr fiber [options] [-o=output-file]

# where options are:
--list          # Generate a list of available models
--model=integer # Specify a model number (1-56)
--model=keyword # Specify a common model by keyword (A-DNA, RNA, Pauling, etc.)
--keyword       # shorthand form for common cases (e.g., --RNA)
--seq=string    # Input base sequence via command line
--seq-file=file # Input base sequence from a file
--repeat=integer # Specify the number of repetition (1)
--mmcif-output  # Set the output format to mmCIF instead of PDB
```

5.2.2 The --list option

With `x3dna-dssr fiber --list`, DSSR generates a descriptive list of 56 models (see below). The citations to the original sources of these models are also provided.

id#	Twist	Rise		Structure	description
	(degree)	(Angstrom)			
1	32.7	2.548	duplex	A-DNA	(calf thymus; generic sequence: A, C, G and T)
2	65.5	5.095	duplex	A-DNA	poly d(ABr5U) : poly d(ABr5U)
3	0.0	28.030	duplex	A-DNA	(calf thymus) poly d(ATCGGAATGGT) : poly d(ACCATTCCGAT)
4	36.0	3.375	duplex	B-DNA	(calf thymus; generic sequence: A, C, G and T)
5	72.0	6.720	duplex	B-DNA	poly d(CG) : poly d(CG)
6	180.0	16.864	duplex	B-DNA	(calf thymus) poly d(CCCCC) : poly d(GGGGG)
7	38.6	3.310	duplex	C-DNA	(calf thymus; generic sequence: A, C, G and T)
8	40.0	3.312	duplex	C-DNA	poly d(GGT) : poly d(ACC)
9	120.0	9.937	duplex	C-DNA	poly d(GGT) : poly d(ACC)
10	80.0	6.467	duplex	C-DNA	poly d(AG) : poly d(CT)
11	80.0	6.467	duplex	C-DNA	poly d(AG) : poly d(CT)
12	45.0	3.013	duplex	D-DNA	poly d(AAT) : poly d(ATT)
13	90.0	6.125	duplex	D-DNA	poly d(CI) : poly d(CI)
14	-90.0	18.500	duplex	D-DNA	poly d(ATATAT) : poly d(ATATAT)
15	-60.0	7.250	duplex	Z-DNA	poly d(GC) : poly d(GC)
16	-51.4	7.571	duplex	Z-DNA	poly d(As4T) : poly d(As4T)
17	0.0	10.200	duplex	L-DNA	(calf thymus) poly d(GC) : poly d(GC)
18	36.0	3.230	duplex	B'-DNA	alpha poly d(A) : poly d(T) (H-DNA)
19	36.0	3.233	duplex	B'-DNA	beta2 poly d(A) : poly d(T) (H-DNA beta)
20	32.7	2.812	duplex	A-RNA	poly (A) : poly (U)
21	30.0	3.000	duplex	A'-RNA	poly (I) : poly (C)
22	32.7	2.560	duplex	Hybrid	poly (A) : poly d(T)
23	32.0	2.780	duplex	Hybrid	poly d(G) : poly (C)
24	36.0	3.130	duplex	Hybrid	poly d(I) : poly (C)
25	32.7	3.060	duplex	Hybrid	poly d(A) : poly (U)
26	36.0	3.010	duplex	10-fold	poly (X) : poly (X)
27	32.7	2.518	duplex	11-fold	poly (X) : poly (X)
28	32.7	2.596	duplex	poly (s2U)	: poly (s2U) (symmetric base-pair)
29	32.7	2.596	duplex	poly (s2U)	: poly (s2U) (asymmetric base-pair)
30	32.7	3.160	*triplex	poly d(C)	: poly d(I) : poly d(C)
31	30.0	3.260	*triplex	poly d(T)	: poly d(A) : poly d(T)
32	32.7	3.040	*triplex	poly (U)	: poly (A) : poly (U) (11-fold)
33	30.0	3.040	*triplex	poly (U)	: poly (A) : poly (U) (12-fold)
34	30.0	3.290	*triplex	poly (I)	: poly (A) : poly (I)
35	31.3	3.410	#quadruplex	poly (I)	: poly (I) : poly (I) : poly (I)
36	60.0	3.155	1-strand	poly (ethyl C)	

```

37   36.0   3.200 duplex B'-DNA beta2 poly d(A) : poly d(U)
38   36.0   3.240 duplex B'-DNA beta1 poly d(A) : poly d(T)
39   72.0   6.480 duplex B'-DNA beta2 poly d(AI) : poly d(CT)
40   72.0   6.460 duplex B'-DNA beta1 poly d(AI) : poly d(CT)
41  144.0  13.540 duplex B'-DNA poly d(AATT) : poly d(AATT)
42   32.7   3.040 *triplex poly (U) : poly d(A) : poly (U) [cf. #32]
43   36.0   3.200 duplex Beta poly d(A) : poly d(U) [cf. #37]
44   36.0   3.233 duplex poly d(A) : poly d(T) (Ca salt)
45   36.0   3.233 duplex poly d(A) : poly d(T) (Na salt)
46   36.0   3.38  duplex B-DNA (BII-type nucleotides; generic sequence: A, C, G and T)
47   40.0   3.32  duplex C-DNA (BII-type nucleotides; generic sequence: A, C, G and T)
48   87.8   6.02  duplex D(A)-DNA poly d(AT) : poly d(AT) (right-handed)
49   60.0   7.20  duplex S-DNA poly d(CG) : poly d(CG) (C_BG_A, right-handed)
50   60.0   7.20  duplex S-DNA poly d(GC) : poly d(GC) (C_AG_B, right-handed)
51   31.6   3.22  duplex B*-DNA poly d(A) : poly d(T)
52   90.0   6.06  duplex D(B)-DNA poly d(AT) : poly d(AT) [cf. #48]
53 -38.7   3.29  duplex C-DNA (generic sequence: A, C, G and T) (deprecated)
54  32.73  2.56  duplex A-DNA (generic sequence: A, C, G and T) [cf. #1]
55   36.0   3.39  duplex B-DNA (generic sequence: A, C, G and T) [cf. #4]
56  105.0  3.40  *triplex Pauling's triplex model (generic sequence: A, C, G, T, U)
-----
```

- Models #1–#41 are based on Arnott (1999): “Polynucleotide secondary structures: an historical perspective”, pp. 1–38 in “Oxford Handbook of Nucleic Acid Structure” edited by Stephen Neidle (Oxford Press, 1999).
- Models #42 and #43 are derived from Chandrasekaran & Arnott: “The structures of DNA and RNA helices in oriented fibers”, pp 31–170 in “Landolt-Bornstein Numerical Data and Functional Relationships in Science and Technology” edited by Wolfram Saenger (Springer-Verlag, 1990).
- Models #44 and #45 are based on Alexeev *et al.* (1987): “The structure of poly(dA)·poly(dT) as revealed by an X-ray fiber diffraction.” *J. Biomol. Struct. Dyn.*, **4**, pp. 989–1011.
- Models #46 and #47 are based on van Dam and Levitt (2000): “BII nucleotides in the B and C forms of natural-sequence polymeric DNA: a new model for the C form of DNA.” *J. Mol. Biol.*, **304**, pp. 541–561.
- Models #48–#55 are based on Premilat and Albiser (2001): “A new D-DNA form of poly(dA-dT)·poly(dA-dT): an A-DNA type structure with reversed Hoogsteen pairing.” *Eur. Biophys. J.*, **30**, pp. 404–410 (and several other publications).
- Model #56 is based on Pauling and Corey (1953): “A proposed structure for the nucleic acids.” *Proc. Natl. Acad. Sci.*, **39**, pp.84–97.

5.2.3 The `--model` option

The `--model` option selects the model type to be generated (see Section 5.2.2). Generally, an integer between 1 to 56 is specified. For example, `--model=4` selects model #4, the classic B-DNA from calf thymus.

The integer can be replaced by its corresponding keyword for the most common cases, including A-DNA(#1), B-DNA(#4), C-DNA(#47), Z-DNA(#15), Pauling triplex(#56), RNA, and G-quadruplex. For example, `--model=Z-DNA` has the same effect as `--model=15`. By default, `--model=RNA` generates a ssRNA model. For dsRNA, the `--model=RNA-duplex` option should be used. The `--model=G4` option creates a parallel G-quadruplex, based on model #35 but with G replacing I.

The most common cases listed above can also be directly specified via the command-line. For example, `--Z-DNA` works just like `--model=Z-DNA` or `--model=15`. The default output with just `x3dna-dssr fiber` is a B-DNA model, with base sequence A12·T12. Thus, the option `--model=B-DNA` or `--model=4` can be omitted.

5.2.4 The `--sequence` option

The `--sequence` (or simply `--seq`) option specifies the base sequence for generic models (see Section 5.2.2). The sequence can be input in shorthand forms for convenience. Thus, `--seq=AAAAAAATTTTCGGG` can be shortened to `--seq=A6T5CG3`. Alternatively, the sequence can be specified as `--seq=A6-T5-C-G3` or `--seq=A6.T5.C.G3` for improved readability. Unrecognized base characters (i.e., non-ACGTU) are simply ignored.

The closely related `--seq-file` option inputs base sequence from a text file. The format is very simple: it should contain just the sequence (e.g., with content AAAGGGTTTT).

For models with fixed base sequence, the `--seq` and `--seq-file` options are simply ignored. For example, with `--model=Z-DNA --seq=A10`, the option `--seq=A10` has no effects at all.

5.2.5 Other options and some examples

The `--repeat=integer` option specifies the number of repetitions of base sequence. It is applicable to models with generic or fixed sequence. By default, the output coordinate file is in the PDB format. With `--mmcif-output`, the output format changes to mmCIF, allowing for huge structures to be generated.

Here are some examples:

