Shaping aligned RNAs

Björn Voß

AG Experimentelle Bioinformatik (Cyanolab)
Institut für Biologie II
Universität Freiburg

Oberseminar Bioinformatik
Lehrstuhl für Bioinformatik
Universität Freiburg

4th May, 2006

Structure Prediction

Single Sequence (Free energy algorithms)

- Minimum free energy structure
- Suboptimal folding
- Stochastic sampling
- Shape abstraction
- Shape probabilities
- . . .

Comparative

- Alignment of folded RNAs (MARNA, RNAforester, . . .)
- Simultaneous aligning and folding (Foldalign, Dynalign, . . .)
- Folding aligned RNAs (RNAalifold)

Structure Prediction

Single Sequence (Free energy algorithms)

- Minimum free energy structure
- Suboptimal folding
- Stochastic sampling
- Shape abstraction
- Shape probabilities
- ...

Comparative

- Alignment of folded RNAs (MARNA, RNAforester, ...)
- Simultaneous aligning and folding (Foldalign, Dynalign, ...)
- Folding aligned RNAs (RNAalifold)

Structure Prediction

Single Sequence (Free energy algorithms)

- Minimum free energy structure
- Suboptimal folding
- Stochastic sampling
- Shape abstraction
- Shape probabilities
- . . .

Comparative

- Alignment of folded RNAs (MARNA, RNAforester, . . .)
- Simultaneous aligning and folding (Foldalign, Dynalign, . . .)
- Folding aligned RNAs (RNAalifold)

RNAlishapes

Structural Analysis of Aligned RNAs

Starting point

RNAshapes Version 2.0 (Voß B., Giegerich R. and Rehmsmeier M., 2006)

- **Unambiguous grammar with unique dangles**
- Shape abstraction (Version 1.0)
- Probabilistic shape analysis
- Boltzmann-weighted sampling
- Suboptimal folding with correct energies

RNAalifold (Hofacker I.L., Fekete M., and Stadler P.F., 2002)

- Structure prediction for aligned RNAs
- Scoring based on free energy and covariance contribution

Methodology

- Implemented in ADP
- Alignments as input – What to do with gaps?
- Score as in RNAalifold

Shape Abstraction

Example: Part of Structure Space of tRNA

```
GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
(((((((((((((((.((((.....(((((((...))))))).))))))))))).........)))))))).
(((((((((((((((.((((.....((((((.....)))))).))))))))))).........)))))))).
(((((((((((((((.(((......(((((((...)))))))..)))))))))).........)))))))).
(((((((((((((((.(((......((((((.....))))))..)))))))))).........)))))))).
(((((((.(((((((.((((.....(((((((...))))))).)))))))))))..........))))))).
(((((((.(((((((.((((.....((((((.....)))))).)))))))))))..........))))))).
(((((((((((((((.(((...((.((((((.....)))))))))))))))))).........)))))))).
(((((((..((((((.((((.....(((((((...))))))).))))))))))...........))))))).
(((((((..((((((.((((.....((((((.....)))))).))))))))))...........))))))).
(((((((((((((((.(((......(((((.......)))))..)))))))))).........)))))))).
(((((((.(((((((.((((.....(((((.......))))).)))))))))))..........))))))).
(((((((((((((((.(((...((.(((((.......))))))))))))))))).........)))))))).
((((((((.....((.((((.....(((((((...))))))).))))))(((.......))).)))))))).
((((((((.....((.((((.....((((((.....)))))).))))))(((.......))).)))))))).
((((((((..(((((.((((.....(((((((...))))))).)))))))))...........)))))))).
((((((((..(((((.((((.....((((((.....)))))).)))))))))...........)))))))).
(((((((((((((((.((....((.(((((((...))))))))).))))))))).........)))))))).
(((((((((((((((.((....((.((((((.....)))))))).))))))))).........)))))))).
(((((((((((.(((.((((.....(((((((...))))))).))))))).))).........)))))))).
(((((((((((.(((.((((.....((((((.....)))))).))))))).))).........)))))))).
((((((..........((((.....(((((((...))))))).))))(((((.......))))).)))))).
((((((..........((((.....((((((.....)))))).))))(((((.......))))).)))))).
(((((((.........((((.....(((((((...))))))).))))(((((.......)))))))))))).
(((((((.........((((.....((((((.....)))))).))))(((((.......)))))))))))).
((((((..(((((((.((((.....(((((((...))))))).)))))))))))...........)))))).
((((((..(((((((.((((.....((((((.....)))))).)))))))))))...........)))))).
((((((((.((((((.(((......(((((((...)))))))..))))))))).........)))))))).
((((((...((((.......)))).(((((((...))))))).....(((((.......))))).)))))).
((((((...((((.......)))).((((((.....)))))).....(((((.......))))).)))))).
(((((((.(((((((.(((......(((((((...)))))))..))))))))))..........))))))).
(((((((.(((((((.(((......((((((.....))))))..))))))))))..........))))))).
```

