# Context Fold 1.00

***Context Fold*** is an RNA secondary structure prediction software package. It contains algorithms for predicting structures of RNA strings which allow the usage of several scoring models, as well as algorithms for setting scoring model parameters by training on datasets of RNA strings with known structures. The main notions are described in the paper:

Shay Zakov, Yoav Goldberg, Michael Elhadad, and Michal Ziv-Ukelson. Rich Parameterization Improves RNA Structure Prediction. Research in Computational Molecular Biology (RECOMB 2011) 546-562.

This document includes installation and basic usage instructions of the software. Check our website for updates:

http://www.cs.bgu.ac.il/~negevcb/contextfold/

For bug reporting and suggestions, please write to:

zakovs@cs.bgu.ac.il, yoavg@cs.bgu.ac.il

## Installation

1. Download the latest version of Context Fold from here.
2. Use any zip extraction software for extracting the downloaded zip file to your preferred directory (e.g. "C:/programs"). The extraction should create a new subdirectory by the name of ContextFold.
3. That's it! The new ContextFold directory contains the binaries, source code, documentation, and pre-trained scoring models of the Context Fold package. Make sure that you have Java 6 or higher installed on your computer (can be downloaded from here).

## Quick start

The main access point to the functionalities of the Context Fold software is the contextFold.app java package. This package contains the following classes:

- Predict: runs the folding prediction algorithm.
- Train: trains scoring model parameters with respect to a given training dataset.
- ValidatePrediction: measures prediction accuracy, for the case of predicting structures of RNAs with known structures.
- PrintModelFeatures: prints the list of feature names and corresponding weights in a given scoring model.

All of the above classes contain a 'main' method, thus can be executed with the 'java' command. For more details, run the programs with the 'man' command-line argument or refer to the javadocs. Below are several basic usage examples. Set the current directory to the main Context Fold installation directory (e.g. C:/programs/ContextFold), and run the programs according to the following examples:

Secondary structure prediction:

- Predicting the secondary structure of the command-line given RNA string AAGGCCUUGGGGAAGGCCUU using the (default) supplied StHighCoHigh trained model, and printing the output to the screen:

  ```
  java -cp bin contextFold.app.Predict in:AAGGCCUUGGGGAAGGCCUU
  ```

- Predicting the secondary structures of all RNA strings in the file C:/RNAdata/sequences.txt using the (default) supplied StHighCoHigh trained model, and saving the output to the file C:/RNAdata/sequences.txt.pred:

  ```
  java -cp bin contextFold.app.Predict in:C:/RNAdata/sequences.txt
  ```

  The input file may contain multiple *RNA items*. Each item starts with 0 or more comment lines, where a comment line must begin with a '>' or a '#' character. Then follows exactly one line of an RNA sequence (over the {A, C, G, U} alphabet), and 0 or more structure lines (over the {(, ), .} alphabet). Each structure has to be of the same length as the corresponding RNA sequence (input structures are used for training and are ignored when predicting, yet can be used for measuring prediction accuracy). The S-Full.txt file under the 'trained' directory is an example for a possible input file. The output file will be of the same format, where each item will include one additional structure that was obtained from the prediction algorithm.

- Predicting the secondary structures of all RNA strings in the file C:/RNAdata/sequences.txt using the supplied StMedCoHigh trained model, and saving the output to the file C:/RNAdata/prediction (see above for input file formatting):

  ```
  java -cp bin contextFold.app.Predict in:C:/RNAdata/sequences.txt
  model:trained/StMedCoHigh.model out:C:/RNAdata/prediction
  ```

Training scoring models:

- Training the (default) StHighCoHigh model on the dataset in the file C:/RNAdata/trainData.txt (the file must contain the 'true' structures of the listed sequences, see above for input file formatting), generating in the current directory the trained model file StHighCoHigh.model:

  ```
  java -cp bin contextFold.app.Train train:C:/RNAdata/trainData.txt
  ```

- Training the StMedCoHigh model on the dataset in the file C:/RNAdata/trainData.txt, generating in the current directory the trained model file StMedCoHigh.model:

  ```
  java -cp bin contextFold.app.Train train:C:/RNAdata/trainData.txt f:StMedCoHigh
  ```

- Re-training the supplied trained Baseline model on the dataset in the file C:/RNAdata/trainData.txt, generating the trained model file trained/BaselineRetrained.model, terminating after 6 iterations or when the prediction accuracy (f1-measure) does not improve in more than 0.5% in 3 consecutive iterations over the training dataset (whichever comes first):

```
java -cp bin contextFold.app.Train train:C:/RNAdata/trainData.txt
f:trained/Baseline.model iter:6 out:trained/BaselineRetrained improve:0.5
```