```
x3dna-dssr fiber --seq=A6C10 --repeat=2 --model=RNA
x3dna-dssr fiber --rna -o=rna-ss.pdb
x3dna-dssr fiber --rna-ds -o=rna-duplex.pdb    # --rna-double, --RNA-duplex
x3dna-dssr fiber --g4 -o=g4.pdb

# The following four commands lead to the same results
x3dna-dssr fiber --seq=A6TC9 --repeat=2 -o=B1.pdb
x3dna-dssr fiber --B-DNA --seq=A6TC9 --repeat=2 -o=B2.pdb
x3dna-dssr fiber --model=b-dna --seq=A6TC9 --repeat=2 -o=B3.pdb
x3dna-dssr fiber --model=b-dna --seq=A6-T-C9 --repeat=2 -o=B4.pdb

x3dna-dssr fiber --seq=A1000 --mmcif-output # B-DNA, in mmCIF output format

x3dna-dssr fiber --pauling --seq=A6      # three strands, all A6
x3dna-dssr fiber --pauling --seq=C6:      # one strand: C6 ('A')
x3dna-dssr fiber --pauling --seq=A6:G2    # two strands: A6 ('A') and G2 ('B')
x3dna-dssr fiber --pauling --seq=A6::G2   # two strands: A6 ('A') and G2 ('C')
x3dna-dssr fiber --pauling --seq=:U3:G2   # two strands: U3 ('B') and G2 ('C')
x3dna-dssr fiber --pauling-dna --seq=U8 --repeat=4 # U converted to T
```

5.3 Customized structures (`rebuild`, following 3DNA)

5.3.1 Introduction

The 3DNA `rebuild` program complements `analyze` by reinforcing and verifying it. These two programs are a defining feature of 3DNA (Lu and Olson, 2003, 2008; Li *et al.*, 2019). The 3DNA `analyze` and `rebuild` programs are based on SCHNAAp/SCHNArP (Lu *et al.*, 1997*a,b*) which implement and extend the rigorous CEHS algorithm (El Hassan and Calladine, 1995) for the analysis/rebuilding of DNA duplexes.

The reversibility of the analysis/rebuilding programs in 3DNA allows scientists to ask what-if questions. By first deriving a complete set of base-pair parameters from an experimental structure, they can then systematically introduce changes in these parameters to see what happens to the shapes of the resulting 3D structures. This is a simple, yet powerful concept. 3DNA is the only widely used DNA/RNA structural bioinformatics tool with this feature. It has led to the discovery of a novel roll-and-slide mechanism to account for DNA folding in chromatin (Tolstorukov *et al.*, 2007). Using modeling studies enabled by 3DNA, the Johnson lab at UCLA has revealed slide as a key parameter (along with roll and twist) in mediating DNA minor groove width (Hancock *et al.*, 2019; Chen *et al.*, 2018; Hancock *et al.*, 2016, 2013; Stella *et al.*, 2010).

The DSSR `analyze` module has completely surpassed the 3DNA `analyze` program (see Section 3.18). Similarly, the `rebuild` module in DSSR replaces the 3DNA `rebuild`

program, with enriched functionality and improved usability. The module must be run as `x3dna-dssr rebuild`, just like a sub-command in Git.

The usage of the `rebuild` module is outlined in the listing below. Major options (except for `--G4-model`) are documented in the following sections.

```
x3dna-dssr rebuild [options] --par-file=base-parameters [-o=output-file]

# where options are:
--backbone      # Select backbone type: B-DNA, A-DNA, RNA, or no backbone (default)
--par-type       # Specify the type of par-file. Local step parameters is the default
  --par-type=helical # Used when the par-file contains local helical parameters
--mmcif-output   # Set the output format to mmCIF instead of PDB
--G4-model=number-G-tetrads:twist # G-quadruplex with # of G-tetrads and twist angle
```

5.3.2 The `--backbone` option

The `--backbone` (or simply `--bb`) option specifies a type of standard backbone conformation. Valid types are: (i) B-DNA with C2'-endo sugar puckering, (ii) A-DNA with C3'-endo sugar puckering, (iii) RNA with O2' atom and C3'-endo sugar puckering, and (iv) `none` without backbone attached (default).

The base building blocks are derived from the work of Clowney *et al.* (1996). The chosen backbone conformation, as a ‘riding’ sugar-phosphate group, is attached to each base in the rebuilding process. The resultant backbone is oftentimes distorted, without proper covalent linkages between neighboring nucleotides. This approximate backbone connection can be optimized, using PHENIX (Adams *et al.*, 2010), while keeping the base atoms fixed. The DSSR-PHENIX combination leads to a model where the base geometry strictly follows the parameters prescribed in the user-specified file, and the backbone is regularized with improved stereochemistry and a ‘smooth’ appearance in ribbon representation.

Specifically, the PHENIX command runs like below:

```
phenix.geometry_minimization dssr-rebuild.pdb min.params
```

Here `dssr-rebuild.pdb` is the file name of a DSSR-rebuilt structure with an approximate backbone (see examples below). The minimization parameters file `min.params` has the following contents:

```
pdb_interpretation {
    link_distance_cutoff = 7.0
}
```

```

selection = name " P " or name " OP1" or name " OP2" or \
name " O5'" or name " C5'" or name " C4'" or \
name " O4'" or name " C3'" or name " O3'" or \
name " C2'" or name " O2'"

```

The PHENIX optimized structure is named `dssr-rebuild_minimized.pdb` (i.e., with suffix `_minimized` before the `.pdb` extension).

5.3.3 Other options and some examples

The `--par-file=base-parameters` option is required. It specifies the file that contains rigid-body parameters for the rebuilding procedure. The type of parameters is specified via the `--par-type` option. By default (i.e., without this option being set explicitly), local step parameters are assumed. With `--par-type=helical`, the parameters file should contain local helical parameters. See Section 3.18.

By default, the output coordinate file is in PDB format. With `--mmcif-output`, the output format changes to mmCIF, allowing for huge structures to be generated.

Here are some examples:

- Rebuilding DNA duplex, using PDB entry 355d (Shui *et al.*, 1998) as an example

```

# The following command generates files: dssr-dsStepPars.txt and dssr-dsHeliPars.txt
x3dna-dssr analyze --rebuild -i=355d.pdb -o=355d-expt.out

# Rename them to 355d-step.txt and 355d-heli.txt respectively
mv dssr-dsStepPars.txt 355d-step.txt
mv dssr-dsHeliPars.txt 355d-heli.txt

# By default (without the --backbone option), only base atoms are included
x3dna-dssr rebuild --par-file=355d-step.txt -o=355d-bases.pdb

# Rebuild two structures using local step and helical parameters, with B-DNA backbone
x3dna-dssr rebuild --backbone=B-DNA --par-file=355d-step.txt -o=355d-step.pdb
x3dna-dssr rebuild --bb=BDNA --par-file=355d-heli.txt --par-type=heli -o=355d-heli.pdb

