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

**User Manual** (Last updated: November 16, 2018)

modified nucleotide, non-canonical base pair, secondary structure, structural motif, junction loop, kink turn, *G-quadruplex*, pseudoknot, cartoon-block schematics, …

by **Dr. Xiang-Jun Lu** (律祥俊)

Department of Biological Sciences
Columbia University, New York
Contents

1 Introduction 5

2 Download and installation 8

3 Usages 9

3.1 Command-line help 9

3.2 Default run on PDB entry 1msy – detailed explanations 11

3.2.1 Summary section 12

3.2.2 Base pairs 14

3.2.3 Multiplets (higher-order co-planar base associations) 19

3.2.4 Helices 21

3.2.5 Stems (canonical pairs with continuous backbones) 22

3.2.6 Isolated canonical pairs 23

3.2.7 Base stacks 23

3.2.8 Atom-base capping interactions 24

3.2.9 Various loops 24

3.2.10 Single-stranded fragments 26

3.2.11 Secondary structure in dot-bracket notation 26

3.2.12 Structural features per nucleotide 28

3.2.13 Backbone torsion angles and suite names 29

Main chain conformational parameters 29

Virtual torsion angles 31

Sugar conformational parameters 32

Assignment of sugar-phosphate backbone suite names 33

3.3 Default run on PDB entry 1ehz (tRNA\textsuperscript{Phe}) – summary notes 34

3.3.1 Brief summary 34

3.3.2 Modified nucleotides 35

3.3.3 The four triplets 35

3.3.4 Relationship between helices and stems 35

3.3.5 Three hairpin loops 36

3.3.6 One four-way junction loop 36

3.3.7 Splayed-apart conformations 37

3.3.8 Pseudoknot 38
3.4 Default run on PDB entry 1jj2 – four auto-checked motifs ........................................ 39
  3.4.1 Kissing loops .................................................. 41
  3.4.2 A-minor motifs ................................................ 41
  3.4.3 Ribose zippers ............................................... 44
  3.4.4 Kink turns ..................................................... 44
3.5 Identification and characterization of G-quadruplexes .............................................. 46
3.6 Detection and characterization of i-motifs .............................................................. 51
3.7 The --more (i.e., --verbose) option ................................................................. 53
  3.7.1 Extra characterization of base pairs .......................................................... 53
  3.7.2 Orientation of helices/stems ........................................................................... 54
  3.7.3 Base-pair morphology parameters for helices/stems ....................................... 55
3.8 The --non-pair option ......................................................................................... 56
3.9 The --json option ............................................................................................... 58
3.10 The --pair-only option ...................................................................................... 60
3.11 The --nmr option .............................................................................................. 60
3.12 The --cartoon-block option ................................................................................ 61
3.13 The --view option .............................................................................................. 62
3.14 The --blocview option ....................................................................................... 64
3.15 The --frame option ............................................................................................ 64
3.16 The --get-hbnd option ....................................................................................... 65
3.17 Additional options ............................................................................................. 67
  3.17.1 The --u-turn option ...................................................................................... 67
  3.17.2 The --po4 option ......................................................................................... 69
  3.17.3 The --pair-list option .................................................................................. 70
  3.17.4 The --idstr option ....................................................................................... 71
  3.17.5 The --symmetry option ................................................................................ 72
  3.17.6 The --prefix option ..................................................................................... 73
  3.17.7 The --auxfile option .................................................................................... 73
  3.17.8 The --cleanup option ................................................................................... 73
  3.17.9 The --isolated-pair option ........................................................................... 73
  3.17.10 The --torsion360 option ............................................................................ 74
  3.17.11 The --select option ................................................................................... 74
  3.17.12 The --dbn-break option ............................................................................. 74
  3.17.13 The --sugar-pucker option ........................................................................ 74
3.17.14 The --raw-xyz option ............................................. 75
3.17.15 The --helical-axis option ......................................... 75
3.17.16 The --simple-junction option ..................................... 75
3.17.17 The --block-file option .......................................... 76
3.17.18 The --block-color option ........................................ 77
3.17.19 The --block-depth option ......................................... 77
3.17.20 The --hbfile-pymol option ....................................... 77
3.17.21 The --hbfile-jmol option ........................................ 77

4 Pseudoknots detection and removal ........................................ 77
4.1 Higher-order pseudoknots ............................................ 78
4.2 Pseudoknot removal .................................................... 79

5 Resources ........................................................................ 79
5.1 Web interface ............................................................. 79
5.2 DSSR-Jmol integration .................................................. 80
5.3 DSSR-PyMOL integration .............................................. 80
5.4 Online support ........................................................... 80

6 Frequently asked questions ................................................ 83
6.1 What does DSSR stand for? ............................................ 83
6.2 How does DSSR compare with other tools? ......................... 83
6.3 How is DSSR related to 3DNA? ....................................... 84
6.4 How to cite DSSR? ....................................................... 85
6.5 Does DSSR work for DNA? ............................................ 85
6.6 Does DSSR detect RNA tertiary interactions? ..................... 85
6.7 Why is feature X not documented in the manual? ............... 86
6.8 May I ask you questions on DSSR via email? ....................... 87
6.9 Can I run DSSR in parallel? ........................................... 87

7 Citations in peer-reviewed journal articles .............................. 88

8 Integrations into other bioinformatics resources ....................... 90

9 Revision history ................................................................ 91
List of Figures

1 Definitions of key nucleic acid structural components in DSSR 6
2 Three- and two-dimensional structure images of PDB entry 1msy 13
3 Parallel M+N and antiparallel M–N orientations between two nucleobases 15
4 Lenotis-Westhof definition of three edges and the cis/trans orientations 17
5 Issues with the three edges in the Leontis-Westhof definition 18
6 Pictorial definitions of rigid-body base morphology parameters 20
7 Base triplet (GUA) identified in PDB entry 1msy 21
8 Common secondary structural elements of RNA 25
9 Splayed-apart unit with four nucleotides identified in PDB entry 1ehz 38
10 Base pentaplet (AUAAG) identified in PDB entry 1jj2 40
11 Kissing-loop motif identified in PDB entry 1jj2 42
12 Types I and II A-minor motifs identified in PDB entry 1jj2 43
13 Ribose zipper identified in PDB entry 1jj2 45
14 Normal k-turn identified in PDB entry 1jj2 46
15 Schematic representation of G-quadruplexes 48
16 Topological descriptors and V-shaped loops of G-quadruplexes 50
17 Cartoon-block representation of i-motifs 52
18 Additional parameters calculated with the --more option 54
19 Four representative DSSR-PyMOL cartoon-block images 63
20 Three different types of U-turns 68
21 Screenshot of the DSSR-Jmol integration 81
22 DSSR-Jmol featured in the cover image of the NAR’17 web-server issue 82
1 Introduction

As the number of experimentally solved RNA-containing structures grows, it is becoming increasingly important to characterize the geometric features of the molecules consistently and efficiently. Existing RNA bioinformatics tools are fragmented, and suffer in either scope or usability. DSSR [1] is an integrated software tool for Dissecting the Spatial Structure of RNA, designed from ground up to streamline the analyses of 3D RNA structures. Figure 1 outlines some key algorithms underlying the DSSR program.

Starting from an RNA structure in Protein Data Bank (PDB) or PDBx/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 (Figure 1B,C) [2] and a set of simple geometric criteria (Figure 1D) to identify all existent base pairs (bp): 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 bp parameters (shear, stretch, stagger, propeller, buckle, and opening) to rigorously quantify the spatial disposition of any two interacting bases. Where applicable, the program also denotes a bp by common names (e.g., WC, reverse WC, Hoogsteen A+U, reverse Hoogsteen A–U, wobble G–U, sheared G–A, imino G–A, Calcutta U–U, dinucleotide platform), the Saenger classification scheme [3] of 28 H-bonding types, and the Leontis-Westhof nomenclature [4] of 12 basic geometric classes.

DSSR detects multiplets (triplets or higher-order base associations) by searching horizontally in the plane of the associated bp for further H-bonding interactions. The program determines double-helical regions (Figure 1E) by exploring vertically in the neighborhood of selected bps for base-stacking interactions, regardless of backbone connection (e.g., coaxial stacking of helices). 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 bp sequence (.bpseq) – that can be fed directly into visualization tools such as VARNA [5]. 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-form double helices (where appropriate) and quantified by helical parameters. DSSR calculates commonly used back-
Figure 1: Definitions of key nucleic acid structural components in DSSR (Reproduced from Figure 1 of the DSSR paper [1]). (A) Nucleotides are recognized using standard atom names and base planarity. This method works for both the standard bases (A, C, G, T and U), and those of modified nucleotides, regardless of their tautomeric or protonation states. (B) Bases are assigned a standard reference frame (25) 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 co-planarity of base rings and the occurrence of hydrogen bonds. This geometric algorithm can find canonical (Watson-Crick and G–U wobble) as well as non-canonical pairs. Higher-order (three or more) co-planar 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. The same algorithm is applied to identify continuous base stacks that are outside of helical regions, by using bases instead of pairs as the assembly unit. Nucleotides not involved in base-stacking interactions are collected into one separate group. A stem is defined as a special type of helix, made up of canonical pairs and with a continuous backbone along each strand. Coaxial stacking is defined by the presence of two or more stems within one helix. An isolated canonical pair is one that is not contained within a stem. (F) ‘Closed’ loops of various types (hairpin, bulge, internal, and junction loops) are delineated by stems or isolated pairs, and specified by the lengths of the intervening, consecutive nucleotide segments. A kissing-loop motif entails formation of one or more canonical pairs between the bases in different hairpin loops. Single-stranded segments that lie outside loops are separately listed.
bone torsion angles (including the virtual $\eta/\theta$ torsions), classifies the backbone into BI/BII conformations and the sugar into $C2'/C3'$-endo like puckers, and assigns the consensus RNA backbone suite names following Richardson et al. [6]. The program also identifies A-minor interactions, splayed-apart dinucleotide conformations, atom-base capping interactions, ribose zippers, G quadruplexes, i-motifs, kissing loops, U-turns, and k-turns etc. Furthermore, DSSR reports non-pairing interactions (H-bonding or base-stacking) between two nucleotides, and contacts involving phosphate groups.

DSSR has been integrated to Jmol, a widely used Java viewer for 3D structures. The DSSR-Jmol integration [7] bridges the DSSR command-line analyzing tool and the Jmol molecular viewer together via a simple JSON interface and a powerful query language. Users can now select DSSR-derived RNA structural features (such as base pairs, double helices, and various loops) as easily as they can select protein $\alpha$-helices and $\beta$-strands. Moreover, fine-grained characteristics of these features can be queried via Jmol SQL for DSSR (see the DSSR-Jmol manual). Notably, the novel representation styles (step diagram and base blocks) and coloring schemes bring RNA visualization to an entirely new level (see Figure 3 of the DSSR-Jmol paper).

DSSR is a new component of the 3DNA suite of programs [8, 9] for the analysis, rebuilding, and visualization of three-dimensional nucleic acid structures. It has been created from scratch by employing my extensive experience in supporting 3DNA, increased knowledge of RNA structures, and refined C programming skills. Overall, DSSR consolidates, refines, and significantly extends the functionality of 3DNA (in ‘find_pair’ and ‘analyze’) for RNA structural analysis.

The software has a set of unique features not available elsewhere (to the best of my knowledge). By connecting dots in RNA structural bioinformatics, it makes many common tasks simple and advanced applications feasible. DSSR is efficient and robust, with sensible default settings and an intuitive output; thus it is accessible to a much broader audience than the classic 3DNA distribution (up to version 2.3).

There is actually more to DSSR than meets the eye. To target the widest possible user base, I’ve deliberately omitted advanced/technical features in the manual (which is already over 90 pages). Moreover, many documented options have additional variations that may be of interests to some applications. If you feel that a relevant functionality should be there but missing from the manual, Simply ask on the 3DNA Forum.

DSSR is being actively maintained and developed. As always, I greatly appreciate user feedback. So far, all reported bugs have been promptly responded and fixed where
appropriate. The program has been checked against all nucleic acid-containing entries in the PDB, without any known issues. Simply put, I strive to make DSSR a practical tool that the user community can count on.

When reporting bugs or submitting feature requests, please provide the specific DSSR version you are using (with `-v` or `--version`). For example, the version that the manual is based upon corresponds to (`x3dna-dssr -v`):

```
******************************************************************
DSSR: an Integrated Software Tool for
Dissecting the Spatial Structure of RNA
v1.8.4-2018nov12, by xiangjun@x3dna.org
This program is being actively maintained and developed. As always,
I greatly appreciate your feedback! Please report all DSSR-related
issues on the 3DNA Forum (forum.x3dna.org). I strive to respond
promptly to any questions posted there.
******************************************************************
```

## 2 Download and installation

DSSR has been implemented in ANSI C as a standalone command-line program. It is self-contained and the executable version (on macOS, Linux and Windows) is small (~1MB), with zero runtime dependencies on third-party libraries. Getting DSSR up and running is very straightforward:

1. Register at the 3DNA Forum to download DSSR and ask questions.

2. Log in, at the top-left corner under “Welcome”, click “Downloads” and “3DNA download”. Proceed to download the DSSR executable (named ‘`x3dna-dssr`’ for Linux/macOS\(^1\) or ‘`x3dna-dssr.exe`’ for Windows) for your operating system and architecture. Currently, compiled versions are available for the most common systems: Linux (32-bit and 64-bit), Intel-based macOS, and Windows (also runs well under Cygwin and MinGW/MSYS).

\(^1\)On macOS with the default Safari browser, the extra `.dms` extension may be added to the downloaded DSSR file. Thus instead of `x3dna-dssr`, one gets `x3dna-dssr.dms`. To avoid confusions in following the manual, please remove the `.dms` extension by running: `mv x3dna-dssr.dms x3dna-dssr`. Note that the Chrome (or Firefox) browser does not have this problem, i.e., the downloaded file is simply `x3dna-dssr`, without any extra extension.
3. Preferably, move DSSR into a folder on your command search path (e.g., ~/bin) so you can run the program conveniently. Or, you can simply put DSSR anywhere (e.g., your current working directory) and specify the path explicitly to execute the program.

4. On macOS or Linux, run the command `chmod u+x x3dna-dssr` to make DSSR executable. On Windows, this step is not necessary.

5. Type `x3dna-dssr -h` to verify your installation.

As of version 1.0.3 (released on March 9, 2014), DSSR is stable in terms of basic functionality and main output format. Where possible, I will try to maintain backward compatibility in future v1.x releases. If you only need fundamental features, you may not need to bother with frequent updates. DSSR is designed with simplicity and robustness in mind: get the job done, and then stay out of the way. On the other hand, DSSR is being actively maintained and developed, with new features added and known bugs fixed. Users are advised to keep up to date: it is as simple as downloading DSSR again from the 3DNA Forum to replace (i.e., overwrite) the old copy.

3 Usages

Although the above introduction summarizes DSSR’s major functionality in dry, ‘abstract’ technical terms, using DSSR effectively is straightforward. This is best illustrated with concrete examples.

3.1 Command-line help

As is the norm of Linux/Unix command-line tools, running DSSR with -h (or --help) provides help information, a portion of which is given below (`x3dna-dssr -h`):

<table>
<thead>
<tr>
<th>Usage:</th>
<th>x3dna-dssr [options]</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Options:</td>
<td></td>
</tr>
<tr>
<td>--help</td>
<td>Print this command-line help information (-h)</td>
</tr>
<tr>
<td>--version</td>
<td>Print version number and exit (-v)</td>
</tr>
<tr>
<td>--citation</td>
<td>Print preferred citation(s) and exit (--cite)</td>
</tr>
<tr>
<td>--input=file</td>
<td>Specify a PDB/mmCIF file for analysis (-i=file)</td>
</tr>
<tr>
<td>--output=file</td>
<td>Designate the main DSSR output file (-o=file)</td>
</tr>
</tbody>
</table>
The above help message should be sufficient to get most users started with DSSR. It is worth noting that the command-line interface is consistent on all operating systems, including the command shell on Windows.

DSSR introduces a consistent and flexible way to process command-line options. Here, each option can be specified via a --key[=value] pair, or -key[=value] or key[=value]; i.e., 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 'lmsy', with default output to stdout
x3dna-dssr --input=lmsy.pdb

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

#3-6, same as #2
x3dna-dssr --output=lmsy.out --input=lmsy.pdb
x3dna-dssr --output=lmsy.out --input=lmsy.pdb
x3dna-dssr --output=lmsy.out --input=lmsy.pdb
x3dna-dssr output=lmsy.out --input=lmsy.pdb

#7 the value 'lehz.pdb' overwrites 'lmsy.pdb'
x3dna-dssr --input=lmsy.pdb input=lehz.pdb