Shape Abstraction

Example: Part of Structure Space of tRNA

```
GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
(((((((((((((((.((((.....(((((((...))))))).))))))))))).........)))))))).
(((((((((((((((.((((.....((((((.....)))))).))))))))))).........)))))))).
(((((((((((((((.(((......(((((((...)))))))..)))))))))).........)))))))).
(((((((((((((((.(((......((((((.....))))))..)))))))))).........)))))))).
(((((((.(((((((.((((.....(((((((...))))))).)))))))))))..........))))))).
(((((((.(((((((.((((.....((((((.....)))))).)))))))))))..........))))))).
(((((((((((((((.(((...((.((((((.....))))))))))))))))))).........)))))))).
(((((((..((((((.((((.....(((((((...))))))).))))))))))...........))))))).
(((((((..((((((.((((.....((((((.....)))))).))))))))))...........))))))).
(((((((((((((((.(((......(((((.......)))))..)))))))))).........)))))))).
(((((((.(((((((.((((.....(((((.......))))).)))))))))))..........))))))).
(((((((((((((((.(((...((.(((((.......))))))))))))))))).........)))))))).
((((((((.....((.((((.....(((((((...))))))).))))))(((.......))).)))))))).
((((((((.....((.((((.....((((((.....)))))).))))))(((.......))).)))))))).
((((((((..(((((.((((.....(((((((...))))))).)))))))))...........)))))))).
((((((((..(((((.((((.....((((((.....)))))).)))))))))...........)))))))).
(((((((((((((((.((....((.(((((((...))))))))).))))))))).........)))))))).
(((((((((((((((.((....((.((((((.....)))))))).))))))))).........)))))))).
(((((((((((.(((.((((.....(((((((...))))))).))))))).))).........)))))))).
(((((((((((.(((.((((.....((((((.....)))))).))))))).))).........)))))))).
((((((..........((((.....(((((((...))))))).))))(((((.......))))).)))))).
((((((..........((((.....((((((.....)))))).))))(((((.......))))).)))))).
(((((((.........((((.....(((((((...))))))).))))(((((.......)))))))))))).
(((((((.........((((.....((((((.....)))))).))))(((((.......)))))))))))).
((((((..(((((((.((((.....(((((((...))))))).)))))))))))...........)))))).
((((((..(((((((.((((.....((((((.....)))))).)))))))))))...........)))))).
((((((((.((((((.(((......(((((((...)))))))..))))))))).........)))))))).
((((((...((((.......)))).(((((((...))))))).....(((((.......))))).)))))).
((((((...((((.......)))).((((((.....)))))).....(((((.......))))).)))))).
(((((((.(((((((.(((......(((((((...)))))))..))))))))))..........))))))).
(((((((.(((((((.(((......((((((.....))))))..))))))))))..........))))))).
```

Shapes

Abstract Shapes of RNA

Classes of similar structures represented by:

- Shape notation:
  Abstract from helix length and length of unpaired regions
  '[' and ']': paired regions, '_': unpaired regions
- *Shrep* (**sh**ape **rep**resentative structure):
  Shape member with lowest free energy