# Re-analyze the two structures rebuilt by DSSR, for verification
x3dna-dssr analyze -i=355d-step.pdb -o=355d-step.out
x3dna-dssr analyze -i=355d-heli.pdb -o=355d-heli.out

```

The local base-pair and step parameters (file `355d-step.txt`) for 355d are listed in Page 77. The two structures `355d-step.pdb` and `355d-heli.pdb` rebuilt by DSSR are identical: the rmsd between them is zero. The base-pair geometry of `355d-step.pdb` is virtually identical to that of `355d.pdb` (the original, experimental structure). There are two ways to validate this point: (i) The rmsd between `355d-step.pdb` and `355d.pdb`

for *base atoms* is only 0.02 Å (Lu and Olson, 2003). (ii) Reanalyzing 355d-step.pdb gives similar base-pair parameters as those of 355d.pdb (compare 355d-step.out and 355d-expt.out).

Finally, running phenix.geometry_minimization 355d-step.pdb min.params optimizes the approximate backbone conformation. The resultant structure is stored in file 355d-step_minimized.pdb which has smoothed backbone connections. Loading 355d-step.pdb and 355d-step_minimized.pdb into PyMOL or Jmol would clearly reveal the differences in backbone geometries.

- Taking DNA duplex 355d as single-stranded

```
# The following command generates files: dssr-ssStepPars.txt and dssr-ssHeliPars.txt
x3dna-dssr analyze --ss --rebuild -i=355d.pdb -o=355d-expt-ss.out

# Rename them to 355d-step-ss.txt and 355d-heli-ss.txt respectively
mv dssr-ssStepPars.txt 355d-step-ss.txt
mv dssr-ssHeliPars.txt 355d-heli-ss.txt

# By default (without the --backbone option), only base atoms are included
x3dna-dssr rebuild --par-file=355d-step-ss.txt -o=355d-bases-ss.pdb

# Rebuild two structures using local step and helical parameters, with B-DNA backbone
x3dna-dssr rebuild --backbone=B-DNA --par-file=355d-step-ss.txt -o=355d-step-ss.pdb
x3dna-dssr rebuild --bb=B-DNA --par-file=355d-heli-ss.txt --par-type=helical -o=355d-
    ↪ heli-ss.pdb

# Re-analyze the two structures rebuilt by DSSR, for verification
x3dna-dssr analyze --ss -i=355d-step-ss.pdb -o=355d-step-ss.out
x3dna-dssr analyze --ss -i=355d-heli-ss.pdb -o=355d-heli-ss.out
```

The local step parameters (file 355d-step-ss.txt) when taking 355d as single-stranded are shown in Page 79. Running commands in the above listing give conclusions similar to those shown above where 355d is taken as a DNA duplex. In fact, all structural types (including G-quadruplexes) can be taken as ‘pseudo’ single-stranded. The procedure can even be applied to ribosomal structures with multiple RNA molecules, e.g., PDB entry 1jj2 (Klein *et al.*, 2001) with 23S rRNA and 5S rRNA.

- Rebuilding ssRNA, using PDB entry 1msy (Correll *et al.*, 2003) as an example

```
1 x3dna-dssr analyze --ss --rebuild -i=1msy.pdb -o=1msy-expt.out
2 mv dssr-ssStepPars.txt 1msy-step.txt
3 x3dna-dssr rebuild --backbone=RNA --par-file=1msy-step.txt -o=1msy-step.pdb
4 x3dna-dssr analyze --ss -i=1msy-step.pdb -o=1msy-step.out
```

Comparing 1msy-step.out and 1msy-expt.out shows that the DSSR-rebuilt model

and experimental structure have very similar base morphologies, as would be expected. The rebuilt structure (`1msy-step.pdb`) has standard RNA backbone conformation (line no. 3), with C3'-endo sugar puckering. As shown in “Summary of structural features per nucleotide” and “Sugar conformational parameters” for PDB entry `1msy`, the RNA nucleotides A2654 and G2655 actually have C2'-endo sugar puckering, forming part of the bulged-G motif with a distinctive S-shaped backbone (Correll *et al.*, 2003; Lu *et al.*, 2010). Such structural heterogeneity is out of the design scope of the DSSR `rebuild` module. Even with `phenix.geometry_minimization 1msy-step.pdb min.params`, the resultant model `1msy-step_minimized.pdb` still has connectivity issue in this region. See Figure 25. The sugar puckering of nucleotides A2654 and G2655 must be switched from C3'-endo to C2'-endo for the backbone to be connected smoothly.

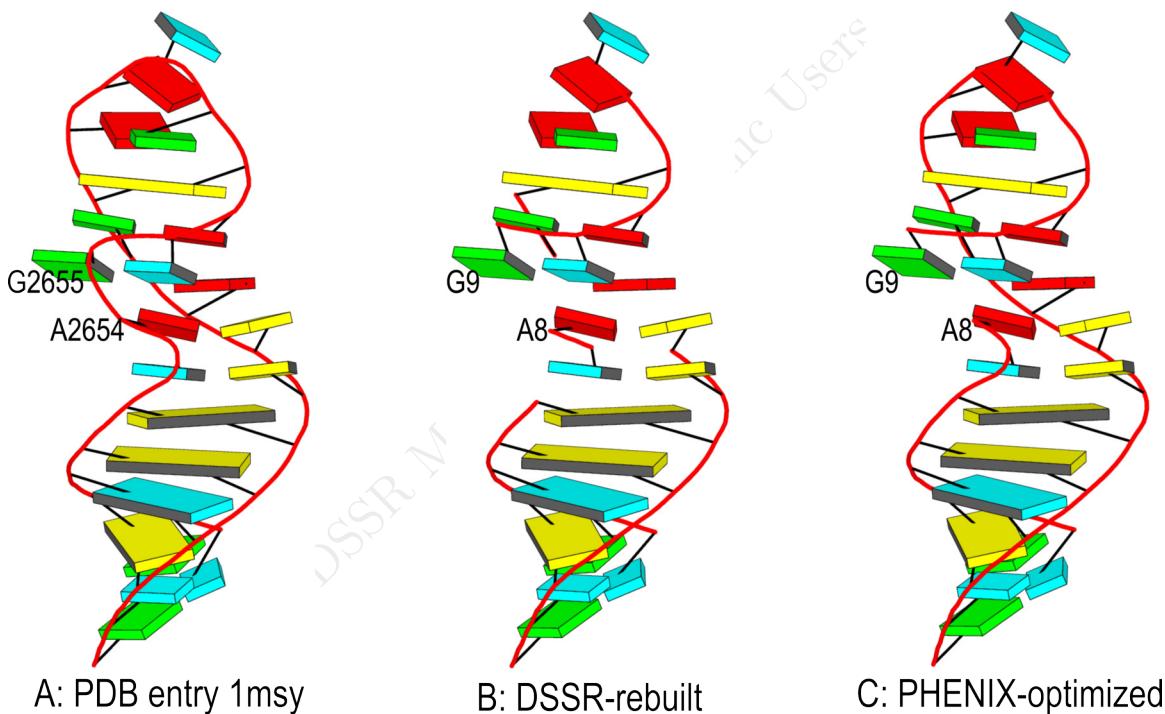


Figure 25: RNA rebuilding with DSSR. (A) The 27-nt long RNA fragment from PDB entry `1msy` (Correll *et al.*, 2003). A2654 and G2655 (PDB designated residue numbers) have C2'-endo sugar puckering. They are part of the bulged-G motif, with backbone in the characteristic S-shape. (B) The raw DSSR-rebuilt structure, using a standard RNA backbone conformation. Note the many breaks in the backbone ribbon. A8 and G9 correspond to A2654 and G2655 in the PDB entry, but with standard C3'-endo sugar puckering in RNA. (C) PHENIX-optimized structure. The backbone ribbon is much smoother than the raw DSSR-rebuilt model. However, the backbone between A8 and G9 are still disconnected. This figure was annotated using Inkscape (<https://inkscape.org>).

The contents of the parameters file `1msy-step.txt` is shown below. Both bases and step parameters can be modified using a text editor to create new models accordingly.

# 27 (no. of nucleotides)							
#	Shift	Slide	Rise	Tilt	Roll	Twist	
U	3.1800	-1.6170	3.4738	-3.4224	5.1912	55.1526	# 1 A.U2647
G	1.3559	-1.2134	3.1106	4.3601	2.5983	41.7785	# 2 A.G2648
C	1.2973	-1.1809	3.2795	2.2126	9.0432	37.1762	# 3 A.C2649
U	0.0197	-1.4679	3.1300	5.5194	4.0159	31.3293	# 4 A.U2650
C	0.3310	-1.7503	3.4643	-0.4444	12.0558	28.6350	# 5 A.C2651
C	2.1190	-0.6070	2.8472	12.3852	2.4408	43.4222	# 6 A.C2652
U	-2.2971	-2.5959	0.9173	-39.2264	162.7954	144.4556	# 7 A.U2653
A	-4.5732	5.1413	2.2432	45.1538	-169.4684	134.6279	# 8 A.A2654
G	-7.8911	-1.7490	0.2215	-1.8717	-2.9905	-12.3938	# 9 A.G2655
U	0.9434	-1.9539	3.8632	-9.3424	1.7996	39.0418	# 10 A.U2656
A	3.7705	-0.7952	3.3527	1.9446	3.5445	61.1256	# 11 A.A2657
C	1.4539	-1.8431	3.5894	-3.6260	9.1707	36.3895	# 12 A.C2658
G	-12.2963	-4.4457	1.5623	79.5040	127.1207	-146.8666	# 13 A.G2659
U	2.9748	1.8361	3.0330	15.4018	8.1120	53.8726	# 14 A.U2660
A	2.3828	1.4379	3.1171	12.0601	14.7522	42.8234	# 15 A.A2661
A	7.3620	-0.0294	2.3929	3.2949	20.8874	49.9240	# 16 A.A2662
G	4.3387	-1.1614	3.1716	5.5819	6.3446	59.9253	# 17 A.G2663
G	-9.7458	-5.1796	4.0060	2.5473	-13.4251	-67.0882	# 18 A.G2664
A	-1.0213	-2.5316	2.9469	7.3642	-6.1103	20.0373	# 19 A.A2665
C	3.6036	-0.9384	3.2939	2.6473	7.6176	55.8084	# 20 A.C2666
C	4.4027	-1.1585	3.4320	-0.3787	6.7697	60.0838	# 21 A.C2667
G	0.5648	-1.7223	3.2142	1.8689	3.9359	30.7234	# 22 A.G2668
G	0.9355	-1.7777	3.2385	1.3265	3.5899	35.2327	# 23 A.G2669
A	-0.3151	-1.6632	3.1128	2.8658	5.3633	29.3319	# 24 A.A2670
G	1.8172	-1.2262	2.9034	8.4923	5.2643	41.2995	# 25 A.G2671
U	-3.0210	-1.0177	3.0490	6.5362	9.2713	14.3552	# 26 A.U2672
G	999999	999999	999999	999999	999999	999999	# 27 A.G2673

6 Utilities of general purposes

As detailed in this manual, DSSR has a plethora of features for 3D nucleic acid structures. It also possesses functionalities that may be of general interest. Four additional DSSR options are documented below.

6.1 The --select option

The `--select` option is used to extract residues based on structural attributes. It comes with many variations to cover most common use cases. These features are easily illustrated with examples, as shown below.

