

# PseudoViewer: web application and web service for visualizing RNA pseudoknots and secondary structures

Yanga Byun and Kyungsook Han\*

School of Computer Science and Engineering, Inha University, Incheon 402-751, Korea

Received February 14, 2006; Revised March 24, 2006; Accepted March 24, 2006

## ABSTRACT

**Visualizing RNA secondary structures and pseudoknot structures is essential to bioinformatics systems that deal with RNA structures. However, many bioinformatics systems use heterogeneous data structures and incompatible software components, so integration of software components (including a visualization component) into a system can be hindered by incompatibilities between the components of the system. This paper presents an XML web service and web application program for visualizing RNA secondary structures with pseudoknots. Experimental results show that the PseudoViewer web service and web application are useful for resolving many problems with incompatible software components as well as for visualizing large-scale RNA secondary structures with pseudoknots of any type. The web service and web application are available at <http://pseudoviewer.inha.ac.kr/>.**

## INTRODUCTION

A large-scale bioinformatics system often consists of several application programs dealing with a large volume of data. For example, a data analysis program generates new data that may be modeled and integrated by other programs. However, the programs may not be interoperable due to the differences in data formats or running platforms. As a result, developing a bioinformatics system with a few application programs requires extra work to make the components compatible. In fact the difficulty in bioinformatics study comes more from the heterogeneity of the programs and data than from the quantity of the data.

Web service resolves some of these problems by exchanging messages between different applications developed by various programming languages. The basic process

of web service is exchanging Simple Object Access Protocol (SOAP) messages described by XML (eXtensible Markup Language). When the server of web services receives the SOAP request with the parameters for calling the method of web service, the server returns the SOAP message in response to the method. An application program can use web services developed in different languages, whose results in turn can be a request message for another web service.

Recently various bioinformatics systems supporting web services have been developed (1). European Bioinformatics Institute provides several web services such as Dbfetch for biological database, ClustalW for multiple alignments of DNA and protein sequences, Fasta for nucleotide comparison (2). Databases of KEGG (3) and DDBJ (4) give web service access. A tool named Taverna was developed for the bioinformatics workflows with several relevant biological web services (5).

An RNA pseudoknot is a tertiary structure element formed when bases of a single-stranded loop pair with the bases outside the loop. Several computer programs are available for drawing RNA secondary structures (6–9), but none of these can draw RNA pseudoknots or show the topological relation of the RNA stems. We developed previously PseudoViewer1 (10) and PseudoViewer2 (11) to automatically visualize RNA pseudoknot structures as planar graphs. Since the first release of PseudoViewer, it has been widely used for drawing RNA secondary structures with pseudoknots. However, it produces an unnecessary number of support lines in order to draw pseudoknots of complex type as planar graphs, and ignores tertiary interactions other than pseudoknots in the input structure data. More importantly, the previous versions of PseudoViewer are written in Microsoft C# and are executable on Windows systems only. To remove these limitations and to handle data in various formats, we developed a web service and web application of PseudoViewer as well as a new visualization algorithm. Users can visualize large-scale RNA secondary structures with pseudoknots of any type from any web browser and any operating system.

\*To whom correspondence should be addressed. Tel: +82 32 860 7388; Fax: +82 32 863 4386; Email: [khan@inha.ac.kr](mailto:khan@inha.ac.kr)

## INPUT AND OUTPUT

PseudoViewer takes as input the structure data in the bracket view, in which a base pair is represented in a pair of matching parentheses (either round parentheses ‘()’ or square brackets ‘[]’). It also allows tertiary interactions other than pseudoknots instead of ignoring them. They are represented in a pair of

matching braces (‘{’) in the structure data, and are displayed in dotted lines in the final drawing [e.g. see the pseudoknot structure of the *Escherichia coli* alpha operon mRNA (12) in the example section of the PseudoViewer web page]. The bracket view describes pseudoknots and secondary structures in one of the following styles.

### (i) Bracket view I

```
# <RNA name>           // optional; this line may be omitted.
<starting position> // optional; starting position is 1 by default.
<base sequence>
<matching parentheses and brackets>

# TYMV
144
UUAGCUCGCCAGUUAGCGAGGUCUGUCCCCACACGACAGAUAAUCGGGUG
CAACUCCCGCCCCUUUCCGAGGGUCAUCGGAACCA
....((((((.....))))).((((((.....))))))....(((...
.....))))((...[[[[[[]]]...]]]])...