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

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

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

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

#16-19 within a key, separator dash(~) and underscore (_)
# are treated the same, and can be omitted
```
3.2 Default run on PDB entry 1msy – detailed explanations

The PDB entry 1msy is a small RNA fragment with 27 nucleotides, containing a GUAA tetraloop (GNRA-type) mutant of the sarcin/ricin domain from *Escherichia Coli* 23 S rRNA [10] that includes a bulged-G motif (the GpU dinucleotide platform [11]). This structure contains several features that nicely illustrate fundamental aspects of DSSR.

Let the PDB-formatted 3D coordinate file be called ‘1msy.pdb’ (as shown below, the PDBx/mmCIF version ‘1msy.cif’ works as well), the output file ‘1msy.out’, and leave all other options in their default settings. Then simply run the command:

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

The screen output provides a brief summary of the run, as shown below. Note that DSSR starts from a raw .pdb or .cif file as directly downloaded from the PDB, and takes little time to analyze small to mid-size (non-ribosomal) structures.

| 1 | total number of nucleotides: 27 |
| 2 | total number of base pairs: 13 |
| 3 | total number of multiplets: 1  |
| 4 | total number of helices: 1    |
| 5 | total number of stems: 1      |
| 6 | total number of isolated WC/wobble pairs: 1 |
| 7 | total number of atom-base capping interactions: 2 |
| 8 | total number of splayed-apart dinucleotides: 1 |
| 9 | total number of hairpin loops: 1 |
|10 | total number of internal loops: 1 |
|11 | total number of non-loop single-stranded segments: 2 |
|12 |                                  |
|13 | Time used: 00:00:00:00          |

For easy reference, Figure 2 shows the 3D structure (A) and the corresponding secondary (2D) structure (B) of 1msy. Here the schematic 3D structure was automatically generated in schematic blocview representation (see also Section 3.12 and Section 3.14),
and the 2D diagram was rendered with VARNA [5], using DSSR-generated 2D structure in dot-bracket notation, connectivity table (.ct) or .bpseq format (see below).

The main output file (‘1msy.out’) contains many sections. We will go over them one by one, along the way, explaining the notations used therein. Additionally, DSSR generates the following auxiliary files (named with prefix ‘dssr-’ by default):

- dssr-stems.pdb -- an ensemble of stems
- dssr-helices.pdb -- an ensemble of helices (coaxial stacking)
- dssr-pairs.pdb -- an ensemble of base pairs
- dssr-multiplets.pdb -- an ensemble of multiplets
- dssr-hairpins.pdb -- an ensemble of hairpin loops
- dssr-iloops.pdb -- an ensemble of internal loops
- dssr-2ndstrs.bpseq -- secondary structure in bpseq format
- dssr-2ndstrs.ct -- secondary structure in connectivity table format
- dssr-2ndstrs.dbn -- secondary structure in dot-bracket notation
- dssr-torsions.txt -- backbone torsion angles and suite names
- dssr-splays.pdb -- an ensemble of splayed-apart units
- dssr-stacks.pdb -- an ensemble of stacks
- dssr-atom2bases.pdb -- an ensemble of atom-base stacking interactions

3.2.1 Summary section

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

Date and time: Fri Nov 16 12:50:33 2018

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 in the Summary section first explains how a nucleotide (nt) is specified in the output file. Typically, a chain id, residue name, and sequence number are sufficient to uniquely 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 shorthand name. For RNA, the four standard nts in three-letter form – ‘⊔⊔A’, ‘⊔⊔C’, ‘⊔⊔G’ and ‘⊔⊔U’ (where ⊔ stands for a space) – are shortened to
Figure 2: Images of PDB entry 1msy. (A) Schematic three-dimensional structure in blocview presentation (see Section 3.14). (B) Two-dimensional diagram produced with VARNA using DSSR-derived secondary structure information.

A, C, G, and U, respectively. For DNA, the three-letter (one-letter) nts are ‘⊔DA’ (A), ‘⊔DC’ (C), ‘⊔DG’ (G), and ‘⊔DT’ (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 prevalent modified nt in RNA, is denoted ‘P’ in DSSR since the small case ‘p’ is reserved for potential modified pseudouridines.

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

---

^2Not to be confused with the phosphorus atom in the backbone phosphate group. In fact, the distinction should be clear in context.
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 meaning of the ‘Date and time:’ line is obvious. The ‘File name:’ line lists the data file analyzed by DSSR, followed by a list of numbers: DNA/RNA chains (here chain ‘A’ with 27 nts), nts (27), atoms (685), waters (109), and metals (0). In cases with more than one chain or type of metal, the output would be as below (using 1jj2 as an example).

<table>
<thead>
<tr>
<th>no. of DNA/RNA chains:</th>
<th>2 [0=2754, 9=122]</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. of metals:</td>
<td>210 [Na=86, Mg=117, K=2, Cd=5]</td>
</tr>
</tbody>
</table>

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 atoms: 86 Na, 117 Mg, 2 K, and 5 Cd.

### 3.2.2 Base pairs

DSSR identifies a total of 13 bps, including not only the canonical Watson-Crick (WC) and wobble pairs, but also non-canonical bps (a dinucleotide platform, reverse Hoogsteen, sheared G–A, etc.).

<table>
<thead>
<tr>
<th>List of 13 base pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>nt1</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
</tbody>
</table>

As the header line (no. 2 in the listing) shows, each bp is characterized by the two constituent nts (nt1 and nt2), and an abbreviated bp type of the form M±N. Here M and N are the one-letter shorthand form of the nt names, which can be in upper or lower cases (for modified nts), as noted above. The symbol ± reflects the two possible relative ‘face’ orientations between the two planar base moieties: the sign is normally ‘−’ as for
canonical WC/wobble bps with opposite faces, and ‘+’ if one of the two base planes is flipped. Figure 3 provides further details on the rationale of the M±N convention [11] that is used consistently in 3DNA [8, 9] and DSSR.

![Figure 3: Comparison of the parallel M+N and antiparallel M–N arrangements of G and U bases found respectively in (A) the G+U platform and (B) the G–U wobble pair. Here G is depicted in green, and U in cyan. For easy comparison, the two pairs are oriented in the standard reference frame of the guanine base, with the shading used to specify the face from which the positive base normal, i.e., z-axis, emanates. The U shares the same face as G in the G+U dinucleotide platform (A); the z-axes of the two bases are roughly parallel, forming the positive dot product expected of M+N pairing. By contrast, the U points in the opposite direction from G in the G–U wobble pair (B); the z-axes of the two bases are roughly anti-parallel, forming the negative dot product expected of M–N pairing. Note that this M–N nomenclature follows the convention expected for the canonical, antiparallel Watson-Crick base pairs: A–U/A–T and G–C.](image)

The name column in the list of base pairs gives the common names of the corresponding bps, where appropriate. Currently, the list includes WC, reverse WC (rWC), Hoogsteen,
reverse Hoogsteen (rHoogsteen), wobble, sheared, imino, Calcutta, and dinucleotide platform, and unnamed bps are designated ‘--’. Other than WC and wobble pairs, the PDB entry 1msy also contains a GpUpA/GpA miniduplex [11] characterized by three common bps (pairs 9 to 11, lines 11-13 in the listing): a platform (G+U), a reverse Hoogsteen (U–A), and a sheared (A–G). However, it is difficult to connect those bp structures with their familiar names, even for a seasoned scientist. With DSSR-assigned names, one can immediately see common bps and their types within a structure. As it turns out, the name column is also handy to delineate quickly stretches of canonical pairs (e.g., bps 2 to 6, lines 4-8 in the listing on Page 14) that form double-helical regions, termed ‘stem’ in DSSR (see Page 22).

The Saenger column provides the classification of 29 bp types with at least two H-bonds between base atoms. This classic list, initially compiled by Saenger [3] for 28 bps, includes an extra G+C bp added to the list by Burkard et al. in an appendix to the second edition of “The RNA World” [12]. Note that in addition to the Roman numerals (I to XXIX) originally used by Saenger, the DSSR output also includes 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 ‘n/a’ is assigned.

The LW column in the list of base pairs (on Page 14) gives Leontis-Westhof classification of bps [4]. As illustrated in Figure 4 (see below), this classification 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 “gives rise to 12 basic geometric types with at least two H bonds connecting the bases” [4]. This geometry-based method captures salient features of bp interactions and strikes a balance between simplicity and expressiveness. The LW scheme is more generally applicable than the Saenger classification, and it is easy to grasp by biologists. As a result, the LW bp classification has become standard in RNA structural bioinformatics.

Strictly speaking, however, the RNA-centric LW classification has its limitations (Figure 5). For one thing, the Sugar edge explicitly includes the 2′-hydroxyl group, rendering it less applicable to DNA structures. Moreover, while the aromatic base can be taken as a

---

3 See my blogpost titled “The Calcutta U-U base pair” for details.
4 It certainly took me a while to become familiar with the rHoogsteen U–A bp and the sheared A–G pair.
5 See my blogpost titled “Number of base pairs with at least two inter-base H-bonds: 28 or 29?” for further details.
rigid body with three fixed edges, the $\chi$ (chi) torsion angle characterizes the internal freedom between base and sugar (anti/syn). When $\chi$ is in the relatively rare but by no means uncommon syn conformation, the Sugar edge, defined by the common anti conformation, no longer seems to exist.

The rich variety of RNA bps 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 [13] were the first to extend the LW nomenclature. We noted the importance of the out-of-plane backbone 'edge' formed by an RNA-specific H-bond between O2'(G) and OP2(U) [11]. The RNA 3D Hub website, hosted by the Leontis-Zirbel co-lab, now lists non-standard bp interactions ncSW, ntSH, and ntHH for the 1msy entry.

The DSSR bp classification is presented in the last column (e.g., ‘cW-W’ for canonical bps, and ‘cm+M’ for the G+U dinucleotide platform). Overall, the DSSR scheme follows the
Figure 5: Example cases where the original Leontis-Westhof definition of the three edges does not strictly apply. The actual results of LW/DSSR classifications are those implemented in the DSSR program. In each figure, H-bonds are shown in dashed lines, C2' atoms in black dots, and O2' atoms labeled. (A) Watson-Crick C-G pair in 355d, a standard B-DNA molecule. The ‘sugar’-edge for C or G does not have an O2' atom. (B) Hoogsteen A+U pair in 1jj2. Here, nucleotide 0.A45 is in the syn conformation; thus the O2' atom is pointing away from (instead of towards) the minor-groove edge of the base. (C) G+U platform (in 1msy) has an out-of-plane backbone “edge”. (D) U-C pair (in 1msy) has only one ‘corner’ H-bond, and the interacting edge for A.C2667 cannot be unambiguously assigned.

Pattern of [ct][WMm]±[WMm] (see Figure 1C). Here, [ct] stands for cis/trans, defined by the positioning of the glycosidic bonds related to a line connecting the centers of base-ring atoms; [WMm] for the interacting edges defined with reference to an idealized WC bp: ‘W’ for the Watson-Crick edge, ‘M’ for the Major-groove edge, ‘m’ for the minor-groove edge; ± for normal (‘−’) vs. flipped (‘+’) base orientations, as noted above. The dot symbol denotes cases where edges or orientations cannot be defined (see bp #7 in the list of base pairs on Page 14, and Figure 5(D)).

In general, the ‘M’ in DSSR corresponds to the Hoogsteen/CH-edge (H) in the LW notation, and the ‘m’ to the LW Sugar-edge (S) if χ is in the anti conformation. In the current implementation, we assumes direct correspondences between M/H, and m/S,
regardless of the anti/syn base-sugar orientations. Moreover, the LW cis/trans is assigned the same way as for DSSR. So both notations are strictly parallel, except for the extra ± character in DSSR.

The M±N bp notation has been used consistently in 3DNA for over a decade, and a whole section titled ‘Base pair parameters’ is included in the first 3DNA publication [8]. The orientation, combined with the six standard bp parameters (shear, stretch, stagger, buckle, propeller, and opening; see Figure 6) derived from the 3DNA ‘analyze’ program, can unambiguously characterize any pair. Conversely, given the M±N notation with six corresponding bp parameters, the 3DNA ‘rebuild’ program can rigorously reconstruct the spatial disposition of the two interacting bases. This reversibility is one of the unique features of 3DNA and can be applied to bps in both DNA and RNA.

Even with the increasing popularity of 3DNA, the value of using M±N with six parameters to classify bps has never received the attention it deserves, especially in the RNA structural bioinformatics community. The incorporation of bp classifications in DSSR provides an opportunity to emphasize the strength of this approach. In DSSR, the three interacting edges strictly center on a base, which can be taken as a rigid body. The standard base reference frame [2] has distinct geometric features to allow for easy identification of the edges (Figure 1C). As is also clear from Figure 1C, for WC pairs, the distinction of the minor-groove vs. the major-groove edges is simply a consequence of the asymmetric glycosidic linkage between the base and sugar moiety. Moreover, the terminology of minor/major grooves is widely used, even in the RNA structural literature – the most obvious being the A-minor motif [14] (see Page 41).

DSSR also produces a file named ‘dssr-pairs.pdb’ containing MODEL/ENDMDL delineated atomic coordinates for each identified bp, expressed in its own reference frame.

3.2.3 Multiplets (higher-order co-planar base associations)

There is only one multiplet, a triplet, in 1msy, as shown below.

<table>
<thead>
<tr>
<th>List of 1 multiplet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 nts=3 GUA A.U2655,A.U2656,A.A2665</td>
</tr>
</tbody>
</table>

For each identified multiplet, the serial number (1) is followed by the number of nts (3), the base sequence in one-letter shorthand form (GUA), and a comma-delimited list of the corresponding nts. The G-tetrad motif, which forms the G-quadruplexes, is also detected.
Figure 6: Pictorial definitions of rigid-body parameters used to describe the geometry of complementary (or non-complementary) base pairs and sequential base pair steps. 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 pair planes 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.
with DSSR (although not in 1msy); it is simply a special multiplet with four Gs.

As for bps, DSSR also generates a file (named ‘dssr-multiplets.pdb’ by default) for MODEL/ENDMDL delineated multiplets. In the file, each multiplet is set to the most extended view with respect to the base rings. The triplet in 1msy is shown in Figure 7. Note the extensive network of H-bonding interactions between the three nts [11].

![Image](image-url)

**Figure 7:** The GUA triplet identified in 1msy. Here A.G2655 and A.U2656 form a G+U dinucleotide platform; A.U2656 and A.A2665 form a reverse Hoogsteen pair. The phosphate group of A.A2665 interacts with the Watson-Crick edge of A.G2655 with two sequence-specific H-bonds. The image was produced with Jmol.

### 3.2.4 Helices

DSSR identifies one helix with 12 bps in 1msy (see below). This section starts with a note explaining the definition of a helix, and other related information. By referring to Figure 2(A), one can immediately see the duplex formed by base-stacking interactions, starting from the very bottom of the structure all the way up to the sheared G–A pair at
the top (part of the GUAA tetraloop). Note that by definition, a helix is composed of at least two stacked bps.

List of 1 helix

Note: a helix is defined by base-stacking interactions, regardless of bp type and backbone connectivity, and may contain more than one stem.

helix#number[stems-contained] bps=number-of-base-pairs in the helix
bp-type: '|' for a canonical WC/wobble pair, '.' otherwise
helix-form: classification of a dinucleotide step comprising the bp above the given designation and the bp that follows it. Types include 'A', 'B' or 'Z' for the common A-, B- and Z-form helices, '.' for an unclassified step, and 'x' for a step without a continuous backbone.

<table>
<thead>
<tr>
<th>helix #</th>
<th>bps</th>
<th>strand-1</th>
<th>strand-2</th>
<th>helix-form</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>5'-UGCUCCUAUACG-3'</td>
<td>3'-GUGAGGCCAGGA-5'</td>
<td>AAA.x...</td>
</tr>
<tr>
<td>1 A.U2647</td>
<td>A.G2673</td>
<td>U-G --</td>
<td>n/a</td>
<td>cWW cW-W</td>
</tr>
<tr>
<td>2 A.G2648</td>
<td>A.U2672</td>
<td>G-U Wobble</td>
<td>28-XXVIII</td>
<td>cWW cW-W</td>
</tr>
<tr>
<td>3 A.C2649</td>
<td>A.G2671</td>
<td>C-G WC</td>
<td>19-XIX</td>
<td>cWW cW-W</td>
</tr>
<tr>
<td>4 A.U2650</td>
<td>A.A2670</td>
<td>U-A WC</td>
<td>20-XX</td>
<td>cWW cW-W</td>
</tr>
<tr>
<td>5 A.C2651</td>
<td>A.G2669</td>
<td>C-G WC</td>
<td>19-XIX</td>
<td>cWW cW-W</td>
</tr>
<tr>
<td>6 A.C2652</td>
<td>A.G2668</td>
<td>C-G WC</td>
<td>19-XIX</td>
<td>cWW cW-W</td>
</tr>
<tr>
<td>7 A.U2653</td>
<td>A.C2667</td>
<td>U-C --</td>
<td>n/a</td>
<td>tW tW-M</td>
</tr>
<tr>
<td>8 A.A2654</td>
<td>A.C2666</td>
<td>A+C --</td>
<td>n/a</td>
<td>tHH tM+M</td>
</tr>
<tr>
<td>9 A.U2656</td>
<td>A.A2665</td>
<td>U-A Hoogsteen</td>
<td>24-XXIV</td>
<td>tWH tW-M</td>
</tr>
<tr>
<td>10 A.A2657</td>
<td>A.G2664</td>
<td>A-G Sheared</td>
<td>11-XI</td>
<td>tHS tM-m</td>
</tr>
<tr>
<td>11 A.C2658</td>
<td>A.G2663</td>
<td>C-G WC</td>
<td>19-XIX</td>
<td>cWW cW-W</td>
</tr>
<tr>
<td>12 A.G2659</td>
<td>A.A2662</td>
<td>G-A Sheared</td>
<td>11-XI</td>
<td>tSH tM-M</td>
</tr>
</tbody>
</table>

Of the 13 bps listed on Page 14, only the G+U dinucleotide platform formed by A.G2655 and A.U2656 is not contained in the helix. Since A.U2656 also forms a reverse Hoogsteen U–A pair with A.A2665 (as part of the triplet, see Figure 7), and the U–A bp is included in the helix, just the so-called bulged G (i.e., A.G2655) is omitted.

The helix-form subsection (lines 13-16 starting with 'strand-1', 'by-type', 'strand-2', and 'helix-form', respectively) gives a quick summary of the stacked bps: base sequences, bp types (canonical or otherwise), and helix conformations 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 (see FAQ on Page 85), the B- and Z-forms are also included in the classification.

3.2.5 Stems (canonical pairs with continuous backbones)

In DSSR, a stem is defined as a helix consisting of only canonical WC/wobble pairs and possessing a continuous backbone along each strand. The literature does not appear to be consistent as to what constitutes a helix, stem, or arm. Hopefully DSSR will help to
clarify the confusion in the field. The default requirement for canonical bps in DSSR follows the
convention widely adopted for RNA secondary structures, as in the mfold/UNAFold
software and the ViennaRNA Package. The PDB entry 1msy contains one stem with six
canonical bps, as shown below (see also Figure 2(B)). Notice how the bp names and the
helix-form sub-section on Page 21 facilitate a quick visual identification of the stem in the helix.

3.2.6 Isolated canonical pairs

DSSR defines an isolated canonical bp as one that is not part of a stem. In 1msy, A.C2658 and A.G2663 form such an isolated C–G WC pair, as detailed below. The [#1] at the beginning means this pair is part of helix #1. Note also the negative indices for isolated canonical bps vs. the positive values for stems. The significance of the distinction will become obvious later on, as internal and hairpin loops are specified.

3.2.7 Base stacks

DSSR defined a base stack as an ordered list of nucleotides assembled together via base-
stacking interactions, regardless of backbone connectivity or pairing involvement. Stacking interactions within a stem are not included by default. In 1msy, note the UAA stack (no. 1
in the listing) in the GUAA tetraloop, and the GGGG stack (no. 3) next to it (Figure 2(A)). Nucleotides not involved in stacking interactions are also listed, if any.

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

3.2.8 Atom-base capping interactions

First described by Quigley and Rich in the tRNA\(^{\text{Phe}}\) structure [15], the phosphate group (actually the exocyclic OP2 atom) can stack over a base ring to cap a helix. Atom-base capping interactions are often observed in other structural motifs, including U-turns or GNRA tetraloops.

In DSSR, the stacking atoms also includes oxygen from water or the sugar moiety. The output for 1msy is given below. Check also the associated PDB file ‘dssr-a2bases.pdb’ to visualize the findings.

<table>
<thead>
<tr>
<th>List of 2 atom-base capping interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>dv: vertical distance of the atom above the nucleotide base</td>
</tr>
<tr>
<td>type</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1 phosphate</td>
</tr>
<tr>
<td>2 sugar</td>
</tr>
</tbody>
</table>

Here the type column can be ‘phosphate’, ‘sugar’, or ‘other’. The meaning of the atom and nt columns should be obvious. As noted on line 2, the last dv column gives vertical distance (in Å) of the atom above the nucleotide base. Thus, the first example (on line 5 in the listing) shows that the OP2 atom of the A2661 phosphate group stacks 3.04 Å over the G2659 base ring.

3.2.9 Various loops

Commonly occurring RNA loops are illustrated in Figure 8. DSSR identifies all of these different types and further distinguishes symmetric vs. asymmetric internal loops.

DSSR finds one hairpin loop and an asymmetric internal loop in 1msy, as detailed below. Note that the list of the 13 nts in the internal loop is too long to fit on this page.

---

DSSR-Jmol · DSSR-PyMOL · DSSR Web Interface
**Figure 8:** Common secondary structural elements of RNA. Single-stranded fragments at the 5‘ and 3‘ ends, and double-helical stems are obvious. ‘Closed’ loops of various types (hairpin, bulge, internal, and junction loops) are labeled.

The list is actually written on the same line as ‘nts=13 CUAGUACGGACCG’.

---

**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.

---

**List of 1 hairpin loop**
1 hairpin loop: nts=6; [4]; linked by [#-1]

- **summary:** [1] 4 [A.2658 A.2663] 1
- nts=6 CUAGU A. C2658 , A. U2659 , A. A2660 , 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

With the note at the top and by referring to Figure 2(B), it should be straightforward to understand the meaning of most of the items in this section. Nevertheless, the two sets of matched brackets are worth further explanation. The hairpin loop contains a total of 6 nts, including the closing canonical pair: the [4] means 4 nts, i.e., a tetraloop; the [#-1]
indicates that the tetraloop is closed by the first isolated pair. The asymmetric internal loop contains 13 nts in total: the \([5,4]\) means 5 nts along one strand, and 4 nts on the other; the \([#1,#-1]\) indicates the internal loop is linked by the first stem and the first isolated pair.

As of DSSR v1.5.8-2016jul09, a “summary” line is added (by default) for each loop. Here is the first set of matched brackets contains the number of stems (including isolated canonical pairs) delineating the loop: 1 for hairpin loops, 2 for bulges and internal loops (\([2]\) for the above internal loop), and 3+ for junctions loops. The following set of numbers corresponds to the bridging nucleotides (e.g., \([5\ 4]\) for the above internal loop). The second set of matched brackets lists the closing canonical pairs in `chain-id.residue-number` format (e.g., \([A.2652\ A.2668\ A.2658\ A.2663]\) for the above internal loop). The next set of numbers are the corresponding lengths of the stems delineating the loop (e.g., \(6\ 1\) for the above internal loop). The meaning of the various components should be obvious by referring to Figure 2(B). 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].

### 3.2.10 Single-stranded fragments

DSSR also characterizes single-stranded fragments not included in various loop regions. For 1msy, there is none (see also Figure 2(B)).

### 3.2.11 Secondary structure in dot-bracket notation

The DSSR-derived secondary structure is written in an extended dot-bracket notation (dbn) with information for pseudoknots (as matched \([\],\ \{\},\ or \(<\>\) pairs etc) and chain breaks (as \&s), which can be fed directly into VARNA \[5\]. The dbn is written in FASTA format where the title line (with a ‘\>' on the first column) is followed by base sequence and the secondary structure, each on a separate line. The dbn for 1msy is as follows.

1 Secondary structures in dot-bracket notation (dbn) as a whole and per chain
2 >1msy nts=27 [whole]
3 UGCUCCUAGUACGUAAGGACCGGAGUG
4 .((((((.....(....)....)....))))).
5 >1msy-A #1 nts=27 0.30(2.47) [chain] RNA
6 UGCUCCUAGUACGUAAGGACCGGAGUG
7 .((((((.....(.....).....))))).

Since 1msy contains only a single (and continuous) chain (‘A’), the dbn contents for the whole and per chain are the same. For the Dickerson DNA dodecamer \[16\] structure
355d, the difference in contents between the whole structure and each chain is obvious from the directions of the brackets (see below).

```
Secondary structures in dot-bracket notation (dbn) as a whole and per chain

>355d nts=24 [whole] CGCGAATTCGCG & CGCGAATTCGCG (((((((((((&)))))))))))))
>355d-A #1 nts=12 3.21(0.51) [chain] DNA CGCGAATTCGCG ((((((((((((
>355d-B #2 nts=12 3.37(0.50) [chain] DNA CGCGAATTCGCG )))))))))))))
```

DSSR also generates a file named `dssr-2ndstrs.dbn` containing the secondary structure for the whole molecule in dbn notation. For better connection to the 2D world, DSSR produces an additional file named `dssr-2ndstrs.ct`, which expresses the secondary structure in connectivity table (.ct) format. First introduced by the mfold program, the .ct format is one of the most commonly used RNA secondary structure formats. The .ct output file for 1msy is listed below. See my blogpost titled “DSSR-derived secondary structure in .ct format” for further details. Moreover, DSSR also generates a secondary structure representation in the .bpseq format in a file named `dssr-2ndstrs.bpseq`.