Shape notation - Different abstraction levels

```
         (((..((((.....)))).((((...))))..)))...((..(((.....)))..))..
Level 1  [_[_]_[_]_]_[_[_]_]_
Level 2  [[][]]_[[]]_
Level 3  [[][]][[]]
Level 4  [[][]]_[]_
Level 5  [[][]][]
```

Abstract Shapes of RNA

Classes of similar structures represented by:

- Shape notation:
  Abstract from helix length and length of unpaired regions
  '[' and ']': paired regions, '_': unpaired regions
- *Shrep* (**sh**ape **rep**resentative structure):
  Shape member with lowest free energy

Shape notation - Different abstraction levels

```
         (((..((((.....)))).((((...))))..)))...((..(((.....)))..))..
Level 1  [_[_]_[_]_]_[_[_]_]_
Level 2  [[][]]_[[]]_
Level 3  [[][]][[]]
Level 4  [[][]]_[]_
Level 5  [[][]][]
```

Wanted

Predict shapes together with their shreps without calculating all structures

ADP grammar for RNA folding

Derive candidates for evaluation (Wuchty, no dangling bases)

```
struct  =  str  <<< comps        |||
           str  <<< singlestrand |||
           nil  <<< empty        ... h

block   = tabulated(
                    closed             |||
           blk  <<< region ~~~ closed ... h)

comps   = tabulated(
           cons <<< block  ~~~ comps        |||
           ul   <<< block                   |||
           cons <<< block  ~~~ singlestrand ... h)

closed  = tabulated(
          (hl <<< base ~~~            region            ~~~ base  |||
           sr <<< base ~~~            closed            ~~~ base  |||
           bl <<< base ~~~      region ~~~ closed       ~~~ base  |||
           br <<< base ~~~      closed ~~~ region       ~~~ base  |||
           ml <<< base ~~~      block  ~~~ comps        ~~~ base  |||
           il <<< base ~~~ region ~~~ closed  ~~~ region ~~~ base   )
           ‘with‘ basepairing                                     ... h)

singlestrand =  ss   <<< region
```

ADP grammar for RNA folding

Derive candidates for evaluation (Wuchty, no dangling bases)

```
struct  =  str  <<< comps        |||
           str  <<< singlestrand |||
           nil  <<< empty        ... h    ← apply choice function

block   = tabulated(
                    closed            |||
           blk  <<< region ~~~ closed ... h)

comps   = tabulated(
           cons <<< block  ~~~ comps        |||
           ul   <<< block                   |||
           cons <<< block  ~~~ singlestrand ... h)

closed  = tabulated(
          (hl <<< base ~~~            region             ~~~ base  |||
           sr <<< base ~~~            closed             ~~~ base  |||
           bl <<< base ~~~      region ~~~ closed        ~~~ base  |||
           br <<< base ~~~      closed ~~~ region        ~~~ base  |||
           ml <<< base ~~~      block  ~~~ comps         ~~~ base  |||
           il <<< base ~~~ region ~~~ closed  ~~~ region ~~~ base    )
           ‘with‘ basepairing                                      ... h)

singlestrand =  ss   <<< region
```

ADP grammar for RNA folding

Derive candidates for evaluation (Wuchty, no dangling bases)

```
struct  =  str  <<< comps        |||
           str  <<< singlestrand |||
           nil  <<< empty        ... h    ← apply choice function

block   = tabulated(                      ← store results in table
                    closed            |||
           blk  <<< region ~~~ closed ... h)

comps   = tabulated(
           cons <<< block  ~~~ comps        |||
           ul   <<< block                   |||
           cons <<< block  ~~~ singlestrand ... h)

closed  = tabulated(
          (hl <<< base ~~~            region             ~~~ base  |||
           sr <<< base ~~~            closed             ~~~ base  |||
           bl <<< base ~~~      region ~~~ closed        ~~~ base  |||
           br <<< base ~~~      closed ~~~ region        ~~~ base  |||
           ml <<< base ~~~      block  ~~~ comps         ~~~ base  |||
           il <<< base ~~~ region ~~~ closed  ~~~ region ~~~ base    )
           ‘with‘ basepairing                                      ... h)

singlestrand =  ss   <<< region
```