```
x3dna-dssr -i=1ehz.pdb --select      # the default is to select all ATOM/HETATM records
x3dna-dssr -i=1ehz.pdb --select=nt    # only nucleotides (DNA or RNA)
x3dna-dssr -i=1ehz.pdb --select=nt-metal # nucleotides and metal ions
x3dna-dssr -i=1oct.pdb --select=aa   # only amino acids
```

```
x3dna-dssr -i=1oct.pdb --select=aa+nt    # amino acids and nucleotides
x3dna-dssr -i=2n2d.pdb --select-model=6 # model no.6 from an NMR ensemble
x3dna-dssr -i=1jj2.pdb --select-chain=0 # chain 0 (the 23S rRNA)

# to select residues, specify chain-id and residue-numbers
x3dna-dssr -i=1ehz.pdb --select-residue='A 34+35+36' # residues [34,35,36] on chain A
x3dna-dssr -i=1ehz.pdb --select-residue='A 34:36'      # same result as above
x3dna-dssr -i=1ehz.pdb --select-residue='A 13:22 A 30:40 A 53:61' # three hairpin loops
x3dna-dssr -i=1ehz.pdb --select-residue='A 13:22+30:40+53:61' # same as above
x3dna-dssr -i=1ehz.pdb --select-residue='A 7:10 A 25:27 A 43:49 A 65+66' # 4-way junction
```

6.2 The --view option

The DSSR `--view` option reorients a structure with regard to the principal moment of inertia. By default, it employs only atoms of nucleic acids. It transforms the whole structure into a vertically extended orientation.

Using tRNA 1ehz as an example, the `--view` option can be used as below. Figure 10A was oriented into the view specified in file `1ehz-view.pdb`.

```
x3dna-dssr -i=1ehz.pdb --view -o=1ehz-view.pdb
```

6.3 The --frame option

The `--frame` option can be used to reorient a structure with reference to a specific base (or base-pair) reference frame, or the middle frame of two related frames. Using the classic B-DNA dodecamer PDB entry 355d (Shui *et al.*, 1998) as an example, DSSR can be run with the `--frame` option as follows:

```
#           1...5..8....
# chain A: 5'-CGCGAATTGCGG -3'
# chain B: 3'-GCGCTTAAGCGC -5'

# reorient 355d in the reference frame of C1 on chain A
x3dna-dssr -i=355d.pdb --frame=A.1 -o=355d-b1.pdb

# reorient 355d in the frame of the Watson-Crick pair C1-G24
x3dna-dssr -i=355d.pdb --frame=A.1:wc -o=355d-bp1.pdb

# ... with the minor-groove of pair C1-G24 facing the viewer
x3dna-dssr -i=355d.pdb --frame=A.1:wc-minor -o=355d-bp1-minor.pdb

# with the minor-groove of the middle AATT tract facing the viewer
x3dna-dssr -i=355d.pdb --frame='A.5:wc-minor A.8:wc' -o=355d-AATT-minor.pdb
```

The abbreviated notation `A.1` refers to nucleotide numbered 1 (as indicated in the coor-

dinates file) on chain A. Here, A.1 denotes C1, as shown at the top of the listing. Similarly, A.5 and A.8 correspond to A5 and T8 on chain A, respectively.

In most cases, such as with 355d, the combination of chain identifier and residue number is sufficient to uniquely identify a nucleotide. However, more information may be required in general to specify a nucleotide accurately. This additional information can include the model number, residue name, and insertion code, which are necessary for unique identification when multiple nucleotides share the same chain id and residue number.

In the above listing, `wc` specifies the Watson-Crick base pair that a particular nucleotide participates in. Meanwhile, `minor` makes the minor-groove edge to face the viewer (Figure 1C and Figure 18A). To visualize the effects of these settings, load the 355d.pdb file and the transformed coordinate files into a molecular visualization tool like PyMOL.

6.4 The --get-hbond option

H-bonding interactions are crucial for defining RNA secondary and tertiary structures. DSSR implements a geometric approach to identify H-bonds in 3D structures of nucleic acids or proteins. The method has been continuously refined, and works well in real-world applications (Lu and Olson, 2003; Lu *et al.*, 2015). With the `--get-hbond` option, DSSR outputs a list of H-bonds.

Running DSSR with the `--get-hbond` option on PDB entry 1msy (Correll *et al.*, 2003) leads to the following results:

```
# H-bonds in '1msy.pdb' identified by DSSR
40
 15  578 #1      p    2.768 O:N O40A.U2647 N10A.G2673
 35  555 #2      p    2.776 O:N O60A.G2648 N30A.U2672
 36  554 #3      p    2.826 N:O N10A.G2648 O20A.U2672
 55  537 #4      p    2.965 O:N O20A.C2649 N20A.G2671
 56  535 #5      p    2.836 N:N N30A.C2649 N10A.G2671
 58  534 #6      p    2.769 N:O N40A.C2649 O60A.G2671
 76  513 #7      p    2.806 N:N N30A.U2650 N10A.A2670
 78  512 #8      p    3.129 O:N O40A.U2650 N60A.A2670
 95  492 #9      p    2.703 O:N O20A.C2651 N20A.G2669
 96  490 #10     p    2.853 N:N N30A.C2651 N10A.G2669
 98  489 #11     p    2.987 N:O N40A.C2651 O60A.G2669
115  466 #12     p    2.817 O:N O20A.C2652 N20A.G2668
116  464 #13     p    2.907 N:N N30A.C2652 N10A.G2668
118  463 #14     p    2.897 N:O N40A.C2652 O60A.G2668
123  151 #15     o    2.622 O:O OP20A.U2653 O2'0A.A2654
135  443 #16     p    2.898 O:N O20A.U2653 N40A.C2667
147  192 #17     x    3.054 O:O O4'0A.A2654 O4'0A.U2656
158  408 #18     p    2.960 N:O N60A.A2654 OP20A.C2666
173  188 #19     o    2.923 O:O O2'0A.G2655 OP20A.U2656
173  378 #20     o    3.093 O:O O2'0A.G2655 O60A.G2664
```

173	379	#21	o	3.343	O:N	02'@A.G2655	N1@A.G2664
181	386	#22	p	2.768	N:O	N1@A.G2655	OP2@A.A2665
183	203	#23	p	2.754	N:O	N2@A.G2655	O4@A.U2656
183	387	#24	p	2.887	N:O	N2@A.G2655	O5'@A.A2665
188	379	#25	p	3.044	O:N	OP2@A.U2656	N1@A.G2664
188	381	#26	p	2.944	O:N	OP2@A.U2656	N2@A.G2664
200	401	#27	p	3.122	O:N	02@A.U2656	N6@A.A2665
201	398	#28	p	2.759	N:N	N3@A.U2656	N7@A.A2665
220	381	#29	p	3.035	N:N	N7@A.A2657	N2@A.G2664
223	371	#30	o	2.963	N:O	N6@A.A2657	O2'@A.G2664
223	382	#31	p	3.039	N:N	N6@A.A2657	N3@A.G2664
242	358	#32	p	2.821	O:N	O2@A.C2658	N2@A.G2663
243	356	#33	p	2.890	N:N	N3@A.C2658	N1@A.G2663
245	355	#34	p	2.887	N:O	N4@A.C2658	O6@A.G2663
258	305	#35	o	2.604	O:N	O2'@A.G2659	N7@A.A2661
258	308	#36	o	3.264	O:N	O2'@A.G2659	N6@A.A2661
268	315	#37	p	2.973	N:O	N2@A.G2659	OP2@A.A2662
268	327	#38	p	2.864	N:N	N2@A.G2659	N7@A.A2662
371	390	#39	o	2.751	O:O	O2'@A.G2664	O4'@A.A2665
550	566	#40	o	3.372	O:O	O2'@A.U2672	O4'@A.G2673

The first line is the header. The second line provides the total number of H-bonds (40) identified in the structure. Afterwards, each line consists of 8 space-delimited fields used to characterize a specific H-bond. Using the first H-bonds (15 578 #1) as an example, the meanings of the 8 fields are:

1. The serial number (15), as denoted in the coordinates file, of the first atom of the H-bond.
2. The serial number (578) of the second atom of the H-bond.
3. The H-bond index (#1), from 1 to the total number of H-bonds.
4. A one-letter symbol showing the atom-pair type (p) of the H-bond. It is p for a donor-acceptor atom pair; o when any H-bonding atom is a donor/acceptor (such as the 2'-hydroxyl oxygen); x for a donor-donor or acceptor-acceptor pair (as in #17); ? if the donor/acceptor status of any H-bonding atom is unknown.
5. Distance between donor/acceptor atoms in Å (2.768).
6. Elemental symbols of the two atoms involved in the H-bond (O:N).
7. Identifier of the first H-bonding atom (O4@A.U2647).
8. Identifier of the second H-bonding atom (N1@A.G2673).

By default, the DSSR --get-hbond command outputs only H-bonds within nucleic acids (DNA and RNA). For 1ehz and 1jj2, DSSR identifies 113 and 5,644 H-bonds respectively. By specifying the option as --get-hbond=nuc+protein, DSSR also outputs H-bonds in proteins

or at the interface of DNA/RNA-protein complexes. For example, when proteins are taken into consideration, DSSR detects a total of 10,395 H-bonds in PDB entry 1jj2 (the 50S large ribosomal subunit).

When `--get-hbond` is used in combination with `--json`, DSSR always reports all H-bonds within the analyzed structure. For each H-bond, DSSR provides the `residue_pair` field (with values like `nt:nt`, `nt:aa`, `aa:aa`, etc.) that allows for easy filtering of a particular H-bonding type, as demonstrated below.

```
x3dna-dssr -i=1oct.pdb --get-hbond --json -o=hbonds.json
jq '.hbonds[] | select(.residue_pair=="nt:nt")' hbonds.json # H-bonds within DNA
jq '.hbonds[] | select(.residue_pair=="nt:aa")' hbonds.json # H-bonds at the interface
jq '.hbonds[] | select(.residue_pair=="aa:aa")' hbonds.json # H-bonds within protein
```

The method for identifying H-bonds in DSSR is robust and efficient. It has been extensively tested using nucleic-acid-containing structures in the PDB. HBPLUS (McDonald and Thornton, 1994) and HBExplore (Lindauer *et al.*, 1996) are two dedicated tools for characterizing H-bonds. However, they no longer seem to be maintained and cannot handle mmCIF format. DSSR may well serve as a pragmatic choice for most applications (Sagendorf *et al.*, 2020).

7 Frequently asked questions (FAQs)

7.1 What does DSSR stand for?

DSSR stands for Dissecting the Spatial Structure of RNA. It could also mean Defining (or Dictionary of) the Secondary Structure of RNA. Note that DSSR has far more to offer than just defining RNA 2D structures as DSSP does for proteins.

7.2 How does DSSR compare with other tools?

Numerous software programs and online resources are currently in use for nucleic acid structural bioinformatics. It is fair to say that each has its unique features and no two tools are identical. Comparative studies as seen in the literature are useful for general understanding of a topic. Without going into technical details, however, such comparisons are mostly superficial and seldom convincing. See our early paper on “Resolving the discrepancies among nucleic

acid conformational analyses” (Lu and Olson, 1999).

DSSR is an integrated and automated computational tool designed from the bottom up to streamline RNA structural bioinformatics. It possesses a combined set of functionalities well beyond the scope of any other software tools in the field. As a simple example, users are encouraged to compare DSSR with any other tools on the classic yeast tRNA^{Phe} (1ehz, see Section 3.3). Pay close attention to the fact that DSSR automatically identifies/annotates[†] the following salient features: 14 modified nucleotides, four base triplets, two helices corresponding to the L-shaped tertiary structure, four stems matching the cloverleaf 2D structure, three hairpin loops, and a [2,1,5,0] four-way junction loop.

7.3 How is DSSR related to 3DNA?

From a historical perspective, DSSR is built upon the 3DNA suite of software programs for the analysis, rebuilding, and visualization of 3D nucleic acid structures (Lu and Olson, 2003, 2008; Li *et al.*, 2019). 3DNA takes advantage of the standard base reference frame (Olson *et al.*, 2001), enabled by our contributions on resolving the discrepancies among nucleic acid conformational analyses (Lu and Olson, 1999; Lu *et al.*, 1999). 3DNA also benefitted greatly from the SCHNAAp and SCHNArP pair of programs for rigorous analysis and reversible rebuilding of double-helical nucleic acid structures (Lu *et al.*, 1997*a,b*). <Specifically, the algorithms that underpinned SCHNAAp/SCHNArP laid the foundation of `analyze/rebuild`, two core components of the 3DNA suite.

Just as 3DNA has replaced SCHNAAp/SCHNArP in functionality and real-world applications, DSSR has superseded 3DNA. Key 3DNA features for analysis, visualization, modeling, and utilities have been integrated into DSSR, with vastly enhanced functionality and significantly improved usability. Notably, the `mutate/fiber/rebuild` modules in DSSR completely supersede the `mutate_bases`, `fiber`, and `rebuild` programs distributed with 3DNA. In contrast to 3DNA’s over thirty C programs and utility scripts, DSSR has only a single binary executable that is self-contained, trivial to get up and running, and does more.

[†]The command is `x3dna-dssr -i=1ehz.pdb` (or `1ehz.cif`) and it takes virtually no time to finish.

8 SNAP for nucleic-acid/protein complexes

DNA/RNA-protein interactions underpin fundamental biological processes such as transcription, splicing, and translation. The increasing number of experimentally determined 3D structures of nucleic acid-protein complexes in the PDB provides unprecedented opportunities to decipher the underlying principles governing the process of DNA/RNA-protein recognition (Sagendorf *et al.*, 2020; Feng *et al.*, 2024; Mitra *et al.*, 2025). We have developed SNAP to streamline the characterization of Structures of Nucleic Acid-Protein complexes (Kribelbauer *et al.*, 2020), starting directly from atomic coordinates in PDB or mmCIF format. Initially distributed as a standalone program (`x3dna-snap`), SNAP was later integrated into DSSR as a module (i.e., `x3dna-dssr snap`) for easy maintenance and usage. Now, users just need one download/installation to have both DSSR and SNAP.

Details about SNAP will be reported elsewhere: this manual already contains more DSSR features than a typical user would need. The following help message should nevertheless get users started quickly. Questions are always welcome on the 3DNA Forum: there is a section dedicated to “DNA/RNA-protein interactions (SNAP)”.