```
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.U2650
3  C  2  4  25  2649 # name=A.C2649, pairedNt=A.G2650
4  U  3  5  24  2650 # name=A.U2650, pairedNt=A.C2649
5  C  4  6  23  2651 # name=A.C2651, pairedNt=A.G2669
6  G  5  7  22  2652 # name=A.G2652, pairedNt=A.C2651
7  U  6  8  21  2653 # name=A.U2653
8  A  7  9  20  2654 # name=A.A2654
9  G  8 10  19  2655 # name=A.G2655
10 U  9 11  18  2656 # name=A.U2656
11 A 10 12  17  2657 # name=A.A2657
12 C 11 13  16  2658 # name=A.C2656, pairedNt=A.G2658
13 G 12 14  15  2659 # name=A.G2659
14 U 13 15  14  2660 # name=A.U2660
15 A 14 16  13  2661 # name=A.A2661
16 A 15 17  12  2662 # name=A.A2662
17 G 16 18  11  2663 # name=A.G2663, pairedNt=A.C2663
18 G 17 19  10  2664 # name=A.G2664
19 U 18 20  09  2665 # name=A.U2665
20 A 19 21  08  2666 # name=A.A2666
21 C 20 22  07  2667 # name=A.C2667
22 G 21 23  06  2668 # name=A.G2668, pairedNt=A.C2668
23 C 22 24  05  2669 # name=A.C2669, pairedNt=A.G2669
24 G 23 25  04  2670 # name=A.G2670, pairedNt=A.U2650
25 A 24 26  03  2671 # name=A.A2671, pairedNt=A.C2649
26 U 25 27  02  2672 # name=A.U2672, pairedNt=A.G2648
27 G 26  0  01  2673 # name=A.G2673
```
3.2.12 Structural features per nucleotide

This section summarizes structural features each nucleotide possesses or is part of. For each identified nucleotide, the output includes one-letter shorthand name, dbn, residue identifier, the rmsd of the base ring atoms, and a comma-separated list of features: anti or syn (χ) orientation, C2’ or C3’-endo sugar puckering, BI or BII backbone conformation, modified or not, if involved in canonical or non-canonical pair, multiplet, helix, stem, or various loops etc.

<table>
<thead>
<tr>
<th>Serial number</th>
<th>Shorthand name</th>
<th>RMSD</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 U</td>
<td>A. U2647</td>
<td>0.011</td>
<td>anti, C3'-endo, non-canonical, non-pair-contact, helix-end, ss-non-loop</td>
</tr>
<tr>
<td>2 G (</td>
<td>A. G2648</td>
<td>0.012</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem-end</td>
</tr>
<tr>
<td>3 C (</td>
<td>A. C2649</td>
<td>0.019</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem</td>
</tr>
<tr>
<td>4 U (</td>
<td>A. U2650</td>
<td>0.019</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem</td>
</tr>
<tr>
<td>5 C (</td>
<td>A. C2651</td>
<td>0.024</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem</td>
</tr>
<tr>
<td>6 C (</td>
<td>A. C2652</td>
<td>0.032</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem-end, internal-loop</td>
</tr>
<tr>
<td>7 U (</td>
<td>A. U2653</td>
<td>0.019</td>
<td>anti, C3'-endo, non-canonical, non-pair-contact, helix, internal-loop, phosphate</td>
</tr>
<tr>
<td>8 A (</td>
<td>A. A2654</td>
<td>0.019</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem-end, internal-loop</td>
</tr>
<tr>
<td>9 G (</td>
<td>A. G2655</td>
<td>0.022</td>
<td>turn, anti, C2'-endo, non-canonical, non-pair-contact, multiplet, internal-loop</td>
</tr>
<tr>
<td>10 U (</td>
<td>A. U2656</td>
<td>0.020</td>
<td>anti, C3'-endo, BI, non-canonical, non-pair-contact, helix, multiplet, internal-loop, phosphate</td>
</tr>
<tr>
<td>11 A (</td>
<td>A. A2657</td>
<td>0.023</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, internal-loop</td>
</tr>
<tr>
<td>12 C (</td>
<td>A. C2658</td>
<td>0.013</td>
<td>anti, C3'-endo, BI, isolated-canonical, non-pair-contact, helix, hairpin-loop, internal-loop</td>
</tr>
<tr>
<td>13 G (</td>
<td>A. G2659</td>
<td>0.033</td>
<td>u-turn, anti, C3'-endo, BI, non-canonical, non-pair-contact, helix-end, hairpin-loop, cap-acceptor, splayed-apart</td>
</tr>
<tr>
<td>14 U (</td>
<td>A. U2660</td>
<td>0.020</td>
<td>turn, u-turn, anti, C3'-endo, non-pair-contact, hairpin-loop, splayed-apart</td>
</tr>
<tr>
<td>15 A (</td>
<td>A. A2661</td>
<td>0.015</td>
<td>u-turn, anti, C3'-endo, BI, non-pair-contact, hairpin-loop, cap-donor, phosphate</td>
</tr>
<tr>
<td>16 A (</td>
<td>A. A2662</td>
<td>0.010</td>
<td>u-turn, anti, C3'-endo, BI, non-canonical, non-pair-contact, helix-end, hairpin-loop, internal-loop, phosphate</td>
</tr>
<tr>
<td>17 G (</td>
<td>A. G2663</td>
<td>0.019</td>
<td>anti, C3'-endo, BI, isolated-canonical, non-pair-contact, helix, hairpin-loop, internal-loop, cap-acceptor</td>
</tr>
<tr>
<td>18 G (</td>
<td>A. G2664</td>
<td>0.014</td>
<td>anti, C3'-endo, BI, non-canonical, non-pair-contact, helix, internal-loop, cap-donor</td>
</tr>
<tr>
<td>19 A (</td>
<td>A. A2665</td>
<td>0.014</td>
<td>anti, C3'-endo, BI, non-canonical, non-pair-contact, helix, multiplet, internal-loop, phosphate</td>
</tr>
<tr>
<td>20 C (</td>
<td>A. C2666</td>
<td>0.016</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, internal-loop, phosphate</td>
</tr>
<tr>
<td>21 C (</td>
<td>A. C2667</td>
<td>0.029</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, internal-loop</td>
</tr>
<tr>
<td>22 G (</td>
<td>A. G2668</td>
<td>0.012</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem-end, internal-loop</td>
</tr>
<tr>
<td>23 G (</td>
<td>A. G2669</td>
<td>0.020</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem</td>
</tr>
<tr>
<td>24 A (</td>
<td>A. A2670</td>
<td>0.019</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem</td>
</tr>
<tr>
<td>25 G (</td>
<td>A. G2671</td>
<td>0.023</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem</td>
</tr>
<tr>
<td>26 U (</td>
<td>A. U2672</td>
<td>0.024</td>
<td>anti, C3'-endo, BI, canonical, non-pair-contact, helix, stem-end</td>
</tr>
<tr>
<td>27 G (</td>
<td>A. G2673</td>
<td>0.010</td>
<td>anti, C3'-endo, non-canonical, non-pair-contact, helix-end, ss-non-loop</td>
</tr>
</tbody>
</table>
3.2.13 Backbone torsion angles and suite names

The auto-generated output file named ‘dssr-torsions.txt’ contains many commonly used backbone parameters, including torsion angles, sugar puckers, and suite names [6]. Given below is a summary of the sections contained in the file so that users can have a better feel of what it has to offer.

Main chain conformational parameters  Here the single-stranded Zp (ssZp) parameter is an extension to the original 3DNA Zp for distinguishing different types of duplexes [17]. Its addition to 3DNA/DSSR has been inspired by the work of Richardson et al. [18], who observed a correlation between the sugar pucker and the perpendicular distance from the 3'-phosphate to the glycosidic bond vector: > 2.9 Å for C3'-endo and < 2.9 Å for C2'-endo sugars. See my blogposts “Single- and double-stranded Zp” and “Sugar pucker correlates with phosphorus-base distance” for more information.

Note also the classifications of the backbone into BI/BII forms, χ into ant/syn conformations, sugar into C2'-endo/ C3'-endo puckers. The collection of relevant information should prove convenient for quick identification of key backbone features of potential interest. To fit the width of the page, the content of the parameters is set in tinier font, and listed separately from the header.

1. alpha: O3'(i-1)-P-O5'-C5'
2. beta: P-O5'-C5'-C4'
3. gamma: O5'-C5'-C4'-C3'
4. delta: C5'-C4'-C3'-O3'
5. epsilon: C4'-C3'-O3'-P(i+1)
6. zeta: C3'-O3'-P(i+1)-O5'(i+1)
7. e-z: epsilon-zeta (BI/BII backbone classification)
8. chi for pyrimidines(Y): O4'-C1'-N1-C2; purines(R): O4'-C1'-N9-C4
9. Range [170, -50(310)] is assigned to anti, and [50, 90] to syn
10. phase-angle: the phase angle of pseudorotation and puckering
11. 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)
14. 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
15. POSITIVE when P lies on the +z-axis side (base in anti conformation);
16. NEGATIVE if P is on the -z-axis side (base in syn conformation)
19. splay: angle between the bridging P to the two base-origins of a dinucleotide.
<table>
<thead>
<tr>
<th>nt</th>
<th>alpha</th>
<th>beta</th>
<th>gamma</th>
<th>delta</th>
<th>epsilon</th>
<th>zeta</th>
<th>e-z</th>
<th>chi</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U</td>
<td>U</td>
<td>9.9(C3'-endo)</td>
<td>4.88</td>
<td>4.90</td>
<td>33.43</td>
<td>179.6(anti)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>G</td>
<td>G</td>
<td>5.3(C3'-endo)</td>
<td>4.47</td>
<td>4.56</td>
<td>22.83</td>
<td>167.8(anti)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>C</td>
<td>12.9(C3'-endo)</td>
<td>4.41</td>
<td>4.66</td>
<td>23.50</td>
<td>160.5(anti)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>U</td>
<td>U</td>
<td>21.3(C3'-endo)</td>
<td>4.30</td>
<td>4.55</td>
<td>21.63</td>
<td>167.2(anti)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>C</td>
<td>12.9(C3'-endo)</td>
<td>4.29</td>
<td>4.58</td>
<td>24.06</td>
<td>159.0(anti)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>A</td>
<td>5.3(C3'-endo)</td>
<td>4.28</td>
<td>4.53</td>
<td>22.84</td>
<td>158.8(anti)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>U</td>
<td>U</td>
<td>15.7(C3'-endo)</td>
<td>4.00</td>
<td>4.62</td>
<td>22.57</td>
<td>145.3(anti)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>A</td>
<td>151.0(C2'-endo)</td>
<td>0.91</td>
<td>0.92</td>
<td>43.08</td>
<td>145.3(anti)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>G</td>
<td>G</td>
<td>0.4(C3'-endo)</td>
<td>4.35</td>
<td>4.42</td>
<td>27.39</td>
<td>173.9(anti)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>U</td>
<td>U</td>
<td>29.4(C3'-endo)</td>
<td>4.76</td>
<td>4.95</td>
<td>95.33</td>
<td>172.9(anti)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>A</td>
<td>A</td>
<td>32.3(C3'-endo)</td>
<td>3.51</td>
<td>3.96</td>
<td>27.58</td>
<td>157.3(anti)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>C</td>
<td>8.0(C3'-endo)</td>
<td>4.39</td>
<td>4.70</td>
<td>26.73</td>
<td>165.0(anti)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>A</td>
<td>A</td>
<td>29.4(C3'-endo)</td>
<td>4.76</td>
<td>4.95</td>
<td>95.33</td>
<td>172.9(anti)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>U</td>
<td>U</td>
<td>32.3(C3'-endo)</td>
<td>3.51</td>
<td>3.96</td>
<td>27.58</td>
<td>157.3(anti)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>A</td>
<td>A</td>
<td>8.0(C3'-endo)</td>
<td>4.39</td>
<td>4.70</td>
<td>26.73</td>
<td>165.0(anti)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>A</td>
<td>A</td>
<td>18.7(C3'-endo)</td>
<td>2.71</td>
<td>4.68</td>
<td>44.81</td>
<td>170.9(anti)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>G</td>
<td>G</td>
<td>27.7(C3'-endo)</td>
<td>4.71</td>
<td>4.82</td>
<td>38.71</td>
<td>170.9(anti)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>G</td>
<td>G</td>
<td>9.7(C3'-endo)</td>
<td>4.48</td>
<td>4.63</td>
<td>74.59</td>
<td>162.8(anti)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>A</td>
<td>A</td>
<td>4.5(C3'-endo)</td>
<td>4.74</td>
<td>4.75</td>
<td>25.77</td>
<td>179.1(anti)</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>A</td>
<td>A</td>
<td>165.7(C3'-endo)</td>
<td>4.57</td>
<td>4.74</td>
<td>32.11</td>
<td>165.3(anti)</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>C</td>
<td>C</td>
<td>16.5(C3'-endo)</td>
<td>4.53</td>
<td>4.79</td>
<td>37.34</td>
<td>165.7(anti)</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>G</td>
<td>G</td>
<td>7.3(C3'-endo)</td>
<td>4.59</td>
<td>4.73</td>
<td>24.02</td>
<td>166.5(anti)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>G</td>
<td>G</td>
<td>13.0(C3'-endo)</td>
<td>4.59</td>
<td>4.65</td>
<td>24.40</td>
<td>166.5(anti)</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>A</td>
<td>A</td>
<td>66.2(C3'-endo)</td>
<td>4.44</td>
<td>4.57</td>
<td>22.81</td>
<td>166.5(anti)</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>G</td>
<td>G</td>
<td>12.6(C3'-endo)</td>
<td>4.44</td>
<td>4.57</td>
<td>22.81</td>
<td>166.5(anti)</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>U</td>
<td>U</td>
<td>9.2(C3'-endo)</td>
<td>4.47</td>
<td>4.62</td>
<td>22.80</td>
<td>163.7(anti)</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>G</td>
<td>G</td>
<td>19.2(C3'-endo)</td>
<td>4.00</td>
<td>4.37</td>
<td>27.53</td>
<td>152.8(anti)</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>G</td>
<td>G</td>
<td>357.2(C2'-exo)</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>141.9(anti)</td>
<td></td>
</tr>
</tbody>
</table>
Virtual torsion angles Three sets of virtual torsion angles are calculated: one is the most commonly used $\eta/\theta$ pair pioneered by Olson [19], the second is its $\eta'/\theta'$ variant (using the C1' atom instead of C4' recently introduced by Pyle et al. [20], and the third (termed $\eta''/\theta''$) takes advantage of the origin of the base reference frame [2] in place of the C4' or C1' atom. This set of base-phosphorus virtual torsions is unique to 3DNA/DSSR: it was first introduced in 3DNA v2.1 as an option to the analyze program (i.e., analyze -torsion) in early 2012, and became available in DSSR as of v1.5.3-2016apr11. See my blogpost titled “Pseudo-torsions to simplify the representation of DNA/RNA backbone conformation” (dated 2012-04-22) for details.

\[
\begin{align*}
\text{eta:} & \quad C4'(i-1)-P(i)-C4'(i)-P(i+1) \\
\text{theta:} & \quad P(i)-C4'(i)-P(i+1)-C4'(i+1) \\
\end{align*}
\]

\[
\begin{align*}
\text{eta':} & \quad C1'(i-1)-P(i)-C1'(i)-P(i+1) \\
\text{theta':} & \quad P(i)-C1'(i)-P(i+1)-C1'(i+1) \\
\text{Ref: Keating et al. (2011): “A new way to see RNA.” Quarterly Reviews of Biophysics, 44(4):433-466]
\end{align*}
\]

\[
\begin{align*}
\text{eta'':} & \quad \text{base}(i-1)-P(i)-\text{base}(i)-P(i+1) \\
\text{theta'':} & \quad P(i)-\text{base}(i)-P(i+1)-\text{base}(i+1) \\
\end{align*}
\]

<table>
<thead>
<tr>
<th>nt</th>
<th>eta</th>
<th>theta</th>
<th>eta'</th>
<th>theta'</th>
<th>eta''</th>
<th>theta''</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U</td>
<td>A. U</td>
<td>2647</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>G</td>
<td>A. G</td>
<td>2648</td>
<td>172.1</td>
<td>-133.3</td>
<td>-163.5</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>A. C</td>
<td>2649</td>
<td>162.7</td>
<td>-140.1</td>
<td>-178.0</td>
</tr>
<tr>
<td>4</td>
<td>U</td>
<td>A. U</td>
<td>2650</td>
<td>167.0</td>
<td>-148.3</td>
<td>-174.5</td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>A. C</td>
<td>2651</td>
<td>165.4</td>
<td>-147.2</td>
<td>177.6</td>
</tr>
<tr>
<td>6</td>
<td>C</td>
<td>A. C</td>
<td>2652</td>
<td>171.3</td>
<td>-142.0</td>
<td>-176.3</td>
</tr>
<tr>
<td>7</td>
<td>U</td>
<td>A. U</td>
<td>2653</td>
<td>172.2</td>
<td>-18.3</td>
<td>-170.0</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>A. A</td>
<td>2654</td>
<td>46.3</td>
<td>172.0</td>
<td>120.5</td>
</tr>
<tr>
<td>9</td>
<td>G</td>
<td>A. G</td>
<td>2655</td>
<td>-44.2</td>
<td>24.9</td>
<td>-82.3</td>
</tr>
<tr>
<td>10</td>
<td>U</td>
<td>A. U</td>
<td>2656</td>
<td>170.9</td>
<td>-121.7</td>
<td>163.1</td>
</tr>
<tr>
<td>11</td>
<td>A</td>
<td>A. A</td>
<td>2657</td>
<td>162.1</td>
<td>-127.4</td>
<td>-177.7</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>A. C</td>
<td>2658</td>
<td>159.4</td>
<td>-135.3</td>
<td>-176.0</td>
</tr>
<tr>
<td>13</td>
<td>G</td>
<td>A. G</td>
<td>2659</td>
<td>167.6</td>
<td>-117.7</td>
<td>-179.4</td>
</tr>
<tr>
<td>14</td>
<td>U</td>
<td>A. U</td>
<td>2660</td>
<td>15.1</td>
<td>-126.1</td>
<td>43.6</td>
</tr>
<tr>
<td>15</td>
<td>C</td>
<td>A. C</td>
<td>2661</td>
<td>160.4</td>
<td>-132.0</td>
<td>-169.4</td>
</tr>
<tr>
<td>16</td>
<td>A</td>
<td>A. A</td>
<td>2662</td>
<td>167.0</td>
<td>-83.0</td>
<td>-174.8</td>
</tr>
<tr>
<td>17</td>
<td>G</td>
<td>A. G</td>
<td>2663</td>
<td>172.6</td>
<td>-154.0</td>
<td>-148.2</td>
</tr>
<tr>
<td>18</td>
<td>G</td>
<td>A. G</td>
<td>2664</td>
<td>166.2</td>
<td>166.9</td>
<td>-168.9</td>
</tr>
<tr>
<td>19</td>
<td>A</td>
<td>A. A</td>
<td>2665</td>
<td>-155.6</td>
<td>141.6</td>
<td>175.0</td>
</tr>
<tr>
<td>20</td>
<td>C</td>
<td>A. C</td>
<td>2666</td>
<td>-178.4</td>
<td>-125.3</td>
<td>-169.0</td>
</tr>
<tr>
<td>21</td>
<td>C</td>
<td>A. C</td>
<td>2667</td>
<td>164.6</td>
<td>-120.7</td>
<td>-172.9</td>
</tr>
<tr>
<td>22</td>
<td>G</td>
<td>A. G</td>
<td>2668</td>
<td>164.9</td>
<td>-150.0</td>
<td>-168.4</td>
</tr>
<tr>
<td>23</td>
<td>G</td>
<td>A. G</td>
<td>2669</td>
<td>171.3</td>
<td>-139.8</td>
<td>-172.8</td>
</tr>
<tr>
<td>24</td>
<td>A</td>
<td>A. A</td>
<td>2670</td>
<td>170.6</td>
<td>-153.2</td>
<td>-173.8</td>
</tr>
<tr>
<td>25</td>
<td>G</td>
<td>A. G</td>
<td>2671</td>
<td>170.4</td>
<td>-134.4</td>
<td>-180.0</td>
</tr>
<tr>
<td>26</td>
<td>U</td>
<td>A. U</td>
<td>2672</td>
<td>172.2</td>
<td>-167.9</td>
<td>-166.6</td>
</tr>
<tr>
<td>27</td>
<td>G</td>
<td>A. G</td>
<td>2673</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DSSR-Jmol · DSSR-PyMOL · DSSR Web Interface
**Sugar conformational parameters**  By default, the sugar pucker analysis follows the work of Altona and Sundaralingam [21]. The phase angle (in the range of 0° to 360°) of pseudorotation is divided equally into ten 36° regions; the two most frequent sugar pucker modes 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, each sugar pucker is assigned into either ∼C3′-endo or ∼C2′-endo (see above in Section 3.2.13 under ‘sugar-type’) by its match against corresponding fiber models. More details on this topic are available from my blogpost titled “Conformation of the sugar ring in nucleic acid structures”.

---

<table>
<thead>
<tr>
<th>nt</th>
<th>v0</th>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>tm</th>
<th>P</th>
<th>Puckering</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U A. U2647</td>
<td>7.5</td>
<td>-34.5</td>
<td>46.2</td>
<td>-43.8</td>
<td>23.1</td>
<td>46.9</td>
</tr>
<tr>
<td>2</td>
<td>G A. G2648</td>
<td>9.5</td>
<td>-31.5</td>
<td>39.4</td>
<td>-35.5</td>
<td>16.8</td>
<td>39.6</td>
</tr>
<tr>
<td>3</td>
<td>C A. C2649</td>
<td>4.0</td>
<td>-28.3</td>
<td>39.9</td>
<td>-38.4</td>
<td>21.9</td>
<td>40.9</td>
</tr>
<tr>
<td>4</td>
<td>U A. U2650</td>
<td>-2.4</td>
<td>-25.5</td>
<td>41.9</td>
<td>-44.0</td>
<td>29.4</td>
<td>44.9</td>
</tr>
<tr>
<td>5</td>
<td>C A. C2651</td>
<td>4.8</td>
<td>-32.5</td>
<td>45.7</td>
<td>-44.6</td>
<td>25.0</td>
<td>46.9</td>
</tr>
<tr>
<td>6</td>
<td>C A. C2652</td>
<td>2.9</td>
<td>-29.7</td>
<td>43.8</td>
<td>-44.0</td>
<td>25.9</td>
<td>45.4</td>
</tr>
<tr>
<td>7</td>
<td>U A. U2653</td>
<td>1.6</td>
<td>-28.2</td>
<td>42.3</td>
<td>-41.3</td>
<td>25.2</td>
<td>44.0</td>
</tr>
<tr>
<td>8</td>
<td>A A. A2654</td>
<td>-33.2</td>
<td>44.3</td>
<td>-38.7</td>
<td>20.1</td>
<td>8.5</td>
<td>44.3</td>
</tr>
<tr>
<td>9</td>
<td>G A. G2655</td>
<td>-37.3</td>
<td>50.1</td>
<td>-43.9</td>
<td>22.9</td>
<td>8.9</td>
<td>50.0</td>
</tr>
<tr>
<td>10</td>
<td>U A. U2656</td>
<td>12.7</td>
<td>-32.9</td>
<td>39.6</td>
<td>-33.2</td>
<td>13.3</td>
<td>39.6</td>
</tr>
<tr>
<td>11</td>
<td>A A. A2657</td>
<td>-6.4</td>
<td>-21.7</td>
<td>39.9</td>
<td>-44.5</td>
<td>32.0</td>
<td>44.6</td>
</tr>
<tr>
<td>12</td>
<td>C A. C2658</td>
<td>0.0</td>
<td>-28.5</td>
<td>44.6</td>
<td>-44.4</td>
<td>28.4</td>
<td>46.9</td>
</tr>
<tr>
<td>13</td>
<td>G A. G2659</td>
<td>-9.4</td>
<td>-20.6</td>
<td>40.1</td>
<td>-45.3</td>
<td>35.5</td>
<td>46.1</td>
</tr>
<tr>
<td>14</td>
<td>U A. U2660</td>
<td>-8.9</td>
<td>-14.6</td>
<td>31.3</td>
<td>-37.4</td>
<td>29.0</td>
<td>37.0</td>
</tr>
<tr>
<td>15</td>
<td>A A. A2661</td>
<td>7.2</td>
<td>-28.7</td>
<td>38.3</td>
<td>35.0</td>
<td>17.5</td>
<td>38.6</td>
</tr>
<tr>
<td>16</td>
<td>A A. A2662</td>
<td>0.0</td>
<td>-21.1</td>
<td>32.9</td>
<td>-33.9</td>
<td>21.4</td>
<td>34.8</td>
</tr>
<tr>
<td>17</td>
<td>G A. G2663</td>
<td>-7.9</td>
<td>-22.7</td>
<td>42.3</td>
<td>-47.3</td>
<td>35.7</td>
<td>47.8</td>
</tr>
<tr>
<td>18</td>
<td>G A. G2664</td>
<td>7.4</td>
<td>-33.6</td>
<td>45.9</td>
<td>-43.1</td>
<td>22.1</td>
<td>46.5</td>
</tr>
<tr>
<td>19</td>
<td>A A. A2665</td>
<td>11.2</td>
<td>-35.6</td>
<td>45.2</td>
<td>-39.6</td>
<td>18.1</td>
<td>45.4</td>
</tr>
<tr>
<td>20</td>
<td>C A. C2666</td>
<td>2.0</td>
<td>-27.6</td>
<td>41.2</td>
<td>-41.2</td>
<td>24.6</td>
<td>42.8</td>
</tr>
<tr>
<td>21</td>
<td>C A. C2667</td>
<td>1.9</td>
<td>-30.7</td>
<td>45.7</td>
<td>-46.0</td>
<td>28.2</td>
<td>47.6</td>
</tr>
<tr>
<td>22</td>
<td>G A. G2668</td>
<td>9.1</td>
<td>-34.4</td>
<td>44.4</td>
<td>-40.9</td>
<td>20.0</td>
<td>44.7</td>
</tr>
<tr>
<td>23</td>
<td>G A. G2669</td>
<td>4.5</td>
<td>-31.3</td>
<td>45.1</td>
<td>-43.5</td>
<td>24.4</td>
<td>46.3</td>
</tr>
<tr>
<td>24</td>
<td>A A. A2670</td>
<td>5.3</td>
<td>-32.9</td>
<td>46.7</td>
<td>-45.3</td>
<td>25.1</td>
<td>47.9</td>
</tr>
<tr>
<td>25</td>
<td>G A. G2671</td>
<td>7.6</td>
<td>-33.2</td>
<td>44.6</td>
<td>-41.4</td>
<td>21.7</td>
<td>45.2</td>
</tr>
<tr>
<td>26</td>
<td>U A. U2672</td>
<td>-0.3</td>
<td>-23.0</td>
<td>35.9</td>
<td>-37.6</td>
<td>23.6</td>
<td>38.0</td>
</tr>
<tr>
<td>27</td>
<td>G A. G2673</td>
<td>17.1</td>
<td>-39.6</td>
<td>45.4</td>
<td>-37.1</td>
<td>12.9</td>
<td>45.4</td>
</tr>
</tbody>
</table>
Assignment of sugar-phosphate backbone suite names  According to Richardson et al. [6], the backbone suite is defined as a sugar-to-sugar version of a nucleotide (in contrast to the traditional definition as a phosphate-to-phosphate unit). A total of 53 backbone conformer bins have been defined and expressed in mnemonic 2-letter names (e.g., ‘1a’, ‘5z’, with outliers signified by ‘!!’). For all the cases tested to date, the DSSR implementation gives identical results to those determined by the Suitename program from the Richardson laboratory.

<table>
<thead>
<tr>
<th>nt bin cluster suiteness</th>
</tr>
</thead>
<tbody>
<tr>
<td>U A. U2647 inc __ 0</td>
</tr>
<tr>
<td>G A. G2648 33p 1a 0.052</td>
</tr>
<tr>
<td>C A. C2649 33p 1a 0.666</td>
</tr>
<tr>
<td>U A. U2650 33p 1a 0.875</td>
</tr>
<tr>
<td>C A. C2651 33p 1a 0.871</td>
</tr>
<tr>
<td>C A. C2652 33p 1a 0.919</td>
</tr>
<tr>
<td>U A. U2653 33p 1a 0.929</td>
</tr>
<tr>
<td>A A. A2654 32p 5z 0.849</td>
</tr>
<tr>
<td>G A. G2655 23t 4s 0.730</td>
</tr>
<tr>
<td>U A. U2656 23p #a 0.842</td>
</tr>
<tr>
<td>A A. A2657 33p 1a 0.693</td>
</tr>
<tr>
<td>C A. C2658 33p 1a 0.884</td>
</tr>
<tr>
<td>G A. G2659 33p 1a 0.894</td>
</tr>
<tr>
<td>U A. U2660 33p 1g 0.736</td>
</tr>
<tr>
<td>A A. A2661 33p 1L 0.688</td>
</tr>
<tr>
<td>A A. A2662 33p 1a 0.692</td>
</tr>
<tr>
<td>G A. G2663 33t 1c 0.321</td>
</tr>
<tr>
<td>G A. G2664 33p 1a 0.878</td>
</tr>
<tr>
<td>A A. A2665 33t 1e 0.875</td>
</tr>
<tr>
<td>C A. C2666 33p 1a 0.891</td>
</tr>
<tr>
<td>C A. C2667 33p 1a 0.887</td>
</tr>
<tr>
<td>G A. G2668 33p 1a 0.756</td>
</tr>
<tr>
<td>G A. G2669 33p 1a 0.625</td>
</tr>
<tr>
<td>A A. A2670 33p 1a 0.914</td>
</tr>
<tr>
<td>G A. G2671 33p 1a 0.878</td>
</tr>
<tr>
<td>U A. U2672 33p 1a 0.912</td>
</tr>
<tr>
<td>G A. G2673 trig !! 0</td>
</tr>
</tbody>
</table>

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).
Note that in assigning suite names, the $\chi$ 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 Dickerson dodecamer (355d), 16 out of 22 suites ($\sim$73%) are classified as outliers (‘!!’).

### 3.3 Default run on PDB entry 1ehz (tRNA$^{\text{Phe}}$) – summary notes

The cloverleaf secondary structure of tRNA has become an iconic image in structural biology. The four stems (the acceptor stem, the D stem, the anti-codon stem, and the T stem) form two halves of the L-shaped tertiary structure through coaxial stacking of the stems. Yet, other than DSSR, there appear to be no alternative software programs that can neatly delineate the L-shaped 3D vs. the cloverleaf 2D structures of a tRNA molecule from PDB coordinates. The cover image of the manual illustrates DSSR’s capability to solve this basic problem, using 1ehz, the crystal structure of yeast phenylalanine tRNA, as an example.

#### 3.3.1 Brief summary

The screen output of a DSSR run on 1ehz (assuming the 3D coordinates file is called ‘1ehz.pdb’) is shown below. Note that DSSR correctly identifies two helices, four stems, three hairpin loops, and one four-way junction loop, among other things.