```

### (ii) Bracket view II

```
# <RNA name>           // optional; this line may be omitted.
<starting base number> // optional; the starting number is 1 by default.
<base sequence>       // base sequences alternates with
<matching parentheses and brackets> // structure data

# TYMV
144
UUAGCUCGCCAGUUAGCGAGGUCUGUCCCCACACGACAGAUAAUCGGGUG
....((((((.....))))).((((((.....))))))....(((...
CAACUCCCGCCCCUUUCCGAGGGUCAUCGGAACCA
.....))))((...[[[[[[]]]...]]]])...

```

When part of the sequence and its structure data is missing, the user can specify them in bracket view II as follows and PseudoViewer visualizes the structure with missing part marked.

```
# <RNA part name >     // optional
<starting base number> // optional
<base sequence>       // base sequences alternates with
<matching parentheses and brackets> // structure data

# <RNA part name>     // optional
<starting base number> // optional
/<base sequence>      // base sequences alternates with
/<matching parentheses and brackets> // structure data

# PKB128 (Berne virus)
1045
UUUAAACUGUUGAGAGGUGCCUGGAGCGCCUGCAGGCAUCUCUGUU
.....((((((.....[[[[[[]]]))))))))....

1150
/UAAUAUGCAGGCA
/.....]]]]].

```

### (iii) Bracket view III

```
$ <starting base number> <base sequence> // sequences alternate with
% <starting base number> <matching parentheses & brackets> // structure.

$ 1 UUAGCUCGCCAGUUAGCGAGGUCUGUCCCCACACGACAGAUAAUCGGGUG
% 1 ....((((((.....))))).((((((.....))))))....(((...
$ 51 CAACUCCCGCCCCUUUCCGAGGGUCAUCGGAACCA
% 51 .....))))((...[[[[[[]]]...]]]])...

```

If the bracket format in input data is not correct, the previous version of PseudoViewer could not visualize the structure (e.g. square brackets are used before round parentheses in the bracket view of an H-type pseudoknot). However, PseudoViewer can now read the structure data and gives a correct bracket view of the structure. PseudoViewer also extends the type of the input format. If a user knows the stem positions only and does not know how to represent the structure in the bracket view, he can simply provide the stem positions in the following format, which is also used in PseudoBase (13). Given stem positions, PseudoViewer automatically generates the structure data in bracket view and visualize the structure.

## (iv) pairing format

```
# <RNA name>           // optional;
<starting position> // optional; starting position is 1 by default.
<base sequence>
```

Positions paired:

```
<Starting position of the stem opening part>-<ending position of the stem opening part>;
<starting position of the stem-closing part>-<ending position of the stem closing part>
```

```
# Hepatitis delta virus (pseudoknot only)
```

```
688
```

```
GGCCGGCAUGGUCCCAGCCUCCUCGCGUGGGCGCCGGCUGGGCA
ACAUUCCGAGGGGACCGUCCCCUCGUAUUGGCGAAUGGGACC
```

Positions paired:

```
688-694; 718-724
```

```
697-703; 766-772
```

```
704-706; 715-717
```

```
708-709; 725-726
```

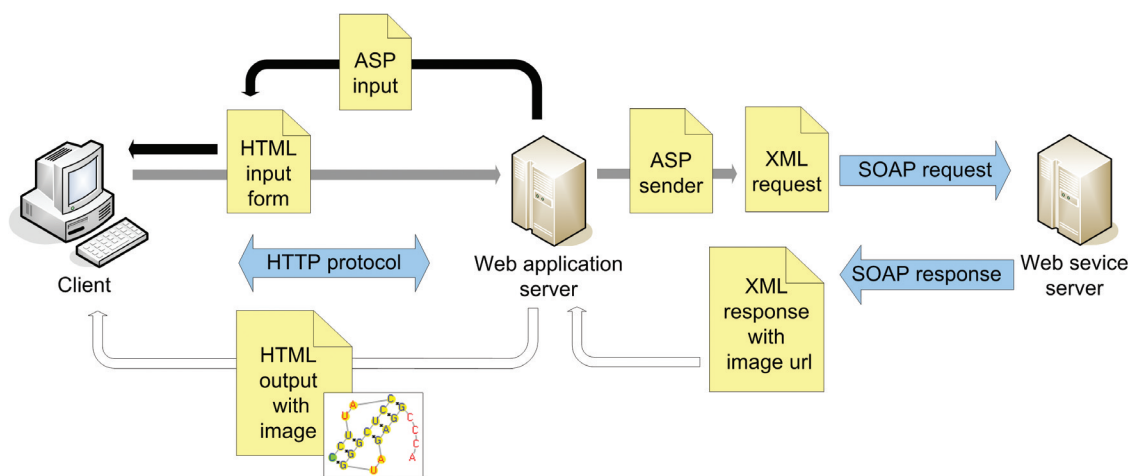
```
731-734; 757-760
```

```
735-743; 747-755
```