```
# x3dna-dssr snap -h

Usage: x3dna-dssr snap [options]

Each option is specified via --key[=val] (or -key[=val] or key[=val];
i.e., two/one/zero preceding dashes are all accepted), where 'key' can
be in either lower, UPPER or MiXed case. Options can be in any order.

Options:
--help          Print this command-line help information (-h)
--input=file    Specify a PDB/mmCIF file for analysis (-i=file)
--output=file   Designate the main SNAP output file (-o=file) [stdout]
--cutoff=float  Set the distance cutoff between interacting aa/nt [4.5]
--type=string   Select interacting nt moiety: base|backbone|both|either [either]
--t-shape       Output T-shaped contacts between planar moieties
--interface    Summarize each nucleotide with its contacted amino acids
--methyl-C     Check interactions between DNA 5mC and protein side chains
--auxfile      Produce numerous auxiliary files (e.g. A-arg.pdb)
--nmr          Process an ensemble of NMR structures
--cleanup      Remove common files with fixed names and exit

Examples:
x3dna-dssr snap -i=1oct.pdb
```

9 Acknowledgements

The development of DSSR has been made possible by NIH grants R01GM096889 (2011–2020) and R24GM153869 (2024–). We thank Cathy Lawson, Pascal Auffinger, Di Liu, Shuxiang Li, Mauricio Esguerra, Honglue Shi, Eugene Baulin, and many other users for their feedback.

References

- Abou Assi,H., Garavís,M., González,C. and Damha,M.J. (2018) i-Motif DNA: structural features and significance to cell biology. *Nucleic Acids Res.*, **46** (16), 8038–8056.
- Adams,P.D., Afonine,P.V., Bunkoczi,G., Chen,V.B., Davis,I.W., Echols,N., Headd,J.J., Hung,L.W., Kapral,G.J., Grosse-Kunstleve,R.W., McCoy,A.J., Moriarty,N.W., Oeffner,R., Read,R.J., Richardson,D.C., Richardson,J.S., Terwilliger,T.C. and Zwart,P.H. (2010) PHENIX: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr. Biol. Crystallogr.*, **66** (Pt 2), 213–21.
- Alexeev,D.G., Lipanov,A.A. and Skuratovskii,I.Y. (1987) The structure of poly(dA)·poly(dT) as revealed by an X-ray fibre diffraction. *J. Biomol. Struct. Dyn.*, **4** (6), 989–1012.
- AlQuraishi,M. and McAdams,H.H. (2013) Three enhancements to the inference of statistical protein-DNA potentials. *Proteins*, **81** (3), 426–442.
- Altona,C. and Sundaralingam,M. (1972) Conformational analysis of the sugar ring in nucleosides and nucleotides. A new description using the concept of pseudorotation. *J. Am. Chem. Soc.*, **94** (23), 8205–12.
- Arnott,S. (1999) Polynucleotide secondary structures: an historical perspective. In *Oxford Handbook of Nucleic Acid Structure*, (Neidle,S., ed.). Oxford University Press, first edition, pp. 1–38.
- Assi,H.A., Harkness,R.W., Martin-Pintado,N., Wilds,C.J., Campos-Olivas,R., Mittermaier,A.K., González,C. and Damha,M.J. (2016) Stabilization of i-motif structures by 2'- β -fluorination of DNA. *Nucleic Acids Res.*, **44** (11), 4998–5009.
- Bayrak,C.S., Kim,N. and Schlick,T. (2017) Using sequence signatures and kink-turn motifs in knowledge-based statistical potentials for RNA structure prediction. *Nucleic Acids Res.*, **45** (9), 5414–5422.
- Brčić,J. and Plavec,J. (2015) Solution structure of a DNA quadruplex containing ALS and FTD related GGGGCC repeat stabilized by 8-bromodeoxyguanosine substitution. *Nucleic Acids Res.*, **43** (17), 8590–8600.
- Burge,S., Parkinson,G.N., Hazel,P., Todd,A.K. and Neidle,S. (2006) Quadruplex DNA: Sequence, topology and structure. *Nucleic Acids Res.*, **34** (19), 5402–5415.
- Burley,S.K., Berman,H.M., Christie,C., Duarte,J.M., Feng,Z., Westbrook,J., Young,J. and Zardecki,C. (2018) RCSB Protein Data Bank: Sustaining a living digital data resource that enables breakthroughs in scientific research and biomedical education. *Protein Sci.*, **27** (1), 316–330.

- Cate,J.H., Gooding,A.R., Podell,E., Zhou,K., Golden,B.L., Kundrot,C.E., Cech,T.R. and Doudna,J.A. (1996) Crystal structure of a group I ribozyme domain: principles of RNA packing. *Science*, **273** (5282), 1678–85.
- Chen,V.B., Arendall,W.B., Headd,J.J., Keedy,D.A., Immormino,R.M., Kapral,G.J., Murray,L.W., Richardson,J.S. and Richardson,D.C. (2010) MolProbity: all-atom structure validation for macromolecular crystallography. *Acta Crystallogr. D Biol. Crystallogr.*, **66** (1), 12–21.
- Chen,W., Mandali,S., Hancock,S.P., Kumar,P., Collazo,M., Cascio,D. and Johnson,R.C. (2018) Multiple serine transposase dimers assemble the transposon-end synaptic complex during IS607-family transposition. *eLife*, **7**, e39611.
- Clowney,L., Jain,S.C., Srinivasan,A.R., Westbrook,J., Olson,W.K. and Berman,H.M. (1996) Geometric parameters in nucleic acids: nitrogenous bases. *J. Am. Chem. Soc.*, **118** (3), 509–518.
- Correll,C.C., Beneken,J., Plantinga,M.J., Lubbers,M. and Chan,Y.L. (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** (23), 6806–6818.
- Darty,K., Denise,A. and Ponty,Y. (2009) VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, **25** (15), 1974–5.
- D'Ascenzo,L., Leonarski,F., Vicens,Q. and Auffinger,P. (2017) Revisiting GNRA and UNCG folds: U-turns versus Z-turns in RNA hairpin loops. *RNA*, **23** (3), 259–269.
- Dickerson,R.E., Bansal,M., Calladine,C.R., Diekmann,S., Hunter,W.N., Kennard,O., von Kitzing,E., Lavery,R., Nelson,H.C.M., Olson,W.K., Saenger,W., Shakkeb,Z., Sklenar,H., Soumpasis,D.M., Tung,C.S., Wang,A.H.J. and Zhurkin,V.B. (1989) Definitions and nomenclature of nucleic acid structure parameters. *EMBO J.*, **8** (1), 1–4.
- Dock-Bregeon,A.C., Chevrier,B., Podjarny,A., Johnson,J., de Bear,J.S., Gough,G.R., Gilham,P.T. and Moras,D. (1989) Crystallographic structure of an RNA helix: [U(UA)6A]2. *J. Mol. Biol.*, **209** (3), 459–474.
- Dvorkin,S.A., Karsisiotis,A.I. and Webba da Silva,M. (2018) Encoding canonical DNA quadruplex structure. *Sci. Adv.*, **4** (8), eaat3007.
- El Hassan,M.A. and Calladine,C.R. (1995) The assessment of the geometry of dinucleotide steps in double-helical DNA; a new local calculation scheme. *J. Mol. Biol.*, **251** (5), 648–64.
- Feng,H., Lu,X.J., Maji,S., Liu,L., Ustianenko,D., Rudnick,N.D. and Zhang,C. (2024) Structure-based prediction and characterization of photo-crosslinking in native protein–RNA complexes. *Nat. Commun.*, **15** (1), 2279.
- Gesteland,R.F., Cech,T. and Atkins,J.F. (1999) *The RNA world: the nature of modern RNA suggests a prebiotic RNA world*. Number 37 in Cold Spring Harbor monograph series, second edition, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.