```
1 total number of nucleotides: 76
2 modified nucleotides: 14
3 total number of base pairs: 34
4 total number of multiplets: 4
5 total number of helices: 2
6 total number of stems: 4
7 total number of isolated WC/wobble pairs: 1
8 total number of atom-base capping interactions: 4
9 total number of splayed-apart dinucleotides: 9
10 consolidated into units: 6
11 total number of hairpin loops: 3
12 total number of junctions: 1
13 total number of non-loop single-stranded segments: 1
14 total number of kissing loops: 1
15
16 Time used: 00:00:00:00
```
3.3.2 Modified nucleotides

The 27-nt long RNA fragment 1msy detailed in Section 3.2 contains only canonical nts (A, C, G, and U). However, 14 out of the 76 nts in tRNA\textsuperscript{Phe} 1ehz are modified (of 11 different types), as listed below.

<table>
<thead>
<tr>
<th>nt</th>
<th>count</th>
<th>list</th>
</tr>
</thead>
<tbody>
<tr>
<td>1MA-a</td>
<td>1</td>
<td>A.1MA58</td>
</tr>
<tr>
<td>2MG-g</td>
<td>1</td>
<td>A.2MG10</td>
</tr>
<tr>
<td>3MC-c</td>
<td>2</td>
<td>A.3MC40,A.3MC49</td>
</tr>
<tr>
<td>4MU-t</td>
<td>1</td>
<td>A.4MU54</td>
</tr>
<tr>
<td>5MG-g</td>
<td>1</td>
<td>A.5MG46</td>
</tr>
<tr>
<td>6H2U-u</td>
<td>2</td>
<td>A.6H2U16,A.6H2U17</td>
</tr>
<tr>
<td>7M2G-g</td>
<td>1</td>
<td>A.7M2G26</td>
</tr>
<tr>
<td>8OMC-c</td>
<td>1</td>
<td>A.8OMC32</td>
</tr>
<tr>
<td>9OMG-g</td>
<td>1</td>
<td>A.9OMG34</td>
</tr>
<tr>
<td>10PSU-P</td>
<td>2</td>
<td>A.10PSU39,A.10PSU55</td>
</tr>
<tr>
<td>11YYG-g</td>
<td>1</td>
<td>A.11YYG37</td>
</tr>
</tbody>
</table>

Lines 3-13 in the above listing provide details about each of the 11 types of modified nts: a 3-letter residue name followed by its 1-letter shorthand form (under column nt), its frequency (under column count), and a comma separated enumeration of the occurrences (under column list). For example, line 5 means 5MC is found twice in 1ehz, with residue numbers 40 and 49, respectively.

From its 3-letter residue name, further information about a modified nt can be obtained via 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 as listed below. Use Jmol/PyMOL to visualize the output file ‘dssr-multiplets.pdb’ for verification.

<table>
<thead>
<tr>
<th>nts</th>
<th>list</th>
</tr>
</thead>
<tbody>
<tr>
<td>3UAA</td>
<td>A.U8,A.A14,A.A21</td>
</tr>
<tr>
<td>3AUU</td>
<td>A.A9,A.U12,A.A23</td>
</tr>
<tr>
<td>3gCG</td>
<td>A.2MG10,A.C25,A.G45</td>
</tr>
<tr>
<td>3CGg</td>
<td>A.C13,A.G22,A.7MG46</td>
</tr>
</tbody>
</table>

3.3.4 Relationship between helices and stems

The connection between the two helices and the four stems is available from the main output file, with related portions excerpted below. The meaning of each section should
be easy to follow, especially in connection with the tRNA\textsuperscript{Phe} (1ehz) secondary structure image shown on the cover of this manual.

<table>
<thead>
<tr>
<th>Helix</th>
<th>bps</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>15</td>
</tr>
<tr>
<td>#2</td>
<td>15</td>
</tr>
</tbody>
</table>

3.3.5 Three hairpin loops

As expected, three hairpin loops are identified, with details listed below.

<table>
<thead>
<tr>
<th>Loop</th>
<th>nts</th>
<th>Linked by</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>[8]</td>
</tr>
<tr>
<td></td>
<td>summary:</td>
<td>[1] 8 [A.13 A.22] 4</td>
</tr>
<tr>
<td></td>
<td>nt(s):</td>
<td>10 CAuuGGGAG A.C13 ,A.A14 ,A.G15 ,A.H2U16 ,A.H2U17 ,A.G18 ,A.G19 ,A.G20 ,A.A21 ,A.G22</td>
</tr>
<tr>
<td></td>
<td>nt(s):</td>
<td>8 AGuuGGGA A.A14 ,A.G15 ,A.H2U16 ,A.H2U17 ,A.G18 ,A.G19 ,A.G20 ,A.A21</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>[9]</td>
</tr>
<tr>
<td></td>
<td>summary:</td>
<td>[1] 9 [A.30 A.40] 4</td>
</tr>
<tr>
<td></td>
<td>nt(s):</td>
<td>11 GAcUgAaApC A.G30 ,A.A31 ,A.OMC32 ,A.U33 ,A.OMG34 ,A.A35 ,A.A36 ,A.YYG37 ,A.A38 ,A. $</td>
</tr>
</tbody>
</table><p>ightarrow$ PSU39 ,A.5MC40 |
|      | nt(s):  | 9 AcUgAaApA A.A31 ,A.OMC32 ,A.U33 ,A.OMG34 ,A.A35 ,A.A36 ,A.YYG37 ,A.A38 ,A.ASU39 |
| 3    | 9   | [7]       |
|      | nt(s):  | 7 tPCGauUC A.5SU54 ,A.PSU55 ,A.C56 ,A.G57 ,A.1MA58 ,A.U59 ,A.C60 |</p>

3.3.6 One four-way junction loop

The four-way junction loop is delineated by the four stems ([#1,#2,#3,#4]), with [2,1,5,0] nucleotides connecting each pair of them, respectively. This junction loop is well-documented: see the cover image on the right, and also Fig. 2 of the [1]DSSR paper.

Note that it contains four modified nucleotides (2MG10, M2G26, 7MG46, 5MC49), which may pose a problem for some RNA structural analysis programs.

<table>
<thead>
<tr>
<th>Junction</th>
<th>nts</th>
<th>Linked by</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>[2,1,5,0]</td>
</tr>
<tr>
<td></td>
<td>nt(s):</td>
<td>16 UUAgCGgCGgUCCgA A.U7 ,A.U8 ,A.A9 ,A.2MG10 ,A.C25 ,A.H2U26 ,A.C27 ,A.G43 ,A.A44 ,A.O46 ,$</td>
</tr>
</tbody>
</table><p>ightarrow$ A.7MG46 ,A.U47 ,A.C48 ,A.5MC49 ,A.G58 ,A.A66 |
|           | nt(s):  | 2 UA A.U8 ,A.A9 |</p>
3.3.7 Splayed-apart conformations

As of v1.6.3-2016oct19, DSSR has added a new section of splayed-apart conformations where the two bases in a dinucleotide are far off from one another. Three simple geometric parameters are calculated, based on the position of the intermediate phosphorus (P) atom and the origins (O1 and O2) of the two base reference frames of a N1-p-N2 dinucleotide.

- **angle** – the O1–P–O2 angle. By default, a dinucleotide with angle $\geq 85^\circ$ is defined as splayed-apart.
- **distance** – the O1–O2 distance in Å.
- **ratio** – the ratio of distance O1–O2 over the sum of distances O1–P and P–O2.

Obviously, the three parameters are highly correlated. The angle criterion is more intuitive, and it has thus been picked up as the criterion for splayed-apart conformation (≥ $85^\circ$ by default).

The output for 1ehz is listed below, where nine such splayed-out cases are identified. Seven of them are located in loop regions, and one in the 3′-end single-stranded terminal.

<table>
<thead>
<tr>
<th>List of 9 splayed-apart dinucleotides</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 A.U7 A.U8</td>
</tr>
<tr>
<td>2 A.H2U16 A.H2U17</td>
</tr>
<tr>
<td>3 A.H2U17 A.G18</td>
</tr>
<tr>
<td>4 A.G19 A.G20</td>
</tr>
<tr>
<td>5 A.7MG46 A.U47</td>
</tr>
<tr>
<td>6 A.U47 A.C48</td>
</tr>
<tr>
<td>7 A.C48 A.5MC49</td>
</tr>
<tr>
<td>8 A.C60 A.C61</td>
</tr>
<tr>
<td>9 A.C75 A.A76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary of 6 splayed-apart units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 nts=2 UU A.U7,A.U8</td>
</tr>
<tr>
<td>2 nts=3 uuG A.H2U16,A.H2U17,A.G18</td>
</tr>
<tr>
<td>3 nts=2 GG A.G19,A.G20</td>
</tr>
<tr>
<td>4 nts=4 gUCc A.7MG46,A.U47,A.C48,A.5MC49</td>
</tr>
<tr>
<td>5 nts=2 CC A.C60,A.C61</td>
</tr>
<tr>
<td>6 nts=2 CA A.C75,A.A76</td>
</tr>
</tbody>
</table>

Consecutive splayed-apart dinucleotides, if any, are merged into splayed-apart units, each with two and more continuous nucleotides. Moreover, the atomic coordinates of the
units are extracted into a NMR-style ensemble file (named ‘dssr-splays.pdb’ by default) for easy visualization. As shown in the Summary section for 1ehz (see above), the eight splayed-apart dinucleotides are consolidated into five units. The largest unit (no. 4 on line #15), with four nucleotides (\(7MG46, U47, C48, 5MC49\)) in the four-way junction loop, is illustrated in Figure 9.

![Figure 9](image_url)

**Figure 9:** Splayed-apart unit with four nucleotides (\(7MG46, U47, C48, 5MC49\)) identified in PDB entry 1ehz. The unit is located in the four-way junction loop. The image was produced with Jmol.

### 3.3.8 Pseudoknot

The tRNA\(^{\text{Phe}}\) 1ehz contains a pseudoknot, due to the formation of the canonical G–C pair between A.G19 and A.C56 (see the long cyan line in the 2D representation on the cover
image). This pair is conserved and located at the elbow of the L-shaped tertiary structure that brings the D-loop and the TΨC-loop together, via essentially a simple kissing loop interaction. The elbow G–C pair has been shown to stack against a base triplet of the T-box riboswitch [22], apparently playing an important role in forming the stem-I-tRNA complex (4lck).

In DSSR dbn output, the bps in first-order pseudoknots are designated by matched square brackets, as shown below for 1ehz.

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

For bps in higher-order pseudoknots, matched curly brackets {}, angle brackets <>, or upper-lower case letters (e.g., Aa, Bb, Cc etc.) are used. Please refer to Section 4 for an example.

### 3.4 Default run on PDB entry 1jj2 – four auto-checked motifs

The crystal structure of the *Halocarcina marismortui* large ribosomal subunit (1jj2) serves as an example how DSSR can analyze complicated RNA structures as easily as smaller ones (such as 1msy and 1ehz discussed in previous sections). Let the PDB 3D coordinate file be called ‘1jj2.pdb’, the screen output is shown below. The running time will obviously depend on hardware configurations; I have seen from 20 to 40 seconds using iMac, Linux, and MacBook Air.