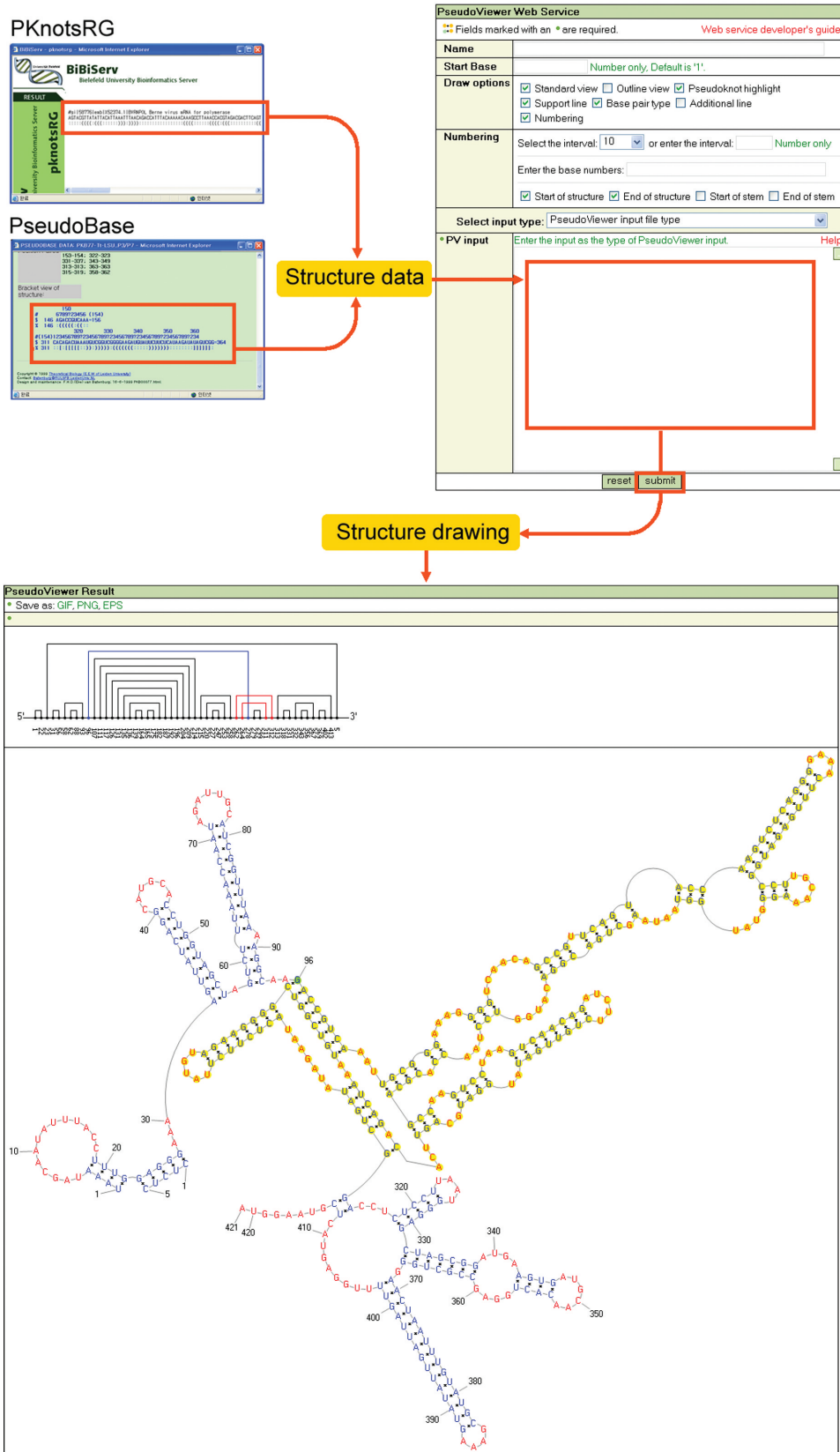
The output of the PseudoViewer web application is structure drawing and the topological relation of the stems. The structure drawing can be saved in a file either in 'gif', 'png', 'eps' or 'svg' format. The topological relation of the stems, such as adjacency, crossing, and inclusion of the structure is very useful for analyzing and predicting RNA structures and classifying RNA pseudoknots.