ADP grammar for RNA folding

Derive candidates for evaluation (Wuchty, no dangling bases)

```
struct  =  str  <<< comps        |||
           str  <<< singlestrand |||
           nil  <<< empty        ... h    ← apply choice function

block   = tabulated(                      ← store results in table
                    closed            |||
           blk  <<< region ~~~ closed ... h)

comps   = tabulated(
           cons <<< block  ~~~ comps        |||
           ul   <<< block                   |||
           cons <<< block  ~~~ singlestrand ... h)

closed  = tabulated(
          (hl <<< base ~~~            region             ~~~ base  |||
           sr <<< base ~~~            closed             ~~~ base  |||
           bl <<< base ~~~      region ~~~ closed        ~~~ base  |||
           br <<< base ~~~      closed ~~~ region        ~~~ base  |||
           ml <<< base ~~~      block  ~~~ comps         ~~~ base  |||
           il <<< base ~~~ region ~~~ closed  ~~~ region ~~~ base    )
           `with` basepairing ← check for basepairing              ... h)

singlestrand =  ss   <<< region
```

Scoring with Algebras

Example candidate

SR(1(*SR*(2(*SR*(3(*HL*(4(5, 6, 7, 8)9))10)11)12) ⇒ ((((....))))

Note: The candidate is composed of operators and indexes and contains no sequence information

Function	Energy	Dot Bracket	Shape
HL a r b =	stackE(a,b) + unpE(r)	'(' + '...' + ')'	'[' + ']'
SR a x b =	x + stackE(a,b)	'(' + x + ')'	x
BL a r x b =	x + stackE(a,b) + unpE(r)	'(' + '...' + x + ')'	x
BR a x r b =	x + stackE(a,b) + unpE(r)	'(' + x + '...' + ')'	x
IL a r x r‘ b =	x + stackE(a,b) + unpE(r+r‘)	'(' + '...' + x + '...' + ')'	x
ML a x b =	x + stackE(a,b)	'(' + x + ')'	'[' + x + ']'
AD x x' =	x + x'	x + x'	x + x'
SS r =	unpE(r)	'...'	”
h =	Minimum	Identity	Identity

RNAshapes – Combination of these algebras

RNAshapes = (Energy, Dot Bracket, Shape)
Choice funtion: Filter for identical shape and keep answer with lowest energy

Multiple Sequence Alignments as Input

Problems & Tasks

- Handle multiple sequences and combine scores
- Check for basepairing (All, majority, at least 1)
- Non-standard base pairs (e.g. one sequence with A-C instead of G-C)
- Insertion in one sequence introduces long gap (energy depends on length)
- Covarying positions are of special interest

Solutions

- Overall score = mean of individual scores
- Basepairing fraction defined by user (default: at least 1)
- Enhanced thermodynamic model:
  - Non-standard base pairs (also those with gaps) get 0.0 kcal/mol
  - Gap-aware handling of singlestranded regions (different from RNAalifold)
  - Covariance score (reward covariance, penalize non-standard base pairs)

Handling Alignments in ADP

Grammar

- The grammar does not work directly on the input, but on indexes.
- Grammar predicates (applied via “with”) may work on the input.

⇒ A Grammar can be used for any kind of sequential data
⇒ Predicates might need to be adapted for input.

“basepairing” predicate

For single sequence S:

$$\mathit{basepairing}(i,j) = \text{if } \mathit{basepair}(S[i],S[j]) = 1 \text{ then } \mathit{true} \text{ else } \mathit{false}\text{, where}$$

$$\mathit{basepair}(x,y) = \begin{cases} 1, (x,y) \in \{(A,U),(U,A),(G,C),(C,G),(G,U),(U,G)\} \\ 0, \text{otherwise} \end{cases}$$

For alignment M (User-defined CUT-off):

$$\mathit{basepairing}(i,j) = \mathit{if}\ \frac{1}{N}\sum_{x\in M} \mathit{basepair}(x[i],x[j]) > CUT \text{ then } \mathit{true} \text{ else } \mathit{false}$$

Algebras

Algebra Adaptation

Algebras that do not directly utilise the input don't need adaptation. Only algebras for computing the free energy or partition function need to be changed.