```
total number of nucleotides: 2876
total number of amino acids: 3701
total number of base pairs: 1467
total number of multiplets: 248
total number of helices: 87
total number of stems: 179
total number of isolated WC/wobble pairs: 54
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: 36
total number of internal loops: 67
total number of junctions: 36
total number of non-loop single-stranded segments: 32
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
```
Notice the large numbers of bps, multiplets, loops (of various types), and four additional motifs – kissing loops, A-minor motifs, ribose zippers, and k-turns – that will be discussed in detail in the following sections.

As a side note, DSSR detects an additional base pentaplet (AUAAG, see Figure 10) which was previously missed using 3DNA [11]. The five nts (A306, U325, A331, A340, and G345), all on chain ‘0’, form five base-base H-bonds as well as four additional H-bonding interactions involving O2′ atoms. Overall, in default settings, DSSR is more sophisticated than the 3DNA ‘find_pair’ program in identifying bps, multiplets, and double helical regions.

Figure 10: The additional base pentaplet (AUAAG) identified by DSSR but missed using 3DNA [9]. Here all five nts are derived from chain 0 of 1jj2. The image was produced with PyMOL.
3.4.1 Kissing loops

The kissing-loop motif is characterized by canonical base-pairing interactions (forming a stem) between two hairpin loops. Five such motifs are identified in 1jj2 as listed below, and the fifth one (line 6 on the listing below) is illustrated in Figure 11.

<table>
<thead>
<tr>
<th>List of 5 kissing loop interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 isolated - pair # -43 between hairpin loops #51 and #53</td>
</tr>
<tr>
<td>2 isolated - pair # -6 between hairpin loops #6 and #8</td>
</tr>
<tr>
<td>3 stem # 8 between hairpin loops #1 and #3</td>
</tr>
<tr>
<td>4 stem # 9 between hairpin loops #1 and #3</td>
</tr>
<tr>
<td>5 stem # 29 between hairpin loops #14 and #57</td>
</tr>
</tbody>
</table>

The stem (#29) referred to above is:

<table>
<thead>
<tr>
<th>stem#29[#15] bps=6</th>
</tr>
</thead>
<tbody>
<tr>
<td>strand-1 5'-CAUCGA-3'</td>
</tr>
<tr>
<td>bp-type</td>
</tr>
<tr>
<td>strand-2 3'-GUAGUU-5'</td>
</tr>
<tr>
<td>helix-form AAA</td>
</tr>
<tr>
<td>1 0. C418 0. G4249 C-G WC 19-XIX cWW cW-W</td>
</tr>
<tr>
<td>2 0. A419 0. U4248 A-U WC 20-XX cWW cW-W</td>
</tr>
<tr>
<td>3 0. U420 0. A4247 U-A WC 20-XX cWW cW-W</td>
</tr>
<tr>
<td>4 0. C421 0. G4246 C-G WC 19-XIX cWW cW-W</td>
</tr>
<tr>
<td>5 0. G422 0. U4245 G-U Wobble 28-XXVIII cWW cW-W</td>
</tr>
<tr>
<td>6 0. A423 0. U4244 A-U WC 20-XX cWW cW-W</td>
</tr>
</tbody>
</table>

The two interacting hairpin loops (#14 and #57) are:

<table>
<thead>
<tr>
<th>14 hairpin loop: nts=9; [7]; linked by [#28]</th>
</tr>
</thead>
<tbody>
<tr>
<td>summary: [1] 7 [0.416 0.424] 5</td>
</tr>
<tr>
<td>nts=9 GGCAUCGAC 0.0416,0.0417,0.0418,0.0419,0.0420,0.0421,0.0422,0.0423,0.0424</td>
</tr>
<tr>
<td>nts=7 GCAUCGA 0.0417,0.0418,0.0419,0.0420,0.0421,0.0422,0.0423</td>
</tr>
<tr>
<td>57 hairpin loop: nts=9; [7]; linked by [#145]</td>
</tr>
<tr>
<td>summary: [1] 7 [0.2442 0.2450] 5</td>
</tr>
<tr>
<td>nts=9 GGUUGAUGC 0.2442,0.2443,0.2444,0.2445,0.2446,0.2447,0.2448,0.2449,0.2450</td>
</tr>
<tr>
<td>nts=7 GUGAUGC 0.2443,0.2444,0.2445,0.2446,0.2447,0.2448,0.2449</td>
</tr>
</tbody>
</table>

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 the A-minor motif [14]. 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 further divided into four subtypes. Of these, only two types (I and II) are believed to be adenine-specific, and they are identified...
Figure 11: One kissing-loop motif identified in 1jj2. Here, one hairpin loop is colored yellow (#46, with nts from 416 to 424 on chain ‘0’), and the other red (#52, with nts from 2442 to 2450 on the same chain). Six base-pair interactions between the two loop regions form stem #29. The image was produced with Jmol.

by DSSR (see Figure 12).

A new type, designated ‘X’ (eXtended, for cases other than the classic types I and II), was introduced into DSSR as of v1.6.6-2017feb20. In types I and II A-minor motifs, the adenine has its minor groove edge facing the minor groove of a canonical pair, and the O2' atom of adenine is involved in H-bonding interactions with the pair. In the miscellaneous type X, the adenine uses its Watson-Crick edge or major-groove edge to interact with the minor groove of a canonical pair, without resorting to the O2' atom. This type of A-minor interactions was recently reported in a crystal structure of the self-cleaving Pistol ribozyme [23].

DSSR is unique in using standard base reference frames (Figure 1B) and bp parameters
to characterize A-minor motifs. The program actually has more features on the ‘A’-minor motifs than documented here. When applied to 1jj2, a total of 156 A-minor motifs are identified (stored in file ‘dssr-Aminors.pdb’), including type X cases. Three sample entries, each for types I, II, and X, are listed below: #4 (lines 2-4 in the listing) for type I, #7 (lines 6-8) for type II (see also Figure 12), and #11 (lines 10-12) for type X.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Type</th>
<th>A-A</th>
<th>G-C</th>
<th>A</th>
<th>G</th>
<th>C</th>
<th>WC</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>I</td>
<td>A</td>
<td>G-C</td>
<td>A</td>
<td>69</td>
<td>0.65</td>
<td>WC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>II</td>
<td>A</td>
<td>G-C</td>
<td>A</td>
<td>152</td>
<td>0.41</td>
<td>WC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>A</td>
<td>U-G</td>
<td>A</td>
<td>166</td>
<td>0.919</td>
<td>Wobble</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 nts (e.g., 0.A69|0.G54,0.C65). The canonical bp name (WC or wobble) is added at the end. The following two lines respectively list the relative orientation (+ or -) of the hydrogen bonds.

**Figure 12:** Two types of A-minor motifs presumably to be specific to adenine. (A) Type I, where the O2' and N3 atoms of adenine lie inside the minor-groove edge of the canonical base pair; (B) Type II, where the O2' of adenine lies outside but N3 remains inside the minor-groove edge. The images were produced with Jmol.

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 nts (e.g., 0.A69|0.G54,0.C65). The canonical bp name (WC or wobble) is added at the end. The following two lines respectively list the relative orientation (+ or -) of the hydrogen bonds.

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 nts (e.g., 0.A69|0.G54,0.C65). The canonical bp name (WC or wobble) is added at the end. The following two lines respectively list the relative orientation (+ or -) of the hydrogen bonds.
Thus for the type I A-minor motif #4 listed above, +0.\text{G}54 means that 0.\text{G}54 and 0.\text{A}69 have similar faces (i.e., the dot product of the z-axes of their base reference frames is positive). Conversely, 0.\text{A}69 and 0.\text{C}65 have opposite faces, so −0.\text{C}65 is placed in the next line.

As one can see from the above listing, in type I A-minor motifs the adenine interacts with both nucleotides of the pair. In type II or O, on the other hand, the adenine interacts with only one of paired nucleotides.

### 3.4.3 Ribose zippers

First described in the P4-P6 domain of a group I intron [24], the ribose zipper is a tertiary interaction that is important for RNA packing. In DSSR, a ribose-zipper motif is defined by two or more (very rare) consecutive H-bonding interactions between ribose 2′-hydroxyl groups from two RNA fragments (see Figure 13).

In 1jj2, DSSR detects a total of 46 ribose zippers, all consisting of 4 nts. The first twelve zippers are listed below, and the top one is illustrated in Figure 13.

<table>
<thead>
<tr>
<th>List of 46 ribose zippers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 nts=4</td>
</tr>
<tr>
<td>2 nts=4</td>
</tr>
<tr>
<td>3 nts=4</td>
</tr>
<tr>
<td>4 nts=4</td>
</tr>
<tr>
<td>5 nts=4</td>
</tr>
<tr>
<td>6 nts=4</td>
</tr>
<tr>
<td>7 nts=4</td>
</tr>
<tr>
<td>8 nts=4</td>
</tr>
<tr>
<td>9 nts=4</td>
</tr>
<tr>
<td>10 nts=4</td>
</tr>
<tr>
<td>11 nts=4</td>
</tr>
<tr>
<td>12 nts=4</td>
</tr>
</tbody>
</table>

3.4.4 Kink turns

The kink-turn (k-turn), first characterized in the large ribosomal subunit of *H. marismortui* [25], is a widespread structural motif in RNA. The motif contains a sharp kink in the RNA helix, with an asymmetric internal loop flanked by C–G bps on one side and sheared G–A bps on the other. The Lilley laboratory established a systematic nomen-

---

6Because 0.\text{G}54 and 0.\text{C}65 form a WC pair, their corresponding bases must have opposite faces related to that of 0.\text{A}69.
Figure 13: A sample canonical ribose zipper identified in 1jj2. This motif consists of the U26–U27 dinucleotide in one strand, and A1318–G1319 in the other, both on chain ‘0’. The images were produced with Jmol.

clature for nucleotides of k-turns based on their structural studies of the motif [26] and maintains a dedicated database for k-turns.

By default, DSSR defines a 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 undecided cases). The normal type actually includes simple (standard and non-standard) and complex k-turns as defined by Lilley [27], as long as they are involved in asymmetric internal loops. DSSR finds a total of 8 k-turns in 1jj2, the first of which (commonly known as H. marismortui Kt-7) is listed below and depicted in Figure 14.

1 Normal k-turn; iloop#44; between stems [#11,#10]; bending-angle=54
2 C#11[CG 0.C93,0.G81] [GA 0.G97,0.A80] NC#10[CG 0.C100,0.G77]
3 strand1 nts=15; GGGCGAAGAACCAUGG 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
4 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.C93

- Line 1 means this is a normal k-turn, derived from internal loop #44 (which is delineated by stems #11 and #10). The bending angle between the two stems is 54°.
- Line 2 shows that the canonical helix consists of stem #11, with a C–G pair (0.C93 and 0.G81) closing internal loop #44 at one end. The non-canonical helix contains stem #10, with a C–G pair (0.C100 and 0.G77) closing the internal loop (#44) at the
other end. The crucial sheared G–A pair is formed by \(0.G97\) and \(0.A80\) (highlighted with thick lines in Figure 14). See reference [26] for nomenclature for C and NC helices of k-turns.

- Lines 3-4 list the two strands composing the k-turn.

![Figure 14: A normal k-turn identified in 1jj2, commonly known as *H. marismortui* Kt-7. The two strands are colored yellow and red, respectively, for easy visualization of the backbone trajectory. The crucial sheared G-A pair is highlighted with thick lines. The image was produced with Jmol.](image)

### 3.5 Identification and characterization of G-quadruplexes

G-quadruplexes (hereafter referred to as G4) are a common type of higher-order DNA and RNA structures formed from G-rich sequences [28]. The building block of G4 is a tetrad of guanines in a cyclic ‘planar’ alignment, with four G+G pairs (cW+M type, see Figure 15). A G4 is formed by stacking of G-tetrads and stabilized by cations at the center of the layers. G4 structures are polymorphic: the four strands can be parallel or anti-parallel, and loops connecting them can be of different types: lateral (edgewise), diagonal, or propeller (double-chain reversal). Moreover, G4 structures can be intra- or intermolecular, and even contain bulges [29]. Overall, G4 can take a large variety of
Without a robust software tool, it is practically impossible to analyze and annotate the large number of G4 structures in the PDB consistently.

From its initial releases, DSSR was able to detect G-tetrads, and listed them in a separate section from the general multiplets. An example list of G-tetrads for PDB entry 5hix is shown below. Here the classification of G-tetrads follows the recent work of Meier et al. [29]. Technically, it is based on the non-planarity of a G4-tetrad and the ‘simple’ base-pair parameters of its four intra G+G pairs.

<table>
<thead>
<tr>
<th>List of 4 G-tetrads</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. glyco-bond=s-s</td>
</tr>
<tr>
<td>2. groove=w-n-</td>
</tr>
<tr>
<td>3. planarity=0.160</td>
</tr>
<tr>
<td>4. type=other</td>
</tr>
<tr>
<td>5. nts=4 GGGG</td>
</tr>
<tr>
<td>6. A.DG1 ,B.DG4 ,A.DG12 ,B.DG9</td>
</tr>
<tr>
<td>7. 2. glyco-bond=s-s</td>
</tr>
<tr>
<td>8. groove=w-n-</td>
</tr>
<tr>
<td>9. planarity=0.137</td>
</tr>
<tr>
<td>10. type=planar</td>
</tr>
<tr>
<td>11. nts=4 GGGG</td>
</tr>
<tr>
<td>12. A.DD2 ,B.DG3 ,A.DG11 ,B.DG10</td>
</tr>
<tr>
<td>13. 3. glyco-bond=s-s</td>
</tr>
<tr>
<td>14. groove=w-n-</td>
</tr>
<tr>
<td>15. planarity=0.160</td>
</tr>
<tr>
<td>16. type=other</td>
</tr>
<tr>
<td>17. nts=4 GGGG</td>
</tr>
<tr>
<td>18. A.DG3 ,B.DG2 ,A.DG10 ,B.DG11</td>
</tr>
<tr>
<td>19. 4. glyco-bond=s-s</td>
</tr>
<tr>
<td>20. groove=w-n-</td>
</tr>
<tr>
<td>21. planarity=0.293</td>
</tr>
<tr>
<td>22. type=bowl</td>
</tr>
<tr>
<td>23. nts=4 GGGG</td>
</tr>
<tr>
<td>24. A.DG4 ,B.DG1 ,A.DG9 ,B.DG12</td>
</tr>
</tbody>
</table>

As of v1.7.0-2017oct19, DSSR has integrated existing features, and created a new module to automatically identify and extensively characterize G4 structures. The underlying algorithms have been further refined in following DSSR releases, and checked against all nucleic-acid-containing structures in the PDB. From v1.8.1-2018oct09, DSSR is considered full-fledged for the analysis of G4 structures. Highlights of the underlying DSSR algorithms and its salient features on G4 will be reported in a dedicated publication.

Five examples of DSSR output are illustrated here, with two G4 structures in schematics representation in Figure 15. Characterizations of four representative G4 examples (PDB entries 2m4p, 2hy9, 5hix, and 6h1k) are shown below, highlighting the different strand polarities, distinct loop types, and varying groove dimensions that are automatically extracted by DSSR.

2m4p

| 1. stem#1[#1] layers=3 INTRA-molecular loops=3 descriptor=3(-P-P-P) note=parallel(4+0) UUUU parallel bulged-strands=1 |
| 2. glyco-bond=----- | groove=----- |
| 3. pm(>,forward) area=8.38 rise=3.65 twist=33.7 nts=4 GGGG A.DG3 ,A.DG8 ,A.DG12 ,A.DG16 |
| 4. 2. glyco-bond=----- | groove=----- |
| 5. pm(>,forward) area=10.73 rise=3.26 twist=31.9 nts=4 GGGG A.DG5 ,A.DG9 ,A.DG13 ,A.DG17 |
| 6. strand#1 U DNA glyco-bond=----- | nts=3 GGG A.DG3 ,A.DG5 ,A.DG6 bulged=nts=1 T A.DT |
| 7. strand#2 U DNA glyco-bond=----- | nts=3 GGG A.DG5 ,A.DG9 ,A.DG10 |
| 8. strand#3 U DNA glyco-bond=----- | nts=3 GGG A.DG12 ,A.DG13 ,A.DG14 |
| 9. strand#4 U DNA glyco-bond=----- | nts=3 GGG A.DG16 ,A.DG17 ,A.DG18 |
| 10. loop#1 type=propeller strands=#1,#2 | nts=1 T A.DT |
| 11. loop#2 type=propeller strands=#2,#3 | nts=1 T A.DT |
| 12. loop#3 type=propeller strands=#1,#4 | nts=1 T A.DT |
| 13. 2hy9

| 14. stem#1[#1] layers=3 INTRA-molecular loops=3 descriptor=3(-P-Lw-Ln) note=hybrid-1(3+1) UUDU anti-parallel |
| 15. glyco-bond=ss- | groove=wm- |
| 16. pm(>,outward) area=13.69 rise=3.15 twist=18.9 nts=4 GGGG 1.DG4 ,1.DG10 ,1.DG18 ,1.DG22 |
| 17. 2. glyco-bond=ss- | groove=wm- |
| 18. pm(>,forward) area=13.40 rise=3.07 twist=27.8 nts=4 GGGG 1.DG5 ,1.DG11 ,1.DG17 ,1.DG23 |
| 19. strand#1 U DNA glyco-bond=ss | nts=3 GGG 1.DG4 ,1.DG5 ,1.DG6 |
| 20. strand#2 U DNA glyco-bond=ss | nts=3 GGG 1.DG10 ,1.DG11 ,1.DG12 |
| 21. strand#3 U DNA glyco-bond=ss | nts=3 GGG 1.DG18 ,1.DG17 ,1.DG16 |
| 22. strand#4 U DNA glyco-bond=ss | nts=3 GGG 1.DG22 ,1.DG23 ,1.DG24 |
| 23. loop#1 type=propeller strands=#1,#2 | nts=3 TTA A.DT |
| 24. loop#2 type=propeller strands=#1,#3 | nts=3 TTA A.DT |
| 25. loop#3 type=propeller strands=#1,#4 | nts=3 TTA A.DT |
Figure 15: Representative G4 structures. Upper left: atomic structure of G-tetrad, the building block of G4 structures. Here the green ‘square’ is created by connecting the C1' atoms of the guanosines, and it is used to simplify the representation of G4 structures of PDB entries 2m4p (lower left) and 5dww (right). Note that the asymmetric unit of 5dww contains four biological units, which are coaxially stacked in two columns.
Here, the **descriptor** component follows the nomenclature for canonical G4 structures, recently proposed by the group of Webba da Silva [30]. This topological framework provides a consistent description of intramolecular G4 structures with three loops. Note that in DSSR, capital letters L, P, and D are used for lateral, propeller, and diagonal loops, and the small case symbols n and w represent narrow and wide groove width. So for PDB entry 2hy9, the topological descriptor is 3(-P-Lw-Ln) in DSSR, instead of 3(-p-lw-ln) where lower case letters l/p are used for lateral/propeller loops and subscripts w/n for wide/narrow groove widths [30], respectively.

As of DSSR v1.8.2-2018oct20, loops not involved in G4 stems (as shown above) are also automatically detected and listed. An example is shown for PDB entry 6h1k. The most notable is the V-shaped loop (V-loop) [31], where one of the four strands is broken (thus no longer qualified as a G4 stem).

**Figure 16** shows five G4 structures with the topological descriptors, including two cases with V-loops. The figure also serves to highlight the simplicity and effectiveness of the DSSR-enabled, PyMOL rendered, schematic block representations.

Listing of DSSR output for PDB entry 5dww, showing the differences of a G4-helix vs. a G4-stem. Generally speaking, a G4-helix could be an assemble of two G4-stems via coaxial stacking of two chains (see **Figure 15**).

<table>
<thead>
<tr>
<th>List of 2 non-stem G4 loops (INCLUDING the two terminal nts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 type=lateral helix=#1 nts=5 GACTG A.DG21,A.DG22,A.DG23,A.DG24,A.DG25</td>
</tr>
<tr>
<td>2 type=V-shaped helix=#1 nts=4 GGGG A.DG25,A.DG26,A.DG27,A.DG28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List of 2 G4-helices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Note: a G4-helix is defined by stacking interactions of G4-tetrads, regardless of backbone connectivity, and may contain more than one G4-stem.</td>
</tr>
<tr>
<td>2 helix#1(2) stem=([#1,#2]) layers=6 inner-solenar</td>
</tr>
<tr>
<td>3 glyco-bond=----- groove=----- mp(&lt;&lt;,backward) area=10.64 rise=3.33 twist=29.0 nts=4 GGGG A.DG21,A.DG22,A.DG23,A.DG24,A.DG25</td>
</tr>
<tr>
<td>5 Bulges are also allowed along each of the four strands.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>List of 4 G4-stems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Note: a G4-stem is defined as a G4-helix with backbone connectivity.</td>
</tr>
<tr>
<td>2 DSSR-Jmol · DSSR-PyMOL DSSR Web Interface</td>
</tr>
</tbody>
</table>
Figure 16: Five G4 structures with topological descriptors [30], including two with V-loops [31]. The structural features were automatically derived by DSSR, and the images were rendered with PyMOL.
3.6 Detection and characterization of i-motifs

Hemi-protonated C+C pairs (of type tW+W, Saenger XV) can form an intercalated four-stranded structure (termed the “i-motif”), consisting of two parallel duplexes. The i-motif has potential roles in gene regulation and can serve as building blocks in nano-structure design.

From early on, DSSR can identify the C+C pairs in i-motifs, which are embedded in DSSR-designated helices. While the components are already there, the i-motifs possess special structural features that deserve additional treatments. As of v1.6.1-2016aug22, DSSR can automatically identify and characterize i-motifs, dissecting each into the four strands, (potentially) three loops, and two double helices, among other features.

Using PDB entry 1a83 as an example (see Figure 17), the default output is as below:

With 5′ and 3′ ends of the DNA chain labeled as shown in Figure 17(A), the meaning of the output should be easy to understand. When the --more option is specified (see below), the base-pair morphology parameters are also available (Section 3.7.3). The intercalation is quantified by the large rise and small twist. In Figure 17(A), the large buckle of the top C+C pair is also obvious. For comparison, PDB entry 2n89 which has an interstand i-motif is presented in Figure 17(B).
Figure 17: Two i-motifs in cartoon-block representations oriented in the best view. (A) PDB entry 1a83 with an intrastrand i-motif and three loops; (B) PDB entry 2n89 with an interstrand i-motif and no loops.
3.7 The --more (i.e., --verbose) option

The --more option triggers additional parameters 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. In what follows, the excerpted portions are based on PDB entry 1msy with the following DSSR command:

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

3.7.1 Extra characterization of base pairs

The G–U wobble pair formed by A.G2648 and A.U2672 (the second bp on Page 14) is used as an example of the additional information provided about base-pair geometry and hydrogen bonding with the --more option (see Figure 18).

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

- Line 1 – specification of the bp, as shown previously on Page 14.
- Line 2 – the first bracket [-167.8(anti)  C3'-endo lambda=42.1] corresponds to A.G2648, and it contains three items: -167.8(anti) is the χ torsion angle formed by O4'-C1'-N9-C4; C3'-endo is the sugar pucker (as is the norm for RNA), and λ (lambda) is the angle N9-C1'-C1' (A.U2672). The second bracket [-152.8(anti)  C3'-endo lambda=68.6] corresponds to A.U2672, with similar meanings for parameters, except that χ is defined by O4'-C1'-N1-C2, and λ is the angle N1-C1'-C1' (A.G2648).
- Line 3 – lengths of three virtual bonds (C1'-C1', N1-N9, C6-C8), and the virtual torsion angle (C1'-N1-N9-C1'). Note here N1/N9 are general terms, referring to either N1 of pyrimidines (Y: C/U/T) or N9 of purines (R: A/G) as appropriate. Similar conventions apply for the labeling of C6/C8.
- Line 4 – detailed H-bonding information (atom names, types, and H-bond distances in brackets).
• Line 5 – inter-base-angle (9°), and a ‘simple’ set of four bp parameters (shear, stretch, buckle and propeller) which are easier to understand than the six rigorous rigid-body parameters (listed below) for non-canonical bps, especially when opening is ~180°.

• Line 6 – the six bp parameters in the order of shear, stretch, stagger, buckle, propeller, and opening (see Figure 6).

Figure 18: Molecular image of the G–U wobble pair formed by A.G2648 and A.U2672 in 1msy with additional parameters (bond lengths, angles, and torsions), as calculated with the DSSR --more option, labeled. The image was produced with Jmol.

3.7.2 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 C1’ and RN9/YN1 atom pairs along each strand [32]. The result for the 1msy stem listed on Page 22 is shown below.

<table>
<thead>
<tr>
<th></th>
<th>helical-rise:</th>
<th>2.60(0.18)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>helical-radius:</td>
<td>9.12(0.79)</td>
</tr>
<tr>
<td>3</td>
<td>helical-axis:</td>
<td>-0.776 -0.167 -0.608</td>
</tr>
<tr>
<td>4</td>
<td>point-one:</td>
<td>24.637 21.051 22.830</td>
</tr>
<tr>
<td>5</td>
<td>point-two:</td>
<td>16.686 19.344 16.602</td>
</tr>
</tbody>
</table>

• Line 1 – here the ‘helical-rise’ line, with numbers 2.60(0.18), represents the average helical rise (2.60) between successive nucleotides in the designated structural
unit and its standard deviation (sd, 0.18) in Å. For a perfectly regular DNA/RNA helix, the sd would be zero. RNA models generated with the 3DNA ‘fiber’ program\(^7\), for example, are characterized by numbers 2.55(0.00). In DSSR (and 3DNA), the sd 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 21), a ‘*’ is appended at the end\(^8\), serving as a reminder that the best-fitted linear helical axis may not be meaningful.

- Line 2 – the ‘helical-radius’ line, 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 A-form RNA, and for A- and B-form DNA. Textbooks normally list the diameter of the DNA duplex as ~20 Å.

- Line 3 – the ‘helical-axis’ line, with three numbers ‘-0.776 -0.167 -0.608’, provides the normalized helical axis vector (in the original coordinate frame). This vector can be used to calculate DNA bending angles (as in DNA-protein complexes), or to quantify the relative orientation between any two fairly straight helices/stems [9].

- Lines 4-5 – the following two lines (starting with ‘point-one’ and ‘point-two’) designate the end points (in the original coordinate frame) of the helical axis of the helix/stem. The two points can be added to the original PDB coordinate file for visualization of the helical axis (e.g., in Jmol/PyMOL) or for rendering of helical regions as cylinders [9].

### 3.7.3 Base-pair morphology parameters for helices/stems

The second dinucleotide step consisting of bps C–G (A.C\( \text{2649} \) with A.G\( \text{2671} \)) and U–A (A.U\( \text{2650} \) with A.A\( \text{2670} \)) in the stems section of 1msy is used as an example of the geometry information available for each base-pair step. Six extra lines are available (lines 2-7 in the listing) with the ‘--more’ option. For the definitions of base and step parameters, see Figure 6 and the two 3DNA papers [8, 9].

---

\(^7\)Here is a sample run: fiber -rna -seq=AAAAUUUGGGCC F RNA.pdb

\(^8\)The 3DNA ‘analyze’ program simply does not output the best-fitted linear helical axis or its two end points (see below) if the helix is strongly curved.
• Line 2 – the six bp parameters in the order of shear, stretch, stagger, buckle, propeller, and opening for the C–G (A.C2649 with A.G2671) pair.

• Line 3 – the six step parameters in the order of shift, slide, rise, tilt, roll, and twist for the dinucleotide step between bps C–G (A.C2649 with A.G2671) and U–A (A.U2650 with A.A2670), calculated based on a middle-step frame [8].

• Line 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, calculated based on a middle-helical frame [8].

• Line 5 – the six bp parameters in the order of shear, stretch, stagger, buckle, propeller, and opening for the U–A (A.U2650 with A.A2670) pair.

• Line 6 – rise and twist derived using the two consecutive C1′–C1′ vectors, each defined by a bp in the dinucleotide step. These two parameters are related to the middle-step frame used to calculate the six step parameters (line 3). When non-WC bps are involved, they normally make more intuitive sense than the corresponding rise and twist values reported on line 3 [8].

• Line 7 – helical rise and helical twist derived using the two consecutive C1′–C1′ vectors, each defined by a bp in the dinucleotide step. These two parameters are related to the middle-helical frame used to calculate the six helical parameters listed on line 4.

3.8 The --non-pair option

With the --non-pair option, DSSR identifies H-bonding and base-stacking interactions between two nts, excluding those duos that have already formed a bp. For 1msy, the running command and relevant results are given below.
As in 3DNA [8], base-stacking is quantified as the area (in Å²) of the overlapped polygon defined by the two bases of the interacting nts, where the base atoms are projected onto the mean base plane. In the output file, values in parentheses measure the overlap of base ring atoms only, and those outside parentheses include exocyclic atoms on the ring. Base-stacking interactions 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, as defined by the direction of the z-axis of the standard base reference frame. The symbols (>>, <<, <<, and >>) follow Major et al., except pm(>>) is called forward instead of upward, and mp(<<) backward instead of downward [33]. Moreover, the inter-base angle is reported; closer to zero means the two bases are nearly parallel.

The H-bonding information should be self-explanatory, except for noting the convention: when a pair of donor/donor or acceptor/acceptor atoms fulfills the DSSR H-bond definition, a ‘*’ is used instead of a ‘-‘ to connect the atoms (e.g., entry#10: 04’*04’ [3.05], line 13 in the listing).

The word connected, if any, means that the two nucleotides are connected by a covalent phosphodiester linkage. The last item gives the minimum distance between base atoms.

The mean plane is the plane with a normal vector defined as the mean z-axis of the standard base reference frames of the two interacting nts.
3.9 The --json option

As of v1.3.0-2015aug27, DSSR has the --json option for outputting analysis results that is strictly compliant with the standard JSON data exchange format. The single JSON file contains numerous DSSR-derived structural features, including those in the default main output, backbone torsions (in dssr-torsions.txt), and hydrogen bonds (via --get-hbond).

According to the official JSON website:

"JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language···"
JSON is a text format that is completely language independent... These properties make JSON an ideal data-interchange language.”

Indeed, the JSON output file makes DSSR readily accessible for integration with other bioinformatics tools or convenient exploration on the command line. Using the classic yeast phenylalanine tRNA 1ehz as an example (1ehz.pdb), let’s go over some simple use-cases. Note the following examples take advantages of jq, a lightweight and flexible 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 2 steps combined
```