- Gutell,R.R., Cannone,J.J., Konings,D. and Gautheret,D. (2000) Predicting U-turns in Ribosomal RNA with Comparative Sequence Analysis. *J. Mol. Biol.*, **300** (4), 791–803.

- Han,X., Leroy,J.L. and Guéron,M. (1998) An intramolecular i-motif: the solution structure and base-pair opening kinetics of d(5mCCT3CCT3ACCT3CC). *J. Mol. Biol.*, **278** (5), 949–965.
- Hancock,S.P., Cascio,D. and Johnson,R.C. (2019) Cooperative DNA binding by proteins through DNA shape complementarity. *Nucleic Acids Res.*, **47** (16), 8874–8887.
- Hancock,S.P., Ghane,T., Cascio,D., Rohs,R., Di Felice,R. and Johnson,R.C. (2013) Control of DNA minor groove width and Fis protein binding by the purine 2-amino group. *Nucleic Acids Res.*, **41** (13), 6750–6760.
- Hancock,S.P., Stella,S., Cascio,D. and Johnson,R.C. (2016) DNA Sequence Determinants Controlling Affinity, Stability and Shape of DNA Complexes Bound by the Nucleoid Protein Fis. *PLoS ONE*, **11** (3), e0150189.
- Hanson,R.M. and Lu,X.J. (2017) DSSR-enhanced visualization of nucleic acid structures in Jmol. *Nucleic Acids Res.*, **45** (W1), W528–W533.
- Howe,J.A., Wang,H., Fischmann,T.O., Balibar,C.J., Xiao,L., Galgoci,A.M., Malinvern,J.C., Mayhood,T., Villafania,A., Nahvi,A., Murgolo,N., Barbieri,C.M., Mann,P.A., Carr,D., Xia,E., Zuck,P., Riley,D., Painter,R.E., Walker,S.S., Sherborne,B., de Jesus,R., Pan,W., Plotkin,M.A., Wu,J., Rindgen,D., Cummings,J., Garlisi,C.G., Zhang,R., Sheth,P.R., Gill,C.J., Tang,H. and Roemer,T. (2015) Selective small-molecule inhibition of an RNA structural element. *Nature*, **526** (7575), 672–677.
- Ingle,S., Azad,R.N., Jain,S.S. and Tullius,T.D. (2014) Chemical probing of RNA with the hydroxyl radical at single-atom resolution. *Nucleic Acids Res.*, **42** (20), 12758–12767.
- Jucker,F.M. and Pardi,A. (1995) GNRA tetraloops make a U-turn. *RNA*, **1** (2), 219–222.
- Karg,B., Mohr,S. and Weisz,K. (2019) Duplex-guided refolding into novel G-quadruplex (3+1) hybrid conformations. *Angew. Chem. Int. Ed.*, **58** (32), 11068–11071.
- Keating,K.S., Humphris,E.L. and Pyle,A.M. (2011) A new way to see RNA. *Q. Rev. Biophys.*, **44** (4), 433–66.
- Klein,D.J., Schmeing,T.M., Moore,P.B. and Steitz,T.A. (2001) The kink-turn: a new RNA secondary structure motif. *EMBO J.*, **20** (15), 4214–21.
- Kribelbauer,J.F., Lu,X.J., Rohs,R., Mann,R.S. and Bussemaker,H.J. (2020) Toward a mechanistic understanding of DNA methylation readout by transcription factors. *J. Mol. Biol.*, **432** (6), 1801–1815.
- Lavery,R., Moakher,M., Maddocks,J.H., Petkeviciute,D. and Zakrzewska,K. (2009) Conformational analysis of nucleic acids revisited: Curves+. *Nucleic Acids Res.*, **37** (17), 5917–5929.
- Lawson,C.L., Berman,H.M., Chen,L., Vallat,B. and Zirbel,C.L. (2024) The Nucleic Acid Knowledgebase: a new portal for 3D structural information about nucleic acids. *Nucleic Acids Res.*, **52** (D1), D245–D254.
- Lemieux,S. and Major,F. (2002) RNA canonical and non-canonical base pairing types: a recognition method and complete repertoire. *Nucleic Acids Res.*, **30** (19), 4250–63.
- Leontis,N.B. and Westhof,E. (2001) Geometric nomenclature and classification of RNA base pairs. *RNA*, **7** (4), 499–512.

- Li,S., Olson,W.K. and Lu,X.J. (2019) Web 3DNA 2.0 for the analysis, visualization, and modeling of 3D nucleic acid structures. *Nucleic Acids Res.*, **47** (W1), W26–W34.
- Lightfoot,H.L., Hagen,T., Tatum,N.J. and Hall,J. (2019) The diverse structural landscape of quadruplexes. *FEBS Lett.*, **593** (16), 2083–2102.
- Lilley,D.M. (2012) The structure and folding of kink turns in RNA. *WIREs RNA*, **3** (6), 797–805.
- Lindauer,K., Bendic,C. and Sühnel,J. (1996) HBexplore: a new tool for identifying and analysing hydrogen bonding patterns in biological macromolecules. *Bioinformatics*, **12** (4), 281–289.
- Lorenz,R., Bernhart,S.H., Honer Zu Siederdissen,C., Tafer,H., Flamm,C., Stadler,P.F. and Hofacker,I.L. (2011) ViennaRNA Package 2.0. *Algorithms Mol. Biol.*, **6**, 26.
- Lu,X.J. (2020) DSSR-enabled innovative schematics of 3D nucleic acid structures with PyMOL. *Nucleic Acids Res.*, **48** (13), e74.
- Lu,X.J., Babcock,M.S. and Olson,W.K. (1999) Overview of Nucleic Acid Analysis Programs. *J. Biomol. Struct. Dyn.*, **16** (4), 833–843.
- 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.
- Lu,X.J., El Hassan,M. and Hunter,C. (1997a) Structure and conformation of helical nucleic acids: Analysis program (SCHNAAp). *J. Mol. Biol.*, **273** (3), 668–680.
- Lu,X.J., El Hassan,M. and Hunter,C. (1997b) Structure and conformation of helical nucleic acids: Rebuilding program (SCHNArP). *J. Mol. Biol.*, **273** (3), 681–691.
- Lu,X.J. and Olson,W.K. (1999) Resolving the discrepancies among nucleic acid conformational analyses. *J. Mol. Biol.*, **285** (4), 1563–75.
- Lu,X.J. and Olson,W.K. (2003) 3DNA: A software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic Acids Res.*, **31** (17), 5108–5121.
- Lu,X.J. and Olson,W.K. (2008) 3DNA: A versatile, integrated software system for the analysis, rebuilding and visualization of three-dimensional nucleic-acid structures. *Nat. Protoc.*, **3** (7), 1213–1227.
- Lu,X.J., Olson,W.K. and Bussemaker,H.J. (2010) The RNA backbone plays a crucial role in mediating the intrinsic stability of the GpU dinucleotide platform and the GpUpA/GpA miniduplex. *Nucleic Acids Res.*, **38** (14), 4868–76.
- Lu,X.J., Shakked,Z. and Olson,W.K. (2000) A-form conformational motifs in ligand-bound DNA structures. *J. Mol. Biol.*, **300** (4), 819–40.
- Mathews,D.H., Sabina,J., Zuker,M. and Turner,D.H. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288** (5), 911–40.
- McDonald,I.K. and Thornton,J.M. (1994) Satisfying hydrogen bonding potential in proteins. *J. Mol. Biol.*, **238** (5), 777–793.

- Meier,M., Moya-Torres,A., Krahn,N.J., McDougall,M.D., Orriss,G.L., McRae,E.K.S., Booy,E.P., McEleney,K., Patel,T.R., McKenna,S.A. and Stetefeld,J. (2018) Structure and hydrodynamics of a DNA G-quadruplex with a cytosine bulge. *Nucleic Acids Res.*, **46** (10), 5319–5331.
- Mitra,R., Cohen,A.S., Sagendorf,J.M., Berman,H.M. and Rohs,R. (2025) DNaproDB: an updated database for the automated and interactive analysis of protein–DNA complexes. *Nucleic Acids Res.*, **53** (D1), D396–D402.
- Mladek,A., Sponer,J.E., Kulhanek,P., Lu,X.J., Olson,W.K. and Sponer,J. (2012) Understanding the sequence preference of recurrent RNA building blocks using quantum chemistry: the intrastrand RNA dinucleotide platform. *J. Chem. Theory Comput.*, **8** (1), 335–347.