Energy

Energy E of subword i, j for alignment M holding N sequences:

$$E_M(i,j) = \frac{1}{N} \sum_{x \in M} E_x(i,j)$$

$\Rightarrow$ Individual energy function:
For sequence S:

$$SR(i,x,j) = x + sr\_energy(S[i], S[j])$$

For alignment M:

$$SR(i,x,j) = x + \frac{1}{N} \sum_{x \in M} sr\_energy(x[i], x[j])$$

Gap-aware single-strand handling

Example

```
CUUCAUCAGUA.AAAGCUUGGAGAAGAAUGAGCUUCAAUGAAAAGCUUUGAAAGGGAAC
GCCUAUGAC....UACUUGUGCGGAGGGUGAUGCCGC.AGAUGUACAAGGAAAGGAGUC
GCCCAGGCAG...AUGUUUUGUGGAGCCGCAACUCCAACACAGAACAUUCAGGGGGAGU
AACUAGGUAGU..UCAAUCAGAGGAGCACAAACUCCAGCGAUGAUUGAUGAGGGAGAUU
AAGCAUGUAUUUGGCGAGGUGUUAAGGAGAAGAACCUCCAAUACUCGCUGAAGAAGGUU
((((.........((((((((((..(((.....)))..)))))))))).......))))
```

Gap-aware single-strand handling

Example

```
CUUCAUCAGUA.AAAGCUUGGAGAAGAAUGAGCUUCAAUGAAAAGCUUUGAAAGGGAAC
GCCUAUGAC....UACUUGUGCGGAGGGUGAUGCCGC.AGAUGUACAAGGAAAGGAGUC
GCCCAGGCAG...AUGUUUGUGGAGCCGCAACUCCAACACAGAACAUUCAGGGGGAGU
AACUAGGUAGU..UCAAUCAGAGGAGCACAAACUCCAGCGAUGAUUGAUGAGGGAGAUU
AAGCAUGUAUUUGGCGAGGUGUUAAGGAGAAGAACCUCCAAUACUCGCUGAAGAAGGUU
((((.........((((((((((..(((.....)))..)))))))))).......))))
```

Internal loop

- Size of subword: 9 nt ⇒ asymmetric internal loop with 9 and 7 nt, resp.
- Actually: 5'-region: 5,6,7,8 or 9 nt, 3'-region: 7 nt
- This means: Mixture of symmetric and asymmetric internal loops with different loop lengths

Handling gaps in unpaired regions

- Different lengths: Recompute size of unpaired regions excluding gaps
- Gaps-only: Switch loop type (e.g. internal → bulge)

Covariance Scoring

Covariation & Inconsistency

Covariation: compensatory (A-U → G-C) and consistent (A-U → G-U)

$$C_{ij} = \sum_{a,b,a',b' \in \{A,C,G,U\}} f_{ij}(a,b) * D(a,b,a',b') * f_{ij}(a',b')$$

$$D(a,b,a',b') = \begin{cases} 0, \text{not } (bp(a,b)|bp(a',b'))|(a,b) = (a',b') \\ 1, a = a' \ xor \ b = b' \\ 2, \text{otherwise} \end{cases}$$

Inconsistency: Non-standard base pairs are allowed (0.0 kcal/mol) but need to be penalised. Gap-Gap pairs don't get penalised.

$$I_{ij} = \frac{1}{N} \sum_{x \in M} \begin{cases} 0, x_i = x_j = gap|bp(x_i, x_j) \\ 1, \text{otherwise} \end{cases}$$