With file 1ehz-dssr.json in hand, one can easily extract DSSR-derived structural features of interest:

```
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 hydrogen bonds
jq .helices 1ehz-dssr.json
jq .stems 1ehz-dssr.json
jq .ntParams 1ehz-dssr.json
# list of nucleotide parameters, including torsion angles and suites
jq '.ntParams[] | select(.is_modified)' 1ehz-dssr.json
# select nucleotide id, delta torsion, sugar puckering and cluster of suite name
jq '.ntParams[] | {nt_id, delta, puckering, cluster}' 1ehz-dssr.json
# same selection as above, but in 'comma-separated-values' (csv) format
jq -r '.ntParams[] | [.nt_id, .delta, .puckering, .cluster] | @csv' 1ehz-dssr.json
```

Here is the result of running jq (v1.5) on selecting multiplets:

```
# jq .multiplets 1ehz-dssr.json
[
  {
    "index": 1,
    "num_nrs": 3,
    "nts_short": "UAA",
    "nts_long": "A.U8,A.A14,A.A21"
  },
  {
    "index": 2,
    "num_nrs": 3,
    "nts_short": "AUA",
    "nts_long": "A.A9,A.U12,A.A23"
  },
  {
    "index": 3,
    "num_nrs": 3,
    "nts_short": "gCG",
    "nts_long": "A.2MG10,A.C25,A.G45"
  }
]```

DSSR-Jmol · DSSR-PyMOL · DSSR Web Interface
With the JSON file, DSSR can now be connected with the bioinformatics community in a ‘structured’ way, with a clearly delineated boundary. Now I can enjoy the freedom of refining the default main output format, without worrying too much about breaking third-party parsers or backward compatibility. Moreover, I no longer need to write an adapter for each integration of DSSR with other tools. It is so nice!

### 3.10 The --pair-only option

DSSR provides far more features than a typical user may normally need. The DSSR --pair-only option outputs just base-pairing information, the most fundamental feature for DNA/RNA structural analysis and annotation. It can be combined with the --more or --json option. DSSR runs approximately 10 times faster with the --pair-only option than without it (in default).

### 3.11 The --nmr option

The --nmr option (--md or --ensemble acceptable) has been introduced for the analysis of an ensemble of NMR structures, as deposited in the PDB. The input file can be in the classic PDB format where each model is delineated with a MODEL/ENDMDL pair, or the new standard PDBx/mmCIF format where each ATOM/HETATM record has an associated model number.

Using PDB id 2n2d as an example, two simple usages are as follows:

```bash
x3dna-dssr -i=2n2d.pdb --nmr -o=2n2d-model.out
x3dna-dssr -i=2n2d.pdb --nmr --json -o=2n2d.json
jq '.models[].parameters.num_Gquartets' 2n2d.json
```

The top-level skeleton of the JSON output is shown below. Note that each member of the “models” array contains three items: an auto-incremental “index” (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 separately. Normally, “index” and “model”
match each other, as is the case for 2n2d. However, it is conceivable that models do not start from one, or the numbers are not continuous. In such cases, “index” and “model” will no longer have the same value for each entry.

```json
{
  "input_file": "2n2d.pdb",
  "num_models": 10,
  "models": [
    {
      "index": 1,
      "model": 1,
      "parameters": { ... }
    },
    ...
    {
      "index": 10,
      "model": 10,
      "parameters": { ... }
    }
  ],
  "start_at": "Fri Oct 9 23:19:10 2015",
  "finish_at": "Fri Oct 9 23:19:12 2015",
  "time_used": "00:00:00:02"
}
```

The --json option makes it easy to parse the output of multiple models pragmatically. In addition to NMR structures, trajectories from molecular dynamics (MD) simulations can also be processed: While the popular MD packages (AMBER, GROMACS, CHARMM etc.) all have specialized binary formats for trajectories or topology files, it should be easy to convert them to the standard PDB or PDBx/mmCIF text format. The combination of --nmr and --json would undoubtedly render DSSR more accessible to the MD community.

### 3.12 The --cartoon-block option

The 3DNA blocview utility program automatically generates a schematic base-block representation of DNA/RNA structures in the most extended view. For small to medium size nucleic acid structures, the resultant images are simple yet highly revealing for base-pairing and stacking interactions. As a result, blocview has become widely used, including in the Nucleic Acid Database (NDB) and PDB. The blocview images are so characteristic that I can identify them right away in articles, PowerPoint slides, or websites.

Initially written more than ten years ago as a Perl script, blocview was later converted to Ruby in 3DNA v2.1. In addition to 3DNA utility programs, it also employs several then-popular third-party software, including Molscript (for the DNA/RNA backbone ribbon and protein cartoons), render in Raster3D, and convert from ImageMagick. Although
the .r3d files can be fed directly into PyMOL for ray-tracing to get high-resolution PNG images, that was just a nice side effect.

Over the years, PyMOL has become a standard tool for molecular visualization, notable especially for creating publication quality graphics images. The --cartoon-block option has been added for better integration between DSSR and PyMOL (no more other dependencies). Here ‘cartoon’ refers to the native PyMOL cartoon representations of proteins and DNA/RNA backbone, and ‘block’ stands for the base or (Watson-Crick) base-pair rectangular blocks. The basic usage is very straightforward, as illustrated by the following examples (Figure 19).

```
1  x3dna-dssr -i=355.d.pdb --cartoon-block=orient -o=355.d.pml
2  x3dna-dssr -i=4rim.pdb1 --symmetry --cartoon-block=orient -o=4rim.pml
3  x3dna-dssr -i=1oct.pdb --block-file=wc-minor --cartoon-block=orient -o=1oct.pml
4  x3dna-dssr -i=467d.pdb1 --symmetry --block-file=wc --cartoon-block=orient -o=467d.pml
```

Please note the following points:

- The output is a script file (.pml) that can be fed directly into PyMOL, or easily customized. The term orient specified in the --cartoon-block option is a PyMOL command to set a molecule in the best view. Note that the .pml script also turns the scenes by -90° to make the longest axis vertical. The images shown in Figure 19 are ray-traced, in PNG format.

- The --symmetry option is used for 4rim and 467d to show the biological unit. Otherwise, only the asymmetric unit (delineated in the first MODEL/ENDMDL pair) would be displayed.

- The term wc in --block-file is used to take each watson-crick base pair as a single, longer rectangular block instead of as two separate base blocks. The term minor colors the minor groove edge of a base or base-pair black.

- See Section 5.3 on the dssr_block plugin for PyMOL which streamlines the creation of schematic representations of base rectangular blocks.

### 3.13 The --view option

The --view option resets a structure with regard to the principle moment of inertia. By default, only atoms of (DNA/RNA) nucleotides are used, and outputs structure in the most
Figure 19: Four representative cartoon-block images ray-traced with PyMOL using DSSR generated .pml files. (A) B-DNA dodecamer (PDB id: 355d). (B) Native structure of intercalation-locked DNA tetraplex (PDB id: 4rim). (C) Crystal structure of the Oct-1 POU domain bound to an octamer site (PDB id: 1oct). (D) Crystal structure of a DNA Holliday junction (PDB id: 467d).
extended orientation (vertically). It has similar functionality to the 3DNA \texttt{rotate.mol}
utility program, and comes with several variants. Combining the options \texttt{--view} and
\texttt{--cartoon-block}, one can achieve similar effect of the 3DNA \texttt{blocview} script.

Using the classic yeast phenylalanine tRNA \texttt{1ehz} as an example, one can run DSSR
with the \texttt{--view} option as follows:

\begin{verbatim}
1 x3dna-dssr -i=1ehz.pdb --view -o=1ehz-view.pdb
2 x3dna-dssr -i=1ehz-view.pdb --cartoon-block -o=1ehz-view.pml
3
4 # the above two steps can also be combined as shown below:
5 x3dna-dssr -i=1ehz.pdb --cartoon-block=view -o=1ehz-bestview.pml  # same as above
6 x3dna-dssr -i=1ehz.pdb --blocview -o=1ehz-bestview.pml  # mimic the 'blocview' script
7
8 # load 1ehz-view.pml or 1ehz-bestview.pml into PyMOL to see the results
\end{verbatim}

\section{The \texttt{--blocview} option}

The \texttt{--blocview} option (\texttt{--blockview} or \texttt{--block-view} also accepted) is a shorthand
form that combines features from the \texttt{--cartoon-block} and \texttt{--view} options. It instructs
DSSR to generate PyMOL inputs to render a cartoon-block image in the most extended
view. The option replaces the original \texttt{blocview} script in 3DNA v2.x and v1.x, with
significant new features. Moreover, the DSSR \texttt{--blocview} option no longer depends on
MolScript and Raster3D. See Section 3.13 above for examples.

\section{The \texttt{--frame} option}

The \texttt{--frame} option can be used to reorient a structure with reference to a specific base
(or base-pair) reference frame, or the \textit{middle} frame of two related frames. It has similar
functionality to the 3DNA \texttt{frame.mol} utility program, but with an easier user-interface.
Using the classic Dickerson DNA dodecamer \cite{16} \texttt{355d} as an example, one can run DSSR
with the \texttt{--frame} option as follows:

\begin{verbatim}
1 # 1...5...8....
2 # strand-A 5'-GGCGAATTCCGG-3'
3 # strand-B 3'-GGCGGATTCGCC-5'
4
5 # with the minor-groove of pair C1-G24 facing the viewer
6 x3dna-dssr -i=355d.pdb --frame=A.1-wc-minor -o=355d-bp1.pdb
7 x3dna-dssr -i=355d-bp1.pdb --cartoon-block -o=355d-bp1.pml
8
9 # with the minor-groove of the middle AATT tract facing the viewer
10 x3dna-dssr -i=355d.pdb --frame='A.5-wc A.8-wc minor' -o=355d-aatt.pdb
\end{verbatim}
The shorthanded notation \texttt{A.1} (also \texttt{A.5} and \texttt{A.8}) means nucleotide \#1 (or \#5 and \#8) on chain A. In the most common case (as in 355d), the combination of chain and residue\# can be used to unambiguously identify a nucleotide. Generally, one may need up to five fields as detailed in Section 3.17.4 to specify a nucleotide.

For further examples, please see the section titled DSSR-NAR paper (on the 3DNA Forum) that is dedicated to the DSSR paper published in \textit{Nucleic Acids Research} (NAR) in 2015. Pay particular attention to the post titled “RNA cartoon-block representations with PyMOL and DSSR”.

### 3.16 The \texttt{--get-hbond} option

H-bonding interactions are crucial for defining RNA secondary and tertiary structures. DSSR/3DNA contains a geometrically based algorithm for identifying H-bonds in nucleic-acid or protein structures given in \texttt{.pdb} or \texttt{.cif} format. Over the years, the method has been continuously refined, and it has served its purpose quite well. As of DSSR v1.1.1-2014apr11, this functionality is directly available through the option \texttt{--get-hbond}. By default, the output only includes H-bonds involving nucleotides.

Running the above command, the output for 1msy is as listed below. The first line gives the header (“\texttt{# H-bonds in ‘1msy.pdb’ identified by DSSR...}”). The second line provides the total number of H-bonds (39) identified in the structure. Afterwards, each line consists of 8 space-delimited columns used to characterize a specific H-bond. Using the first one (line 3 in the listing) as an example, the meaning of each of the 8 columns is:

1. The serial number, as denoted in the \texttt{.pdb} or \texttt{.cif} file, of the first atom (15) of the H-bond.

2. The serial number (578) of the second H-bond atom.

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’ for a donor/acceptor (such as the 2'-hydroxyl oxygen) with any other atom; ‘x’ for a donor-donor or acceptor-acceptor pair (as in #17, line 19 in the listing); ‘?’ if the donor/acceptor status of any H-bond atom is unknown.

5. Distance in 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-bond atom (O4@A.U2647).

8. Identifier of the second H-bond atom (N1@A.G2673).

# H-bonds in '1msy.pdb' identified by DSSR, Xiang-Jun Lu (xiangjun@x3dna.org)

<table>
<thead>
<tr>
<th>#</th>
<th>H-bond</th>
<th>Type</th>
<th>Distance</th>
<th>Atom1</th>
<th>Atom2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15 578 #1</td>
<td>p</td>
<td>2.768</td>
<td>0: N</td>
<td>O4@A.U2647 N1@A.G2673</td>
</tr>
<tr>
<td>2</td>
<td>35 555 #2</td>
<td>p</td>
<td>2.776</td>
<td>0: N</td>
<td>O6@A.U2648 N3@A.U2672</td>
</tr>
<tr>
<td>3</td>
<td>36 554 #3</td>
<td>p</td>
<td>2.826</td>
<td>N: O</td>
<td>N1@A.G2648 O2@A.U2672</td>
</tr>
<tr>
<td>4</td>
<td>55 537 #4</td>
<td>p</td>
<td>2.965</td>
<td>0: N</td>
<td>O2@A.C2649 N2@A.U2671</td>
</tr>
<tr>
<td>5</td>
<td>56 535 #5</td>
<td>p</td>
<td>2.836</td>
<td>N: O</td>
<td>N3@A.C2649 N1@A.G2671</td>
</tr>
<tr>
<td>6</td>
<td>58 534 #6</td>
<td>p</td>
<td>2.769</td>
<td>N: O</td>
<td>N4@A.C2649 O6@A.U2671</td>
</tr>
<tr>
<td>7</td>
<td>76 513 #7</td>
<td>p</td>
<td>2.806</td>
<td>N: N</td>
<td>N3@A.U2650 N1@A.A2670</td>
</tr>
<tr>
<td>8</td>
<td>78 512 #8</td>
<td>p</td>
<td>3.129</td>
<td>0: N</td>
<td>O4@A.U2650 N6@A.A2670</td>
</tr>
<tr>
<td>9</td>
<td>95 492 #9</td>
<td>p</td>
<td>2.703</td>
<td>0: N</td>
<td>O2@A.C2651 N2@A.U2669</td>
</tr>
<tr>
<td>10</td>
<td>96 490 #10</td>
<td>p</td>
<td>2.853</td>
<td>N: N</td>
<td>N3@A.C2651 N1@A.G2669</td>
</tr>
<tr>
<td>11</td>
<td>98 489 #11</td>
<td>p</td>
<td>2.987</td>
<td>N: O</td>
<td>N4@A.C2651 O6@A.U2669</td>
</tr>
<tr>
<td>12</td>
<td>115 466 #12</td>
<td>p</td>
<td>2.817</td>
<td>0: N</td>
<td>O2@A.C2652 N2@A.U2668</td>
</tr>
<tr>
<td>13</td>
<td>116 464 #13</td>
<td>p</td>
<td>2.907</td>
<td>N: N</td>
<td>N3@A.C2652 N1@A.G2668</td>
</tr>
<tr>
<td>14</td>
<td>118 463 #14</td>
<td>p</td>
<td>2.897</td>
<td>N: O</td>
<td>N4@A.C2652 O6@A.U2668</td>
</tr>
<tr>
<td>15</td>
<td>123 151 #15</td>
<td>o</td>
<td>2.622</td>
<td>0: O</td>
<td>OP2@A.U2653 02’@A.A2654</td>
</tr>
<tr>
<td>16</td>
<td>135 443 #16</td>
<td>p</td>
<td>2.898</td>
<td>0: N</td>
<td>O2@A.U2653 N4@A.C2657</td>
</tr>
<tr>
<td>17</td>
<td>147 192 #17</td>
<td>x</td>
<td>3.054</td>
<td>0: O</td>
<td>O4’@A.A2654 04’@A.U2656</td>
</tr>
<tr>
<td>18</td>
<td>158 408 #18</td>
<td>p</td>
<td>2.960</td>
<td>N: N</td>
<td>N6@A.A2654 OP2@A.C2666</td>
</tr>
<tr>
<td>19</td>
<td>173 188 #19</td>
<td>o</td>
<td>2.923</td>
<td>0: O</td>
<td>OP2@A.G2655 OP2@A.U2656</td>
</tr>
<tr>
<td>20</td>
<td>173 378 #20</td>
<td>o</td>
<td>3.093</td>
<td>0: O</td>
<td>OP2@A.G2655 OP2@A.G2655</td>
</tr>
<tr>
<td>21</td>
<td>173 379 #21</td>
<td>o</td>
<td>3.343</td>
<td>0: O</td>
<td>OP2@A.G2655 OP2@A.G2664</td>
</tr>
<tr>
<td>22</td>
<td>181 386 #22</td>
<td>p</td>
<td>2.768</td>
<td>N: O</td>
<td>N1@A.G2655 OP2@A.A2665</td>
</tr>
<tr>
<td>23</td>
<td>183 203 #23</td>
<td>p</td>
<td>2.754</td>
<td>N: O</td>
<td>N2@A.G2655 OP2@A.U2656</td>
</tr>
<tr>
<td>24</td>
<td>183 387 #24</td>
<td>p</td>
<td>2.887</td>
<td>N: O</td>
<td>N2@A.G2655 05’@A.A2665</td>
</tr>
<tr>
<td>25</td>
<td>188 379 #25</td>
<td>p</td>
<td>3.044</td>
<td>0: N</td>
<td>OP2@A.U2653 N4@A.C2657</td>
</tr>
<tr>
<td>26</td>
<td>188 381 #26</td>
<td>p</td>
<td>2.944</td>
<td>0: N</td>
<td>OP2@A.U2653 N2@A.G2664</td>
</tr>
<tr>
<td>27</td>
<td>200 401 #27</td>
<td>p</td>
<td>3.122</td>
<td>0: N</td>
<td>OP2@A.U2656 N6@A.A2665</td>
</tr>
<tr>
<td>28</td>
<td>201 398 #28</td>
<td>p</td>
<td>2.759</td>
<td>N: N</td>
<td>N3@A.U2656 N7@A.A2665</td>
</tr>
<tr>
<td>29</td>
<td>220 381 #29</td>
<td>p</td>
<td>3.035</td>
<td>N: N</td>
<td>N7@A.A2657 N2@A.G2664</td>
</tr>
<tr>
<td>30</td>
<td>223 371 #30</td>
<td>o</td>
<td>2.963</td>
<td>0: N</td>
<td>N6@A.U2657 02’@A.G2664</td>
</tr>
<tr>
<td>31</td>
<td>223 382 #31</td>
<td>o</td>
<td>3.039</td>
<td>0: N</td>
<td>N6@A.U2657 N3@A.G2664</td>
</tr>
<tr>
<td>32</td>
<td>242 358 #32</td>
<td>p</td>
<td>2.821</td>
<td>0: N</td>
<td>O2@A.C2658 N2@A.G2663</td>
</tr>
<tr>
<td>33</td>
<td>243 356 #33</td>
<td>p</td>
<td>2.890</td>
<td>0: N</td>
<td>O2@A.C2658 N1@A.G2664</td>
</tr>
<tr>
<td>34</td>
<td>245 355 #34</td>
<td>p</td>
<td>2.887</td>
<td>0: N</td>
<td>O4@A.C2658 O6@A.U2663</td>
</tr>
<tr>
<td>35</td>
<td>258 305 #35</td>
<td>o</td>
<td>2.604</td>
<td>0: O</td>
<td>OP2@A.G2659 N7@A.A2661</td>
</tr>
<tr>
<td>36</td>
<td>258 308 #36</td>
<td>o</td>
<td>3.264</td>
<td>0: O</td>
<td>OP2@A.G2659 N6@A.A2661</td>
</tr>
<tr>
<td>37</td>
<td>268 315 #37</td>
<td>p</td>
<td>2.973</td>
<td>N: N</td>
<td>N2@A.G2659 OP2@A.A2662</td>
</tr>
<tr>
<td>38</td>
<td>268 327 #38</td>
<td>p</td>
<td>2.864</td>
<td>N: N</td>
<td>N2@A.G2659 N7@A.A2662</td>
</tr>
<tr>
<td>39</td>
<td>371 390 #39</td>
<td>o</td>
<td>2.751</td>
<td>0: O</td>
<td>OP2@A.G2664 04’@A.A2665</td>
</tr>
<tr>
<td>40</td>
<td>550 566 #40</td>
<td>o</td>
<td>3.372</td>
<td>0: O</td>
<td>OP2@A.U2672 04’@A.G2673</td>
</tr>
</tbody>
</table>
In its default settings, DSSR detects 117 H-bonds for 1ehz (yeast phenylalanine tRNA) and 5,756 H-bonds for 1jj2 (H. marismortui large ribosomal subunit), respectively, among nucleotides. Note that the program can identify H-bonds not only in RNA and DNA, but also in protein, or their complexes (as in 1jj2). While there exist dedicated tools for finding H-bonds, such as HBPLUS or HBexplore, DSSR may well be sophisticated enough to fulfill most practical needs.

3.17 Additional options

In addition to the basic functionality documented explicitly via command-line --help (see Section 3.1), DSSR also contains many other options, which are either peripheral or experimental. DSSR aims for simplicity, so I have deliberately documented only these fundamental features.

From experience, I feel a lengthy manual and extra features could be detrimental, especially before a product is well established – they may blur the key message and scare new users away. I personally do not like bloated software. Thus, I have strived to make DSSR ‘minimal’, with least barriers to access. For most users, the default settings should be sufficient to get the job done. For those who want to know more of what DSSR has to offer, the following sections describe some extra options. If you have further questions, please do not hesitate ask them on the 3DNA Forum.

3.17.1 The --u-turn option

DSSR is able to detect two types of U-turns [34] (see Figure 20): the UNR-type [Figure 20(A)] originally identified by Quigley and Rich in yeast tRNA^Phe [15], and the GNRA-type [Figure 20(B)] later established by Jucker and Pardi in GNRA tetraloops [35]. The two U-turns detected in 1ehz are listed below. Note the list of each U-turn includes two additional nts, one at each end.