- Nguyen,L.A., Wang,J. and Steitz,T.A. (2017) Crystal structure of Pistol, a class of self-cleaving ribozyme. *Proc. Natl. Acad. Sci.*, **114** (5), 1021–1026.
- Nissen,P., Ippolito,J.A., Ban,N., Moore,P.B. and Steitz,T.A. (2001) RNA tertiary interactions in the large ribosomal subunit: the A-minor motif. *Proc. Natl. Acad. Sci.*, **98** (9), 4899–903.
- Olson,W.K. (1980) Configurational statistics of polynucleotide chains. An updated virtual bond model to treat effects of base stacking. *Macromolecules*, **13** (3), 721–728.
- Olson,W.K., Bansal,M., Burley,S.K., Dickerson,R.E., Gerstein,M., Harvey,S.C., Heinemann,U., Lu,X.J., Neidle,S., Shakked,Z., Sklenar,H., Suzuki,M., Tung,C.S., Westhof,E., Wolberger,C. and Berman,H.M. (2001) A standard reference frame for the description of nucleic acid base-pair geometry. *J. Mol. Biol.*, **313** (1), 229–237.
- Olson,W.K., Li,S., Kaukonen,T., Colasanti,A.V., Xin,Y. and Lu,X.J. (2019) Effects of noncanonical base pairing on RNA folding: structural context and spatial arrangements of G·A pairs. *Biochem.*, **58** (20), 2474–2487.
- Parisien,M., Cruz,J.A., Westhof,E. and Major,F. (2009) New metrics for comparing and assessing discrepancies between RNA 3D structures and models. *RNA*, **15** (10), 1875–1885.
- Pauling,L. and Corey,R.B. (1953) A proposed structure for the nucleic acids. *Proc. Natl. Acad. Sci.*, **39** (2), 84–97.
- Popenda,M., Miskiewicz,J., Sarzynska,J., Zok,T. and Szachniuk,M. (2020) Topology-based classification of tetrads and quadruplex structures. *Bioinformatics*, **36** (4), 1129–1134.
- Premilat,S. and Albiser,G. (2001) A new D-DNA form of poly(dA-dT).poly(dA-dT): an A-DNA type structure with reversed Hoogsteen pairing. *Eur. Biophys. J.*, **30** (6), 404–410.
- Quigley,G. and Rich,A. (1976) Structural domains of transfer RNA molecules. *Science*, **194** (4267), 796–806.
- Rhodes,D. and Lipps,H.J. (2015) G-quadruplexes and their regulatory roles in biology. *Nucleic Acids Res.*, **43** (18), 8627–8637.
- Richardson,J.S., Schneider,B., Murray,L.W., Kapral,G.J., Immormino,R.M., Headd,J.J., Richardson,D.C., Ham,D., Hershkovits,E., Williams,L.D., Keating,K.S., Pyle,A.M., Micallef,D., Westbrook,J. and

- Berman,H.M. (2008) RNA backbone: consensus all-angle conformers and modular string nomenclature. *RNA*, **14** (3), 465–81.
- Saenger,W. (1984) *Principles of nucleic acid structure*. Springer advanced texts in chemistry, first edition, Springer-Verlag, New York.
- Sagendorf,J.M., Markarian,N., Berman,H.M. and Rohs,R. (2020) DNAProDB: An expanded database and web-based tool for structural analysis of DNA-protein complexes. *Nucleic Acids Res.*, **48** (D1), D277–D287.
- Sarver,M., Zirbel,C.L., Stombaugh,J., Mokdad,A. and Leontis,N.B. (2008) FR3D: finding local and composite recurrent structural motifs in RNA 3D structures. *J. Math. Biol.*, **56** (1-2), 215–52.
- Sheng,J., Li,L., Engelhart,A.E., Gan,J., Wang,J. and Szostak,J.W. (2014) Structural insights into the effects of 2'-5' linkages on the RNA duplex. *Proc. Natl. Acad. Sci.*, **111** (8), 3050–3055.
- Shui,X., McFail-Isom,L., Hu,G.G. and Williams,L.D. (1998) The B-DNA dodecamer at high resolution reveals a spine of water on sodium. *Biochem.*, **37** (23), 8341–55.
- Smit,S., Rother,K., Heringa,J. and Knight,R. (2008) From knotted to nested RNA structures: A variety of computational methods for pseudoknot removal. *RNA*, **14** (3), 410–416.
- Stella,S., Cascio,D. and Johnson,R.C. (2010) The shape of the DNA minor groove directs binding by the DNA-bending protein Fis. *Genes Dev.*, **24** (8), 814–826.
- Sussman,D., Nix,J.C. and Wilson,C. (2000) The structural basis for molecular recognition by the vitamin B 12 RNA aptamer. *Nat. Struct. Biol.*, **7** (1), 53–57.
- Tan,D., Piana,S., Dirks,R.M. and Shaw,D.E. (2018) RNA force field with accuracy comparable to state-of-the-art protein force fields. *Proc. Natl. Acad. Sci.*, **115** (7), E1346–E1355.
- Theimer,C.A., Blois,C.A. and Feigon,J. (2005) Structure of the human telomerase RNA pseudoknot reveals conserved tertiary interactions essential for function. *Mol. Cell*, **17** (5), 671–682.
- Tolstorukov,M.Y., Colasanti,A.V., McCandlish,D.M., Olson,W.K. and Zhurkin,V.B. (2007) A novel roll-and-slide mechanism of DNA folding in chromatin: implications for nucleosome positioning. *J. Mol. Biol.*, **371** (3), 725–38.
- van Dam,L. and Levitt,M.H. (2000) BII nucleotides in the B and C forms of natural-sequence polymeric DNA: a new model for the C form of DNA. *J. Mol. Biol.*, **304** (4), 541–561.
- Wahl,M.C., Rao,S.T. and Sundaralingam,M. (1996) The structure of r(UUCGCG) has a 5'-UU-overhang exhibiting Hoogsteen-like trans U•U base pairs. *Nat. Struct. Biol.*, **3** (1), 24–31.
- Wang,H., Mann,P.A., Xiao,L., Gill,C., Galgoci,A.M., Howe,J.A., Villafania,A., Barbieri,C.M., Malinvernii,J.C., Sher,X., Mayhood,T., McCurry,M.D., Murgolo,N., Flattery,A., Mack,M. and Roemer,T. (2017) Dual-targeting small-molecule inhibitors of the staphylococcus aureus FMN riboswitch disrupt riboflavin homeostasis in an infectious setting. *Cell Chem. Biol.*, **24** (5), 576–588.e6.
- Wang,J., Daldrop,P., Huang,L. and Lilley,D.M.J. (2014) The k-junction motif in RNA structure. *Nucleic Acids Res.*, **42** (8), 5322–5331.

- Westhof,E. and Sundaralingam,M. (1983) A method for the analysis of puckering disorder in five-membered rings: the relative mobilities of furanose and proline rings and their effects on polynucleotide and polypeptide backbone flexibility. *J. Am. Chem. Soc.*, **105** (4), 970–976.
- Yang,H., Jossinet,F., Leontis,N., Chen,L., Westbrook,J., Berman,H. and Westhof,E. (2003) Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Res.*, **31** (13), 3450–3460.
- Zhang,J. and Ferré-D'Amaré,A.R. (2013) Co-crystal structure of a T-box riboswitch stem I domain in complex with its cognate tRNA. *Nature*, **500** (7462), 363–366.
- Zhang,N., Gorin,A., Majumdar,A., Kettani,A., Chernichenko,N., Skripkin,E. and Patel,D.J. (2001) V-shaped scaffold: a new architectural motif identified in an A·(G·G·G·G) pentad-containing dimeric DNA quadruplex involving stacked G(anti)·G(anti)·G(anti)·G(syn) tetrads. *J. Mol. Biol.*, **311** (5), 1063–79.
- Zuker,M., Mathews,D.H. and Turner,D.H. (1999) Algorithms and Thermodynamics for RNA Secondary Structure Prediction: A Practical Guide. In *RNA Biochemistry and Biotechnology*, (Barciszewski,J. and Clark,B.F.C., eds), vol. 90. Kluwer Academic Publishers pp. 11–43.