⇒**Covariance Score:** $cv_{ij} = -C_{ij} + I_{ij}$

RNAlishapes

Implementation

- Implemented in Haskell-ADP (Hopefully soon in C)
- Following analysis modes are supported:
  - Optimal consensus structure
  - Suboptimal consensus structures
  - Shape analysis (5 abstraction levels)
  - Boltzmann-weighted sampling (like SFOLD)
  - Shape probabilities
- User options:
  - Weight of covariance score
  - Minimum fraction of actual base pairs at pairing positions
- Reads alignments in CLUSTALW format

tRNAs - Proof of Correctness

Alignment and Predicted Consensus

images/tRNA_example_ungap_ali_coloured.pdf

tRNAs - Suboptimal Consensus Structures

Energy range: 3 kcal/mol

```
       GCGUUCGUAGCUCAGUU-GGU--AGAGCAUCUGGUUUUGACCCUGAAUGUCAUGGGUUCGAAUCCCGUCGGUCGCG
-28.97 (((((((..((((...........))))..((((((...))))))......(((((.......)))))))))))).
-29.03 (((((((..((((...........))))..(((((.....)))))......(((((.......)))))))))))).
-30.31 (((((((..((((...........))))..((((.......))))......(((((.......)))))))))))).
-29.72 (((((((..((((.((...))...))))..((((.......))))......(((((.......)))))))))))).
-29.44 (((((((..(((.............)))..((((.......))))......(((((.......)))))))))))).
-30.15 (((((((..((((...........)))).(((((((...))))))).....(((((.......)))))))))))).
-30.21 (((((((..((((...........)))).((((((.....)))))).....(((((.......)))))))))))).
-31.49 (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).
-28.96 (((((((..((((...........)))).(((((.......))))).....((((.........))))))))))).
-29.56 (((((((..((((.((...))...)))).(((((((...))))))).....(((((.......)))))))))))).
-29.62 (((((((..((((.((...))...)))).((((((.....)))))).....(((((.......)))))))))))).
-30.9  (((((((..((((.((...))...)))).(((((.......))))).....(((((.......)))))))))))).
-29.28 (((((((..(((.............))).(((((((...))))))).....(((((.......)))))))))))).
-29.34 (((((((..(((.............))).((((((.....)))))).....(((((.......)))))))))))).
-30.62 (((((((..(((.............))).(((((.......))))).....(((((.......)))))))))))).
-29.43 (((((((..((((...........)))).(((((.......)))))......((((.......)))).))))))).
-28.84 (((((((..((((.((...))...)))).(((((.......)))))......((((.......)))).))))))).
-29.02 ((((((...((((...........))))..((((.......))))......(((((.......))))).)))))).
-28.86 ((((((...((((...........)))).(((((((...))))))).....(((((.......))))).)))))).
-28.92 ((((((...((((...........)))).((((((.....)))))).....(((((.......))))).)))))).
-30.2  ((((((...((((...........)))).(((((.......))))).....(((((.......))))).)))))).
-29.61 ((((((...((((.((...))...)))).(((((.......))))).....(((((.......))))).)))))).
-29.33 ((((((...(((.............))).(((((.......))))).....(((((.......))))).)))))).
```

tRNAs - Suboptimal Shapes

Most abstract shape, Energy range: 15 kcal/mol

```
        GCGUUCGUAGCUCAGUU-GGU--AGAGCAUCUGGUUUUGACCCUGAAUGUCAUGGGUUCGAAUCCCGUCGGUCGCG
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......))))))))))))).  [[] [] []]
-28.25  (((((((......................(((((.......))))).....(((((.......))))))))))))).  [[] []]
-25.89  (((((((((((((...........)))).(((((.......))))).))..(((((.......))))))))))))).  [[[] []] []]
-20.74  ((((((((((..((.((.((.((..(((.(((((.......)))))..)))..))..)))).)).)))))))))).  []
-19.11  .........((((...........)))).(((((.......))))).....(((((.......)))))........  [] [] []
-18.5   ((((((...((((...........))))((((((.......)))).......((((.......)))))))))))).  [[] [[] []]]
```