```
x3dna-dssr -i=1ehz.pdb --u-turn -o=1ehz-uturn.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


This section is based on my blogpost titled “UNR- and GNRA-type U-turns” written when the functionality of U-turn detection in DSSR was added in September 2013.
Figure 20: Different types of U-turns. (A) The classic UNR-type U-turn (1ehz) originally identified in tRNA by Quigley and Rich [15]; (B) The GNRA-type U-turn (1msy) established by Jucker and Pardi [35]; (C) A novel GCAG-type U-turn detected in the *H. marismortui* large ribosomal subunit (1jj2). The images were produced with Jmol.

As the name implies, a U-turn is characterized by a chain reversal of the RNA backbone (normally within a few nts). Among other factors, the U-turn is stabilized by two key H-bonding interactions involving O2'/OP2 and base atoms, as illustrated by the thin black lines in Figure 20.

Thirty-four U-turns are found in 1jj2: in addition to the well documented UNR- and
GNRA-type U-turns, DSSR also finds other variants. An example is shown in Figure 20(C), where the U-turn is formed via a GCAG fragment instead of a GNRA tetraloop. Here, the N1 (not N2) atom of 0.G1809 makes an H-bond with OP2 of 0.G1812. The 0.G1809 N2 atom is H-bonded to 0.G1812 O5’ to add further stabilization to the U-turn. Unlike in GNRA-type U-turns where the G and A form a sheared G–A pair, the two Gs (0.G1809 and 0.G1812) in the GCAG-type U-turn identified here do not form a bp.

An examination of the chemical structures of nitrogenous bases shows clearly other possible ways to connect RNA base donors to the phosphate oxygen acceptors. DSSR allows for the exploration of such variations, and more.

3.17.2 The --po4 option

The phosphate group in RNA is negatively charged, with four oxygen atoms that could serve as H-bond acceptors or be in coordination with metals. The exocyclic OP2 (and OP1, to a less extent) atoms are especially prominent in H-bonding interactions or stacking over base rings (as a cap, see Section 3.2.8), as shown above in Figure 20 for the various types of U-turns. The relevant result of running DSSR on 1ehz with the --po4 option is listed below.

```
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]"
15. A.U50 OP1-hbonds [1]: "O2'@A.U47[2.71]"
16. A.S57 OP2-cap: "A.PSU56"
17. A.MA58 OP2-hbonds [1]: "N3@A.PSU56[2.77]"
18. A.C60 OP1-hbonds [1]: "N4@A.C61[3.12],OP2-hbonds [1]: "O2'@A.1MA58[2.42]"

Here entry #1 (line 4 in the listing) means the OP1 atom on A.U7 is in coordination with MG580 at a distance of 2.60 Å. Entry #2 (line 5 in the listing) means the OP2 atom on A.A9 is H-bonded with the N4 atom of A.C13, at a distance of 3.01 Å. When both OP1
and OP2 atoms from a given nt are involved in H-bonding interactions, as in #14 (line 17 in the listing) for A.5MC49 and #18 (line 21 in the listing) for A.C60, the two sets of H-bonds are listed separately.

### 3.17.3 The --pair-list option

By default, DSSR automatically identifies all base pairs that fulfill some simple geometric criteria. While the criteria have been extensively tested and work well in real-world applications, the resultant base pairs sometimes may not be as desired. No size fits all – users may find certain pairs missing or spurious from the default DSSR output. That’s where the --pair-list option comes in. For the convenience of users, the option has two modes: writing or reading. The intended use-case is illustrated in detail below, using 1msy as an example.

- **Writing mode:** by simply specifying the --pair-list option, DSSR will output an additional auxiliary file, dssr-pairs.txt.

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

  The file dssr-pairs.txt contains a list of bps involved in the formation of double helices, just as the default output from the 3DNA find pair program. It has a simple, intuitive format: the first line contains the numbers of bps in the list, followed by as many lines each with a couple of sequential numbers specifying the two bases in a pair. Empty lines (those containing only white spaces) are ignored. In each line, the # and following characters are comments, and can be disregarded.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>number of pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>number of pairs</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>A.U2647 A.G2673</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>A.G2648 A.U2672</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>A.C2649 A.G2671</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>A.U2650 A.A2670</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>A.C2651 A.G2669</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>A.C2652 A.G2668</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>A.U2653 A.C2667</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>A.A2654 A.C2666</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>A.U2655 A.A2665</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>A.A2657 A.G2664</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>A.C2658 A.G2663</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>A.G2659 A.A2662</td>
</tr>
</tbody>
</table>

  Note that the list for 1msy includes 12 base-pairs (out of the 13 shown on Page 14), as in the helix (see Section 3.2.4).
The output file `dssr-pairs.txt` can be easily edited (i.e., by removing or adding bps) as appropriate, renamed (e.g., to `my-pairs.txt` to avoid being overwritten), and then fed back into DSSR in the reading mode (see below). For the sake of illustration, I’ve removed the two beginning G–U pairs and some inline comments, as shown below.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>25</td>
<td># 3</td>
<td>A.C2649</td>
<td>A.G2671</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td># 4</td>
<td>A.U2650</td>
<td>A.A2670</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>23</td>
<td># 5</td>
<td>A.C2651</td>
<td>A.G2669</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td># 6</td>
<td>A.C2652</td>
<td>A.G2668</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>21</td>
<td># 7</td>
<td>A.U2653</td>
<td>A.C2667</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>18</td>
<td># 10</td>
<td>A.A2657</td>
<td>A.G2664</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>17</td>
<td># 11</td>
<td>A.C2658</td>
<td>A.G2663</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>16</td>
<td># 12</td>
<td>A.G2659</td>
<td>A.A2662</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Reading** mode: by providing a file with the list of pairs to the `--pair-list` option, DSSR will use these specified bps instead identifying them *ab initio*. All following steps of DSSR are exactly as before, e.g., for deriving multiplets, helices/stems, and loops etc.

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

By combining the writing and reading modes in DSSR, users can easily control the pairs to be analyzed. A similar pattern of flexibility is achieved by the `find_pair/analyze` pair of programs in 3DNA.

### 3.17.4 The `--idstr` option

DSSR has been designed with simplicity in mind, with sensible default settings targeted towards the most common use cases. Powerful users, however, are more than likely to want to build upon DSSR and to parse its output files for specific applications. The option `--idstr=long` makes it easy to parse DSSR findings by providing strictly formatted id strings. If you would like to build a more streamlined connection between DSSR and your software, please contact the author directly. On the other hand, the option `--idstr=short` lists just residue name and number, for the simplest cases.

For example, the listings of the first 1msy bp without (default) and with the option `--idstr=long` or `--idstr=short` are given below for comparison. Note that the default nt
identifier A.U2647 now becomes ‘.A.U.2647.’ when the option --idstr=long is specified or simply U2647 with --idstr=short; similarly, A.G2673 becomes ‘.A.G.2673.’ or G2673.

| # without the --idstr option (default) |
| 1 A.U2647 | A.G2673 | U-G | n/a | cWW cW -W |
| # with the --idstr=long option |
| 1 .A.U.2647 | .A.G.2673 | U-G | n/a | cWW cW -W |
| # with the --idstr=short option |
| 1 U2647 | 02673 | U-G | n/a | cWW cW -W |

The long id string consists of five components, which are put in strict order, as given below. A complete example – with model number 1, chain id ‘C’, residue name ‘A’ (for adenosine), sequence number 24, and insertion code ‘L’ – is ‘1.C.A.24.L’, the corresponding default form is ‘1:C.A24^L’, and the short form is A24.

| model-number.chain-id.residue-name.residue-number.insertion-code |

3.17.5 The --symmetry option

By default, DSSR only analyzes the first structure (model) of a given .pdb for .cif file. For x-ray crystal structures, however, the asymmetric unit could contain only a faction of the biological unit. As a simple example, the asymmetric unit of PDB id 4ms9 is single-stranded (4ms9.pdb), while the biological unit is a double helix (4ms9.pdb1).

Running DSSR on 4ms9.pdb, or 4ms9.pdb1 with its default settings, finds no base pairs. In such cases, one needs to use the --symmetry (short form: --symm) option as shown below for the desired result:

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

Please note the following points:

- DSSR just reads ATOM/HETATM records, as provided. It does not perform auto-expansion of an asymmetric unit into a biological unit based on crystallographic symmetric information in the PDB/mmCIF file.

- Users must supply the biological unit file (4ms9.pdb1) and the specify the --symm option for the desired result.

- In my understanding, the PDB MODEL/ENDMDL ensemble is ambiguous for its contents. It refers most commonly to the different conformations of an ensemble of
NMR structures, where each one is independent. It is also used for collecting together
symmetry-related components in x-ray crystal structures, as discussed here.

### 3.17.6 The --prefix option

By default, DSSR auxiliary output files are prefixed with `dssr`, as in `dssr-pairs.pdb`. The ‘fixed’ generic names are overwritten on repeated DSSR runs in a directory, thus reducing *pollution* of file names. 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.

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

### 3.17.7 The --auxfile option

Sometimes (e.g., in the analyses of molecular dynamics simulations of RNA structures), one may be just interested in the DSSR main output file (in text or JSON). This is where the `--auxfile=no` option comes into play: it instructs DSSR not to generate any auxiliary files as those noted on Page 12.

### 3.17.8 The --cleanup option

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

### 3.17.9 The --isolated-pair option

By default, DSSR treats isolated canonical pairs as a special case of stems (whether pseudoknotted or not) in delineating various ‘closed’ loops. As shown in Figure 2B, taking the isolated C2658–G2663 into account reveals the reported GUAA tetraloop [10] and a [5,4] internal loop. On the other hand, by excluding the isolated C–G pair via `--isolated-pair=not-in-loop`, both the tetraloop and the internal loop will be gone, leading to an enlarged hairpin loop of 17 nucleotides (C2652 to G2668) including C2658 and G2663. Overall, this option is more relevant for junction loops when comparing DSSR output with literature.
3.17.10 **The --torsion360 option**

By default, DSSR outputs (virtual) backbone torsion angles (see Section 3.2.13) in the range of $-180^\circ$ to $+180^\circ$. Following a user’s request, the --torsion360 option has been added to produce torsion angles in the range of $0^\circ$ to $+360^\circ$.

3.17.11 **The --select option**

This versatile option can be used to extract a type of structural component (including nucleotides, DNA, RNA, protein, metal etc) or any combination thereof, from an input .pdb or .cif file. As a simple example, to extract only the nucleotides in 1ehz (excluding water and metal ions), one can run:

```
1 x3dna-dssr --select -i=1ehz.pdb -o=1ehz-nts.pdb
2 x3dna-dssr --select=nt -i=1ehz.pdb -o=1ehz-nts.pdb # gives the same results as above
```

3.17.12 **The --dbn-break option**

By default, DSSR employs the symbol ‘&’ to separate multiple chains or chain breaks in dbn, for compatibility with VARNA. By using --dbn-break, one can choose any of the characters in the string “&,:|+”. For example, by running the following command on the Dickerson DNA dodecamer [16] structure 355d, one would get a whole-structure dbn composed of those from the two chains connected by ‘+’:

```
1 x3dna-dssr --dbn-break=+ -i=355d.pdb
2 >355d nts=24 [whole]
3 CGCGAATTCGCG+CGCGAATTCGCG
4 (((((((((())))))))))
```

With --dbn-break=no, no symbol will be used to separate different chains or intra-chain breaks: for 355d, the dbn would be (((((((((()))))))))).

3.17.13 **The --sugar-pucker option**

Following 3DNA, DSSR uses the Altona and Sundaralingam (1972) [21] definitions for the conformational analysis of sugar rings. By specifying --sugar-pucker=westhof83, the Westhof and Sundaralingam (1983) [36] formula would be employed instead, giving identical amplitude of pucker and the phase angle of pseudorotation as those from Curves+ [37].
3.17.14 The --raw-xyz option

By default, DSSR-derived auxiliary files (such as dssr-pairs.pdb, among several others) are automatically transformed for better visualization (see Figure 7) and easy comparison. At times, however, it may be desirable to keep the original atomic coordinates of selected residues, as shown in Figures 2-3 of the DSSR paper [1].

3.17.15 The --helical-axis option

The --helical-axis option introduces a new auxiliary file, dssr-helicalAxes.pdb by default, containing the least-squares fitted linear helical axes for all helices that DSSR detects. The output is in PDB format, with two end-points for each helix, that can be fed into PyMOL (or Jmol) for visualizing the helices.

Using the yeast tRNA\textsuperscript{Phe} (1ehz) as an example, the following command creates a file dssr-helicalAxes.pdb:

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

with the following content:

```
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
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
```

3.17.16 The --simple-junction option

The --simple-junction option produces an auxiliary file, dssr-simplifiedJcts.pdb by default, containing simplified representation of the junction loops that DSSR detects. The output is in PDB format, with C1' atomic coordinates for consecutive nucleotides in a junction, that can be fed into PyMOL for schematic visualization.

Using the yeast tRNA\textsuperscript{Phe} (1ehz) as an example, the following command creates a file dssr-simplifiedJcts.pdb:

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

with the following content:

```
REMARK DSSR: helix#1
ATOM 1 P1 G A 1 52.699 41.360 47.877 1.00 99.85 H1 P
ATOM 2 P2 C A 56 78.559 72.327 36.597 1.00 37.81 H1 P
ATOM 3 P1 A A 36 68.999 35.486 -4.554 1.00 81.67 H2 P
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
```
with the following content:

```
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
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
ENDMDL
```

3.17.17 The **--block-file** option

The **--block-file** option outputs rectangular blocks for bases (or base pairs with **--block-file=wc**, among other variations) in Raster3D .r3d format that can be fed into PyMOL, in combination with the cartoon representation of the backbone. See the DSSR-NAR paper section on the 3DNA Forum for examples.
3.17.18 **The --block-color option**

The **--block-color** option facilitates flexible color customizations of blocks/edges (e.g., the minor groove). As an example, **--block-color='A blue; T red'** would color base A blue, and T red. See my blog post titled “The DSSR **--block-color** option” for details.

3.17.19 **The --block-depth option**

The **--block-depth** option is closely related to **--block-file**, and it is used to set the thickness of the rectangular blocks. With **--block-depth=1.2**, the block becomes 1.2 Å thick, where the default value is 0.5 Å.

3.17.20 **The --hbfile-pymol option**

The **--hbfile-pymol** option can be used to draw a molecular structure in PyMOL, with DSSR-detected H-bond information. See Supplementary Figures 1,3,4,7 in the DSSR-NAR paper section on the 3DNA Forum for examples (e.g. Supplementary Figure 1 – four base triplets in yeast phenylalanine tRNA (1ehz)).

3.17.21 **The --hbfile-jmol option**

In parallel to **--hbfile-pymol**, the **--hbfile-jmol** option can be used to draw a molecular structure in Jmol, with DSSR-detected H-bond information.

4 **Pseudoknots detection and removal**

An RNA pseudoknot results from WC pairing between nts in a hairpin loop and those outside the enclosing stem. Pseudoknots are abundant in RNA structures in the PDB, and are known to play essential functional roles [38]. However, since base pairs are not fully nested, pseudoknotted structures also possess a challenge to many (e.g., dynamic programming based) RNA computational tools. DSSR aims to provide a pragmatic way for characterizing RNA pseudoknots to better understand their structural features and functional roles.

From early on, DSSR-derived RNA secondary structures in dbn had taken pseudoknots into consideration, as shown in the pseudoknot identified in tRNA^{Phe} 1ehz (Section 3.3.8).
Nevertheless, since RNA pseudoknot is a (relatively) complicated issue, I had originally planned to put off exploring the topic deeper until DSSR is well-established. So for DSSR releases prior to v1.1.3-2014jun18, the dbn output had been over-simplified to the first level, with matched []s only, even for RNA structures with higher-order pseudoknots.

### 4.1 Higher-order pseudoknots

In early May 2014, I noticed the Antczak *et al.* RNApdbee article [39], and I was delighted to see the first DSSR citation. Significantly, even before a paper on DSSR is published, the software has already be ranked in the top three (together with RNAView and MC-Annote) for the identification and classification of RNA base pairs. Nevertheless, the RNApdbee paper also remarks that “3DNA/DSSR improperly classifies base pairs (within residues in red) and the structure [1ddy] is recognized as the first-order pseudoknot”. This citation and a closely related question on ‘higher-order pseudoknots’ prompted me to further refine DSSR so that it can derive proper dbn for RNA structures with higher-order pseudoknots.

As of DSSR v1.1.3-2014jun18, the relevant output of running DSSR on PDB entry 1ddy is as below. Note that the whole 1ddy entry contains four RNA chains (A, C, E, and G), and DSSR can handle each properly.
4.2 Pseudoknot removal

A closely related issue is knot removal, a topic nicely summarized by Smit et al. [40] in their publication “From knotted to nested RNA structures: A variety of computational methods for pseudoknot removal”. The \texttt{--nested} (abbreviated to \texttt{--nest}) option can be used to remove pseudoknots to get a nested secondary structure in \texttt{dbn}.

Again, using PDB entry 1ddy as an example, the relevant output of running DSSR with option \texttt{--nested} is as follows:

```
This structure contains 2-order pseudoknot

o You have chosen to remove the pseudoknots with the '\texttt{--nested}' option so only a fully nested secondary structure representation remains.