The output of the PseudoViewer web service is of four kinds as requested by clients: (i) URL of structure drawing, (ii) raw data of the structure drawing, (iii) structure drawing in the eps

format, and (iv) 2D positions of the bases in the structure drawing. In case the client requests the URL to be returned for the structure drawing, the server saves the structure image on the server and notifies the user of the URL address. The raw data output is for direct display or for saving in an image file. The 2D positions of the bases can be used for editing or for generating a new image by rendering the positions. Since the number of the structures included in the request message is not limited, many structure drawings can be obtained at one try.



**Figure 1.** Message transfer paths in web application. Messages in black paths are transferred first, messages in gray paths are transferred next, and messages in white paths are transferred last. For the client request, the web application server generates an HTML page for the input form by ASP. The web application server transforms the input to the XML SOAP request message, which is sent to the web service server. The web application server receives the XML SOAP response message with URL of the drawing image and delivers it to the client via HTML document.



**Figure 2.** As input PseudoViewer can take the structure data in one of the three bracket views or in the pairing format. The structure data in the pairing format provided by PseudoBase (13) as well as that in the PseudoViewer format provided by pknottsRG (14) can be directly visualized by PseudoViewer just by copy and paste operation in the PseudoViewer web application. PseudoViewer visualizes the topological relation of stems and entire structure drawing, which can be saved in a file in the 'png', 'gif', 'eps' or 'svg' format. The structure shown in this example is the *Tetrahymena* Group I Intron (15).

```

<WSPVRun xmlns="http://wilab.inha.ac.kr/WSPseudoViewer/">
  <WSPVRequest>
    <Option Scale="1" Output_name="E. coli mRNA" Output_type="url png" />
    <WSPVIn_data>
      <Sequence>
        <Sequence_data Name="E. coli mRNA" Start_Base="144">
          UGUGCGUUUCCAUUUGAGUAUCCUGAAAACGGGCUUUUCAGCAUGGAACGUACAUAUUAUAGUAGGAG
          UGCAUAGUGGCCCGUAUAGCAGGCAUUAACAUCUCCUGA
        </Sequence_data>
      </Sequence>
      <Bracket_view>
        ((((((.((((.....[[[[.....[[[.....{{{(.)})))))))))).....
        .....]]]]].....]]]]].....}}})
      </Bracket_view>
    </WSPVIn_data>
  </WSPVRequest>
</WSPVRun>

<WSPVRunResponse xmlns="http://wilab.inha.ac.kr/WSPseudoViewer/">
  <WSPVResponse>
    <WSPVOutput>
      <Name>E. coli mRNA</Name>
      <Images>
        <WSPVOut_URL>
          http://165.246.44.42/PseudoViewer/PVWebService/WSPVOutput/Ecoli.png
        </WSPVOut_URL>
      </Images>
    </WSPVOutput>
  </WSPVResponse>
</WSPVRunResponse>