Less abstract shape, Energy range: 7 kcal/mol

```
        GCGUUCGUAGCUCAGUU-GGU--AGAGCAUCUGGUUUUGACCCUGAAUGUCAUGGGUUCGAAUCCCGUCGGUCGCG
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......))))))))))))).    [[] [] []]
-30.9   (((((((..((((.((...))...)))).(((((.......))))).....(((((.......))))))))))))).    [[[]] [] []]
-28.25  (((((((......................(((((.......))))).....(((((.......))))))))))))).    [[] []]
-27.0   (((((((...................((.(((((.......)))))..)).(((((.......))))))))))))).    [[[]] []]
-26.1   ((((.((..((((...........)))).(((((.......))))).....(((((.......))))))).)))).    [[[] [] []]]
-25.89  (((((((((((((...........)))).(((((.......))))).))..(((((.......))))))))))))).    [[[] []] []]
-25.86  (((((((..((((...........)))).((((.((...)).)))).....(((((.......))))))))))))).    [[] [[]] []]
-25.51  ((((.((..((((.((...))...)))).(((((.......))))).....(((((.......))))))).)))).    [[[[]] [] []]]
-25.33  (((((((..((((...........)))).(((((.......))))).....((.((.......)).))))))))).    [[] [] [[]]]
-25.3   (((((((((((((.((...))...)))).(((((.......))))).))..(((((.......))))))))))))).    [[[[]] []] []]
-25.27  (((((((..((((.((...))...)))).((((.((...)).)))).....(((((.......))))))))))))).    [[[]] [[]] []]
-24.74  (((((((..((((.((...))...)))).(((((.......))))).....((.((.......)).))))))))).    [[[]] [] [[]]]
```

tRNAs - Boltzmann-weighted sampling

20 samples

```
        GCGUUCGUAGCUCAGUU-GGU--AGAGCAUCUGGUUUUGACCCUGAAUGUCAUGGGUUCGAAUCCCGUCGGUCGCG
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-28.49  .((((((..((((...........)))).(((((.......))))).....(((((.......))))))))))))..    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
-31.49  (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    [[][][]]
```

tRNAs - Shape probabilities

Shape probabilities

```
       GCGUUCGUAGCUCAGUU-GGU--AGAGCAUCUGGUUUUGACCCUGAAUGUCAUGGGUUCGAAUCCCGUCGGUCGCG
-31.49 (((((((..((((...........)))).(((((.......))))).....(((((.......)))))))))))).    0.99606776      [[][][]]
-28.25 (((((((......................(((((.......))))).....(((((.......)))))))))))).    4.781712e-4     [[][]]
-25.89 (((((((((((((...........)))).(((((.......))))).))..(((((.......)))))))))))).    3.4460078e-3    [[[][]][]]
-22.38 (((((((..((...))...((.....)).(((((.......))))).....(((((.......)))))))))))).    7.973717e-6     [[][][][]]
```

Attenuators of bacterial trp-operons

Biology

- Attenuation is important mechanism of gene regulation
- Formation of alternating structures
- Functions:
  - Inhibition of translation initiation
  - Premature termination of transcription

Example: trp-operon

- 8 leader regions
- Multiple sequence alignment, ClustalW

Attenuators of bacterial trp-operons

Shape analysis

images/trp_attenuator_ali_structure1_coloured.png

Low quality Alignment

T-box sequences, Avg. PI 59%

images/t-box_alignment_structure_coloured.png

Summary

Algorithm

- Structural analysis of aligned RNAs
- Combines shape abstraction and alignment folding
- Covariance Scoring
- Gap-aware thermodynamics
- User-defined pairing cut-off

Applications

- Modes: MFE, suboptimal, shapes, sampling, shape porbabilities
- Structural features of RNA families, e.g.
  - Robustness of MFE
  - Switching
- Improved predictions for low-quality, esp. gap-rich, alignments

Discussion

Pros, Cons & Outlook

- Strong dependence on alignment quality
  ⇒ Use MARNA?
- Computationally expensive
  - Gap-counting
  - Haskell implementation
- Improve structure analysis and prediction
- Replace RNAalifold within RNAz

Acknowledgements

- Wolfgang Hess (Freiburg University)
- Robert Giegerich and Marc Rehmsmeier (Bielefeld University)
- Cyanolab people (Freiburg University)

images/group2.jpg

Thank You!