Secondary structures in dot-bracket notation (\texttt{dbn}) as a whole and per chain

>1ddy nts=140 \{whole\}
GGACCGGGUGCGCAUAACCCACCACUGCGAGCAA&GGACCGGGUGCGCAUAACCCACCACUGCGAGCAA
\leftrightarrow GGACCGGGUGCGCAUAACCCACCACUGCGAGCAA&GGACCGGGUGCGCAUAACCCACCACUGCGAGCAA

\ldots (((..)))\ldots \ldots & \ldots (((..)))\ldots
\leftrightarrow \ldots \ldots & \ldots (((..)))\ldots

\ldots \ldots

>1ddy-A #1 nts=35 0.46(2.84) \{chain\} RNA
GGACCGGGUGCGCAUAACCCACCACUGCGAGCAA

\ldots (((..)))\ldots 

>1ddy-C #2 nts=35 0.66(2.88) \{chain\} RNA
GGACCGGGUGCGCAUAACCCACCACUGCGAGCAA

\ldots (((..)))\ldots

>1ddy-E #3 nts=35 0.72(2.88) \{chain\} RNA
GGACCGGGUGCGCAUAACCCACCACUGCGAGCAA

\ldots (((..)))\ldots

>1ddy-G #4 nts=35 0.54(2.87) \{chain\} RNA
GGACCGGGUGCGCAUAACCCACCACUGCGAGCAA

\ldots (((..)))\ldots
```

5 Resources

5.1 Web interface

A simple web interface to DSSR is available, which aims to showcase key features of the program by providing only its main output file. Moreover, the input files (\texttt{.pdb} or \texttt{.cif}) uploaded to the website must be less than five megabytes (5MB) in size. To take full advantage of DSSR, users need to install the program and run it locally.

For convenience, the link DSSR Web Interface is available at the centered footer on each page of the manual. With internet connection, simply clicking the link will lead you there.
5.2 DSSR-Jmol integration

Supported by Dr. Robert Hanson, DSSR has been integrated into Jmol, a popular molecular graphics program. A paper titled “DSSR-enhanced visualization of nucleic acid structures in Jmol” has been published in the 2017 web-server issue of NAR. The DSSR-Jmol integration [7] fills a gap in RNA structural bioinformatics, serving a huge user base of researchers, educators, and students alike. Its functionality can be accessed via either the Jmol application, or the JSmol-based website (see the screenshot in Figure 21). The http://jmol.x3dna.org website adheres to web standards and is fully functional in all modern browsers on various computer/operating systems (including handheld devices, such as tablets and smart phones). The web interface is simple and intuitive, and new users can get started easily. It is feature rich for practical applications; it has become my go-to place when I want to check a PDB structure. The website also allows power users to take full advantage of Jmol scripting via a command-line console window.

A dedicated section on 3DNA Forum, titled “DSSR-Jmol integration”, has been created with scripts and data files so others can reproduce results reported in the DSSR-Jmol paper [7]. Notably, the paper has been featured in the cover image of the 2017 web-server issue of NAR (see Figure 22). The “DSSR-Jmol integration” section on the 3DNA Forum also contains details on “how the cover image was created”.

5.3 DSSR-PyMOL integration

Thomas Holder, the PyMOL Principal Developer at Schrödinger, has developed a DSSR plugin for PyMOL (dssr_block). In contrast to the DSSR-Jmol integration which takes advantage of DSSR’s RNA structural analysis features, the dssr_block command focuses on the cartoon-block schematic representations. See my blogpost titled “DSSR base blocks in PyMOL, interactively” for more information.

5.4 Online support

The 3DNA homepage (x3dna.org, linked at the left header) contains DSSR-related news and information. Questions and suggestions are always welcome on the 3DNA Forum (forum.x3dna.org, linked at the centered header). The most up-to-date DSSR User Manual is available by clicking the right header.
Figure 21: Screenshot of the DSSR-Jmol integration web-interface, highlighting the two reverse Hoogsteen pairs (U8–A14 and 5MU54–1MA58) in the classic yeast phenylalanine tRNA (PDB entry: 1ehz), and the contextual menu for more options. To play, check the URL: http://jmol.x3dna.org/.
**Figure 22:** 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 id: 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 (a kissing-loop motif) at the top is highlighted in red (upper left corner). Structure of the *Thermus thermophilus* 30S ribosomal subunit in complex with antibiotics (PDB id: 1fjg) in step diagram. The 16S ribosomal RNA 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 phenylalanine tRNA (PDB id: 1ehz) in step diagram, with the three hairpin loops highlighted in red and the [2,1,5,0] four-way junction loop in blue (upper right corner). Structure of the Pistol self-cleaving ribozyme (PDB id: 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).

Details on how the cover image was created are available.
6 Frequently asked questions

6.1 What does DSSR stand for?

As of March 2015, DSSR stands for Dissecting the Spatial Structure of RNA. Previously, it means Defining the (Secondary) Structure of RNA. From early on, I knew DSSR does more than just defining RNA secondary structures as DSSP does for proteins, and that was why I put the word ‘secondary’ in parenthesis. As noted in my initial post “DSSR, what’s it and why bother?”, “the acronym could have two other possible interpretations, as would be obvious when the program gains a wider recognition.” Thus, ‘Dissecting the Spatial Structure of RNA’ can be taken as one of the two interpretations. The other meaning will reveal itself when DSSR is firmly established.

6.2 How does DSSR compare with other tools?

Dozens of 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 a general understanding. Without going into the gory details, however, such comparisons are mostly superficial (see my early paper on “Resolving the discrepancies among nucleic acid conformational analyses”).

DSSR is an integrated and automated computational tool designed from the bottom up to streamline the analysis and annotation of RNA structures. It possesses a combined set of functionalities well beyond the scope of any known software tools in the field. One does not need to go further than comparing DSSR results for the classic yeast tRNA\textsuperscript{Phe} (1ehz, see Section 3.3) with those from any other tools. Pay close attention to the fact that 1ehz contains 14 modified nucleotides. Among other features, yeast tRNA\textsuperscript{Phe} has four base triplets, two helices corresponding to the L-shaped tertiary structure, four stems matching the cloverleaf secondary structure, three hairpin loops and a [2,1,5,0] four-way junction loop.

Functionality is just one facet of a scientific program. For real-world applications, a software product must be solid in design and implementation. DSSR is efficient and robust due to extensive tests against all nucleic-acid-containing structures in the PDB and continued refinements based on user feedback. The program is small (∼1MB in size of distributed binary files), self-contained (without runtime dependencies on third-party
libraries), and possibly the simplest to set up. Related questions are always promptly answered on the 3DNA Forum. And, yes, it also has a decent user manual (the one you are reading right now).

Give DSSR a try, compare it with ‘similar’ tools in terms of usability, functionality and support, and see the differences! In the long run, only users can and will determine the fate of a product, be it software or otherwise. I will keep improving and expanding DSSR to maximize its usefulness to the community in the years to come.

6.3 How is DSSR related to 3DNA?

As stated explicitly in the Introduction (Section 1), DSSR is part of the 3DNA suite of software programs. It is a representative of what would become 3DNA version 3 (v3), consisting of a handful of programs that are mutually independent yet closely connected. SNAP is another such program, tailored for the analysis of structures of nucleic-acid and protein complexes. All 3DNA v3 programs would be prefixed with x3dna, as in x3dna-dssr and x3dna-snap, to avoid possible naming collisions and to make their context clear.

The 3DNA software package was initially developed around year 2000. Currently at v2.3, the 3DNA code base has largely unchanged from v1.5 corresponding to the 2003 NAR paper [8]. Over the years, 3DNA has been too widely used and integrated into other bioinformatics resources to allow for significant modifications without breaking existing user interfaces. When the 3DNA project got funded in late 2011, I decided to build a completely new code base to better take advantage of my extensive user-support experience, expanded domain knowledge, and refined programming skills. That’s how DSSR and SNAP came into being.

As a result, 3DNA v2.3 will be in maintenance mode: I will continually support this ‘classic’ version by quickly fixing any identified bugs and answering user questions, as always. By and large, though, no more significant new features are planned for v2.3. Where appropriate, related v2.3 functionality would be incorporated into DSSR and other v3 programs. For example, key features of frame.mol, rotate.mol, find_pair, and blocview from 3DNA v2.3 have already been distilled into DSSR, as of v1.5.5-2016may25.
6.4 How to cite DSSR?

The DSSR article [1], published in *Nucleic Acids Res.* in 2015, should be used as the primary citation:


Data files and scripts for reproducing all reported figures, tables, and sample DSSR output are available on the 3DNA Forum, in the DSSR-NAR paper section. Note that the 2015 NAR article [1] corresponds to DSSR v1.2.8-2015Jun15. Since the program is being continuously refined, the results from later DSSR releases may not match exactly those originally reported in the paper (including its supplementary data). The discrepancies, however, are few and nonessential, occurring only in edge cases.

If you have used the DSSR-Jmol integration [7], please also cite:


Data files and scripts for reproducing the reported results are available in the DSSR-Jmol integration section on the 3DNA Forum.

6.5 Does DSSR work for DNA?

Yes, DSSR works for DNA the same way it does for RNA.

6.6 Does DSSR detect RNA tertiary interactions?

Yes, to some extent at least. For example, coaxial stacking interactions, A-minor motifs and ribose zipapers are classified as tertiary interactions in the literature. There are certain blurry areas as whether to classify certain structural units (e.g., k-turns, kissing loops) as (super) secondary structural elements or as tertiary interactions.

Many RNA tertiary motifs have been reported in the literature, and more are surely to be discovered. At this stage, the goal is not to characterize RNA tertiary interactions comprehensively in DSSR. Nevertheless, more RNA tertiary motifs will gradually be included, as appropriate. With secondary structure elements already delineated, DSSR could arguably be expanded to categorize RNA tertiary interactions systemically.
6.7 Why is feature X not documented in the manual?

DSSR has many more features than those documented in this manual. Moreover, even documented features may have extra variations not mentioned here. This is on purpose, for the following reasons:

- DSSR already contains numerous documented features, probably far too more than a typical user would normally need. The manual is already near 100 pages, and I know that the acronym TLNR stands for ‘Too Long, No Read’. On the other hand, no matter how much effort I may spend on the documentation, a user with special needs can still easily find something missing. As a compromise, I strive for the accuracy of documented features in the manual, targeting the most common use cases.

- Some features are experimental and may change in later releases of DSSR. Documenting them prematurely is not only a waste of time, but also causes confusions when DSSR no longer works after an update.

- Not surprisingly, certain undocumented features are on specific research topics I would like to pursue further (including publishing in peer-reviewed journal articles), before making them openly available.

The 3DNA Forum is the place for each and every DSSR question you may still have after reading the manual. I monitor the Forum regularly, and always respond to questions posted there, quickly and concretely.

If a feature is not yet available in DSSR, you’re welcome make a request on the 3DNA Forum. I aim to give each and every feature request a careful consideration, but will only consider to implement the ones that I can fully understand. Additionally and importantly, the new features must make sense to me (I can be more easily convinced with specific examples), and should be of at least potential general interest. I’d like to make it clear that feature-rich is not a priority or major goal of DSSR; what I strive for is a coherent set of features that are robust and efficient, practical for real world applications, and easily adaptable to new environments.

Simply put, I’m committed to the accuracy of documented features, ready to answer any questions when asked. DSSR offers more than it claims, a characteristic that should be taken as a virtue instead of the other way around. If you’ve any DSSR-related questions, simply ask for help on the 3DNA Forum.
6.8 May I ask you questiones on DSSR via email?

In general, the 3DNA Forum is the way to go for DSSR-related questions. Over the years, I’ve received numerous emails asking for help on 3DNA/DSSR. As a result, I even created the following canned response in gmail:

“Thanks for your interest in using 3DNA. Please be aware that for the benefit of the 3DNA user community at large, I do not provide private email support; the 3DNA Forum (http://forum.x3dna.org/) has been created specifically for open discussions about all 3DNA-related issues. In other words, *any* 3DNA-associated questions are welcome and should be directed there. Specifically, please do *not* be shy in sharing openly and concretely difficult experiences you may have in installing or using the software.

By asking your questions on the public 3DNA Forum, you are not only benefiting yourself but also the user community. I monitor the Forum regularly and always respond to posts promptly. I look forward to ‘seeing’ you on the 3DNA Forum (http://forum.x3dna.org/).”

On the other hand, I always keep an open mind to any potential collaboration on a research topic of mutual interest. DSSR has many more novel features than I could possibly pursue in depth myself. Explored further, some of the (undocumented) features can easily lead to scientific publications. Over the years, I’ve been lucky to collaborate with quite a few fellow scientists where we communicated mostly via email (and Skype). If you find your research project could better benefit from my expertise in RNA structures and in 3DNA/DSSR/SNAP, let’s chat via email.

6.9 Can I run DSSR in parallel?

DSSR is written in ANSI C, a well-established small language so I can easily fix any identified bugs. The program does not take modern hardware multicore or multiprocessor into consideration. My primary goal so far has been to make DSSR robust and efficient. In my experience, speed has never been a real bottleneck: I regularly test DSSR against the near 10K nucleic-acid-containing structures in the PDB, without problem.

On the other hand, you can run multiple instances of DSSR to better take advantage of what your hardware has to offer. I’ve recently played around with GNU Parallel with
several times of speedup for analyzing the \( \sim 10K \) PDB RNA/DNA structures. In this context, one may want to use the \--auxfile=no\ option (see Page 73) first introduced in DSSR v1.7.6-2018mar22.

7 Citations in peer-reviewed journal articles

Since its initial release in early 2013, DSSR has been cited a few times in peer-reviewed journal articles. Prior to the 2015 DSSR Nucleic Acids Res. publication [1], the citations have been made either to the GpU dinucleotide platform article [11] or one of the two 3DNA papers [8, 9]. The following list of articles is based on my own reading, and is very likely to be incomplete. In each entry, the term DSSR is mentioned explicitly in either the main article or the corresponding online method/supplementary section.

The citations to DSSR can only increase, and the number is presumably to become significant, especially now that the DSSR paper [1] has come out. As a result, the list is limited to the first 20 citations I know of, which was reached on February 22, 2017.


### 8 Integrations into other bioinformatics resources

In addition to analyzing RNA structures in research publications (see Section 7), DSSR has already been integrated into many bioinformatics resources. This comes as no surprise: by design, DSSR serves perfectly as a component in other pipelines that need 3D RNA/DNA structural information. In addition to the DSSR-Jmol (Section 5.2) and DSSR-PyMOL (Section 5.3) integrations, here is a list of other bioinformatics tools where DSSR has played a role.

- **WebSTAR3D**, a web server for RNA 3D structure alignment
- **Universe of RNA Structures (URS)**, a web-interface to URS database (URSDB) that includes all RNA-containing PDB entries
- **RNApdbee**, a webserver to derive RNA secondary structures from PDB files
- **RNA-ff1 within Xplor-NIH**, a new force field for calculating solution NMR structures of RNA
- **A new RNA force field** from D. E. Shaw Research
- **RiboSketch**, a visualization tool of multi-stranded RNA/DNA secondary structures

The above list is by no means complete: the resources are just the ones I came to know. I’ve no doubt that the list will grow substantially in the near future. DSSR is a gem still waiting to be fully appreciated. If you’d like to have your tool listed here, please drop me a message.
9 Revision history

This manual is being continuously refined and expanded, both independently and along with the development of the DSSR program per se. As always, user feedback to improve this document is greatly appreciated, and will be properly acknowledged.

1. 2014-03-09: initial release – kick to ball rolling!

2. 2014-03-19: extensive revisions and significant improvements based on feedback from Dr. Wilma Olson; regenerated Jmol images with better quality.

3. 2014-03-24: added the overlooked subsection ‘Orientation of helices/stems’ (on page 40, proofread by Dr. Wilma Olson); fixed a few typos and inconsistencies.

4. 2014-04-09: denoted unnamed bps as -- for easy parsing (thanks to feedback from Dr. Robert Hanson); added helical radius info for helices/stems, and made the helical rise parameter explicit (thanks to Dr. Wilma Olson); changed pars to -pars in the output file for consistency; upgraded DSSR to v1.1.0-2014apr09 due to format changes.

5. 2014-04-11: added the option --get-hbond to find and output all H-bonds in a structure (thanks to Dr. Wilma Olson for proofreading the section); renamed file dssr-torsions.dat to dssr-torsions.txt.

6. 2014-04-19: refinements based on feedback from Dr. Pascal Auffinger; added the option --torsion360 following Cathy Lawson’s suggestion.

7. 2014-09-23: converted the manual from Word to \LaTeX. Now all the hyperlinks and cross references are active, and the listings are auto-synced with the latest release of DSSR via a Ruby script.

8. 2015-02-03: revised the sections on base pair classifications, and installation section, and fixed several typos in the manual.

9. 2015-03-28: updated to v1.2.6; added an FAQ entry in Section 6.1 to explain “What does DSSR stand for?” – now it ‘officially’ means: Dissecting the Spatial Structure of RNA.
10. 2015-04-21: added an FAQ entry in Section 6.2 to address the question of “How does DSSR compare with other tools?”; added the section “Citations in peer-reviewed journal articles”.

11. 2015-06-09: documented options `--prefix` and `--cleanup`, and expanded the section on ‘Sugar conformational parameters’ (Section 3.2.13).

12. 2015-06-15: added “Additional options” (Section 3.17.8) to document auxiliary options; in syn with DSSR v1.2.8-2015jun15 release.

13. 2015-07-25: updated to v1.2.9-2015jul25. Added to the 3DNA Forum a new section DSSR-NAR paper with details for reproducing the table and figures (including supplementary ones) reported in the DSSR paper [1]; augmented the manual with more previously undocumented options.

14. 2015-08-27: added documentation for the `--json` option to be in sync with the v1.3.0-2015aug27 release.

15. 2015-08-29: revised tag names for the `--json` output based on feedback from Dr. Wilma Olson, and updated examples.

16. 2015-09-03: added documentation for the four simple bp parameters and those for non-pairing interactions. Incidentally, the PDF manual now has 70 pages.

17. 2015-09-18: revised JSON output for better DSSR-Jmol integration (thanks to Dr. Robert Hanson). Specifically, a “metadata” property is introduced to collect miscellaneous information, thus simplifying the top-level name space. In sync with the v1.3.6-2015sep18 release.

18. 2015-10-02: added documentation for the `--symmetry` and `--cartoon-block` options, with examples.

19. 2015-10-10: added documentation for the `--nmr` option and the section on “Structural features per nucleotide”. In sync with the v1.4.0-2015oct10 release.

20. 2015-11-18: in sync with the v1.4.4-2015nov18 release, with refinements on the identification of base-pairs and multiplets. Now DSSR outputs base-capping interactions by default.

22. 2016-04-02: documented the `--block-color` option, and the DSSR-PyMOL integration.

23. 2016-04-11: added a brief description of the base-phosphorus virtual torsions, which became available from DSSR by default as of v1.5.3-2016apr11.

24. 2016-05-25: documented the `--view` option. Significantly reorganized options via command-line (`-h`) and in this manual.

25. 2016-07-09: documented the `summary` line for each loop.

26. 2016-08-08: documented the `--pair-list` option (as of v1.6.0-2016aug06).

27. 2016-08-22: added documentation on the i-motifs (as of v1.6.1-2016aug22), and a screenshot of the DSSR-Jmol integration web-interface.

28. 2016-10-19: added documentation on splayed-apart conformations (as of v1.6.3-2016oct19).

29. 2017-02-20: documented the eXtended type X A-minor motifs to cover cases other than the conventional types I and II, in sync with the v1.6.6-2017feb20 release. Revisited header/footer links to make them more explicit.

30. 2017-07-02: updated the manual with the NAR’17 web-server paper on DSSR-Jmol integration; Added the cover image of the NAR’17 web-server issue, featuring the DSSR-Jmol integration.


32. 2017-11-20: updated to v1.7.2-2017nov20. Fixed a bug with abasic sites (as in PDB entry 4ifd), and a bug in listing of modified nucleotides (as in PDB entry 2c4z in its biological assembly).

33. 2018-03-15: documented the `--blockview` option.
34. 2018-03-22: updated to v1.7.6-2018mar22. Documented the --auxfile option, added three more FAQ entries and the section on DSSR integrations.

35. 2018-09-01: updated to v1.7.8-2018sep01. Classified G-tetrads by different types of non-planarity; sped up the analysis of large ensembles (--nmr) as for the trajectories from MD simulations; introduced the “Linker” G+A base-pair name/type; revised the algorithm for H-bonding identification, plus numerous other minor code refactoring and refinements.

36. 2018-09-06: updated to v1.7.9-2018sep06. Sped up further the analysis of NMR ensembles or MD trajectories; revised algorithms for identifying base pairs and multiplets in special cases; improved the mmCIF parser; plus minor code/manual refinements.

37. 2018-09-18: v1.8.0-2018sep18 released. Significantly improved the characterization of G-quadruplexes, (1) revised the algorithm for the calculation of G-tetrad step parameters (twist/rise); (2) new features for the assignment of groove widths (medium, narrow, or wide), classification of stacking interactions based on the two faces of G-tetrads, and categorization of higher-order associations (coaxial stacking). Other refinements related to the identification of base-pairs and multiplets.


39. 2018-10-20: v1.8.2-2018oct20 released. Identified non-stem loops, such as V-loops [31], in G4 structures.

40. 2018-10-29: v1.8.3-2018oct29 released. Added the --pair-only option and a descriptive note on G4 structures, among other refinements.

10 Acknowledgements

The development of DSSR has been made possible by the NIH grant R01GM096889. I’d like to thank Drs. Wilma Olson and Harmen Bussemaker for many helpful discussions during the course of the work. Drs. Wilma Olson and Andrew Colasanti helped
clarify the notes in DSSR output files. I acknowledge the following users – jyvdf3asdg2, kailsen, MarcParisien, jctoledo, Auffinger, febos, acolasanti, hansonr, cllawson, Sylverlin, cigdem, lvelve0901, jms89, meier74, chemikeris, Marcel Heinz, rcsb_pdb, and Bernhard10 (3DNA Forum ids) – for employing early versions of DSSR, and/or for providing feedback which helped improve the software.

References