```

**Figure 3.** Example of the request and response XML messages. It requests URL of the drawing in the 'png' format. Input data includes the secondary structure in the bracket view. The web service returns the URL of the structure drawing as response.

```

URL wsdl =
  new WSDL("http://165.246.44.42/WSPseudoViewer/WSPseudoViewer.asmx?WSDL");
//All classes can be generated by WSDL2Java in Apache axis,
WSPVNumbering numbering=new WSPVNumbering();
  numbering.setBase_numbers("1");
  numbering.setInterval(10);
  numbering.setNumbering_option("default");
WSPVOption option=new WSPVOption();
  option.setOutput_type("url png"); //request the url of the result image
  option.setOutput_name("test");
  option.setDrawing_option("default");
  option.setNumbering(numbering);
WSPVInFileData pvinput=new WSPVInFileData();
  pvinput.setPV_file(/* string data read from PseudoViewer input file */);
  pvinput.setOption(option); //add option
WSPVRequest pvrq=new WSPVRequest();
  pvrq.setOption(option); //add option
  pvrq.setStructures(new WSPVInput[]{pvinput}); //add input data
//make call for 'WSPVRunRpc' method
Call call= (Call)ser.createCall(new QName("WSPseudoViewerSoap"), "WSPVRunRpc");
//register the type of the inheritance classes.
WSPVRegisterTypeMapping.registerTypeMapping(call);
call.setReturnType(new
  QName("http://wilab.inha.ac.kr/WSPseudoViewer/", "WSPVResponse"),
      WSPVResponse.class);
//invoke method and get response
WSPVResponse str=(WSPVResponse)call.invoke(new Object[] {pvrq});
System.out.println(((WSPVOutURL)str.getPVResult()[0]).getImages()[0].
  getImageURL());

```

**Figure 4.** Example of Java client for the PseudoViewer web service. Its request message includes data from PseudoViewer input file, and the URL of the structure drawing is returned.



## WEB APPLICATION

Web application helps the user use the web service easily with any web browser (Figure 1). It interacts with the user via HTML pages and exchanges the XML based SOAP messages with the web service server by Active Server Page (ASP). An ASP document generates an XML document and this XML document is shown as an HTML page. The PseudoViewer server supports many features that enhance the utility and drawing quality of the software.

### Input and output

There are three ways to input data: (i) entering sequence and structure data separately, (ii) entering structure data in one of the PseudoViewer formats described earlier, and (iii) loading a file with structure data. The right part of Figure 2 shows an input form of the web application for entering the structure name and the start base. The numbering options are for marking base numbers in the structure drawing. The user can select the interval of base numbers or enter specific base numbers. Since PseudoViewer supports many kinds of bracket views, other structure data (such as tertiary interactions other than pseudoknots) in one of these bracket views can also be visualized by our system.

As output, the PseudoViewer web application produces not only structure drawing but also the topological relation of the stems. The topological relation of the stems is not the 3D orientation of stems but indicates the adjacency, crossing or inclusion of the stems (for an example of the topological relation of stems, see the upper pane of the structure drawing window in Figure 2). The topological relation identified from the structure data by PseudoViewer is very useful for analyzing and predicting RNA structures and classifying RNA pseudoknots. The structure drawing can be saved in a file either in the 'gif', 'png', 'eps' or 'svg' format.

### Example of using web application

Figure 2 shows a simple example of using the web application with the structure data from other web resources. The structure data in the pairing format provided by PseudoBase (13) as well as that in the PseudoViewer format provided by pknotsRG (14) can be directly visualized by PseudoViewer just by copy and paste operation in the PseudoViewer web application.

## WEB SERVICE

The PseudoViewer web service can be integrated into other systems to visualize RNA structures. Since the PseudoViewer web service runs by exchanging XML based SOAP messages, a system developed in any programming language can use the service. Sample clients written in C# and Java can be downloaded from the download page at <http://pseudoviewer.inha.ac.kr>. Visualizing the RNA structure with 4000 bases by the PseudoViewer web service took about 20 s on Pentium IV 2.4 GHz CPU, 516 MB RAM, Windows 2000 server system.

### Request and response messages

The request message includes the RNA sequence and the structure data either in the bracket view or pairing format.

The drawing options used in the web application are optional; if drawing options are not specified in the request message, structure drawing is returned as default type. The output of four kinds can be produced: URL of structure drawing, raw data of the structure drawing, structure drawing in the eps format, and 2D positions of the bases in the structure drawing. In case the URL of structure drawing is requested by the user, the server saves the structure image on the server and notifies the user of the URL address. The raw data output is for direct display or saving in an image file. The 2D positions of the bases can be used for editing or for generating a new image by rendering the positions. Since the number of the structures included in the request message is not limited, many structure drawings can be obtained at one try. Figure 3 shows a simple example of request and response messages. The details of the messages are in the Web Service Description Language (WSDL) page at <http://pseudoviewer.inha.ac.kr/WSPseudoViewer/WSPseudoViewer.asmx?WSDL>.

### Example of Java client

The web service for clients in Java requires the libraries of Apache AXIS (<http://ws.apache.org/axis/>), JavaBeans Activation Framework (<http://java.sun.com/products/javabeans/>) and JavaMail (<http://java.sun.com/products/javamail/>). WSDLtoJava, supported by Apache AXIS, makes Java classes from WSDL. Even if developers do not know the exact structure of WSDL, they can make a client program with the classes. The PseudoViewer web service supports three methods of rpc/encoded type for Java clients: (1) 'WSPVRUNRpc' method for requesting the structure data and for returning the structure image, (2) 'WSPVServerTestRpc' method for returning 'Server ok' message when the server runs well, and (3) 'WSPVServerRequestTestRpc' method for checking the request structure. The 'WSPVRUNRpc' method is the main method, and the other two methods are for checking programming code status. Figure 4 shows a request message of a Java client.

## ACKNOWLEDGEMENTS

This work was supported by the Korea Science and Engineering Foundation (KOSEF) under grant R01-2003-000-10461-0. Funding to pay the Open Access publication charges for this article was provided by KOSEF.

*Conflict of interest statement.* None declared.

## REFERENCES

- Stein, L. (2002) Creating a bioinformatics nation. *Nature*, **417**, 119–120.
- Pillai, S., Silventoinen, V., Kallio, K., Senger, M., Sobhany, S., Tate, J., Velankar, S., Golovin, A., Henrick, K., Rice, P. *et al.* (2005) SOAP-based services provided by the European Bioinformatics Institute. *Nucleic Acids Res.*, **33**, 25–28.
- Kawashima, S., Katayama, T., Sato, Y. and Kanehisa, M. (2003) KEGG API: A Web Service Using SOAP/WSDL to Access the KEGG System. *Genome Inform.*, **14**, 673–674.
- Miyazaki, S., Sugawara, H., Gojobori, T. and Tateno, Y. (2003) DNA Data Bank of Japan (DDBJ) in XML. *Nucleic Acids Res.*, **31**, 13–16.
- Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A. *et al.* (2004) Taverna: a

- tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, **20**, 3045–3054.
6. Andronescu, M., Fejes, A.P., Hutter, F., Hoos, H.H. and Condon, A. (2004) A new algorithm for RNA secondary structure design. *J. Mol. Biol.*, **336**, 607–624.
  7. De Rijk, P., Wuyts, J. and De Wachter, R. (2003) RnaViz 2: an improved representation of RNA secondary structure. *Bioinformatics*, **19**, 299–300.
  8. Han, K., Kim, D. and Kim, H.-J. (1999) A vector-based method for drawing RNA secondary structure. *Bioinformatics*, **15**, 286–297.
  9. Matzura, O. and Wennborg, A. (1996) RNAdraw: an integrated program for RNA secondary structure calculation and analysis under 32-bit Microsoft Windows. *Comput. Appl. Biosci.*, **12**, 247–249.
  10. Han, K., Lee, Y. and Kim, W. (2002) PseudoViewer: automatic visualization of RNA pseudoknots. *Bioinformatics*, **18**, S321–S328.
  11. Han, K. and Byun, Y. (2003) PseudoViewer2: Visualization of RNA pseudoknots of any type. *Nucleic Acids Res.*, **31**, 3432–3440.
  12. Gluick, T.C., Gerstner, R.B. and Draper, D.E. (1997) Effects of  $Mg^{2+}$ ,  $K^+$ , and  $H^+$  on an equilibrium between alternative conformations of an RNA pseudoknot. *J. Mol. Biol.*, **270**, 451–463.
  13. van Batenburg, F.H.D., Gulyaev, A.P. and Pleij, C.W.A. (2001) PseudoBase: structural information on RNA pseudoknots. *Nucleic Acids Res.*, **29**, 194–195.
  14. Reeder, J. and Giegerich, R. (2004) Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics*, **5**, 104–115.
  15. Karbstein, K., Tang, K.H. and Herschlag, D. (2004) A base triple in the *Tetrahymena* group I core affects the reaction equilibrium via a threshold effect. *RNA*, **10**, 1730–1739.