**Summary**

The majority of traditional computational approaches to RNA secondary structure prediction consist of two components:
- Dynamic programming algorithm for decoding the structure, and
- A parameterization method.

While the dynamic programming algorithm is quite similar for all these methods parameter inference is usually either in a probabilistic way or using experimental thermodynamical data. A recently, (1, ) proposed probabilistic approach used a conditional maximum likelihood scheme for parameter inference. This model could outperform existing thermodynamic models. We will use a large margin method related to Support Vector Machines. The central idea is to find a parameter vector that assigns highest score to correct and lower score to incorrect structures.

**Problem Setting**

An RNA Secondary Structure for a nucleotide sequence of length $N$ can be seen as a set of ordered pairs $(i,j)$ denoting that nucleotide at position $i$ is paired with nucleotide at position $j$.

If $(i,j)$ and $(i',j')$ are two pairs (w.l.o.g $i \leq i'$), then either :
- $i = i'$ and $j = j'$,
- $i < j$ and $i' < j'$, or
- $i < j'$ and $i' j < j$.

So our input domain is $X = N^2$ where $N = \{1,2,3,..,N\}$ and our output domain is $Y = \{(i,j) | 1 \leq i < j \leq N\}$. Based on these pairs one can identify substructures such as stems, hairpins, etc.

**The Dynamic Programming Component**

- In thermodynamic models: Total free energy of $y$ is the sum the energies of the substructures.
- In a probabilistic setting: Log probability of a structure is the sum of the log probabilities. These values can be expressed as a dot product between parameters and feature (substructure) counts:

$$y = \text{argmax}_y P(y|x) = \text{argmax}_y \langle w, \phi(x,y) \rangle$$

which calculates the structure $y \in Y$ whose probability of being exactly equal to the correct one $\hat{y}$ is optimal. The result of the above maximization is usually calculated using Dynamic Programming and is called Viterbi decoding.

Informally the dynamic programming algorithm has to check for each position $i$ against all remaining positions $j \in \{1,2,..,N\}$ whether a pair occurs between these positions. When looking at two particular positions $i$ and $j$ of the nucleotide sequence we can identify several possibilities:

1. A perfect match $i = j$ and $j = j'$, which is exactly the output of the Viterbi decoding.
2. This means we have to find a $w$ which maximizes the score (Note that our parameters $w$ cannot be interpreted as probabilities anymore) of the correct structure.
3. In optimization lingos this means we want constraints to hold such that:

$$F(x,y) \geq F(x,y') + 1 \quad \forall y \in Y,$$

which means informally: For each example $(x,y) \in D$ should assign highest score to the correct structure $y$, and a lower score to all incorrect structures $y'$. The $+1$ term is a so-called margin we enforce.

- Generating a constraint for each possible incorrect structure leads to a number of constraints exponential in the size of the structure.
- We use a technique called column generation to approximate the solution

This is done by using a modified viterbi which incorporates some loss terms:

$$\min_{y \in Y} \sum_{i,j \in Y} \max \{0,\psi(x,y)\}$$

- Above maximization can be efficiently computed via dynamic programming.
- It was reported that maximum expected accuracy outperforms minimum expected accuracy. Results reported with using plain Viterbi are very close.

**Slack / Margin rescaling workarounds showed no benefit.**

Rfam consists of consensus structures we instead assume that objective is convex.

$$\epsilon + \psi(x,y)$$

Choice of Features and Regularization

- The number of features: 1216.
- Features include hairpin, stem sizes, occurrence of certain motifs, etc.
- Feature dependencies/relations modeled via regularization term

Our quadratic program has a regularization term $\epsilon^2$.

Naive regularization could be $P = 1$ (identity matrix)

We would like to include biological prior knowledge in $P$, such as:
- Loop size parameters should be "smooth".
- No difference between $A$ and $U$.
- Coupling of the parameters: $(i,j) + (j,i)$

$$\Rightarrow$$ Proper regularization affects sensitivity / specificity by $\pm 5\%$ (in total) each.

**Structural Loss**

What if there is no best structure ?

- Maybe the highest scoring structure is not the correct one but the second highest scoring.
- Achieving structural loss of zero is not possible at all.

It might make sense to allow for certain number of positions to be incorrect during training.

$$\Rightarrow$$ $\lambda$ could be insensitive to a certain loss range.

We investigated several forms of $\epsilon$-insensitive loss, namely:
- Loss insensitive for $\lambda$ positions,
- Loss insensitive for $\lambda/2$ length($x$) positions, and
- Loss insensitive for $\epsilon_1$, $\epsilon_2$ length($x$) at least $\lambda$ positions.

The $\lambda/2$ insensitive loss performed best for these cases. However only marginal better than a standard Hamming loss.

**Results**

- Data set proposed in (1, )
- Data set is a subset of the Rfam database (2, )
- Consists of 11 secondary structures

Our method is denoted by SER.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls</td>
<td>0.73</td>
<td>0.86</td>
</tr>
<tr>
<td>UA</td>
<td>0.78</td>
<td>0.85</td>
</tr>
<tr>
<td>SVM</td>
<td>0.69</td>
<td>0.66</td>
</tr>
<tr>
<td>RFP</td>
<td>0.66</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Results were obtained by five-fold cross validation.

**Discussion**

As we have seen there is still a question to be addressed: Are probabilistic models superior to large-margin methods in structure prediction ?

The conditional likelihood approach and our large-margin method have a lot in common:
- Feature set seems pretty fixed and very similar
- Regularization offers a lot of tuning possibilities but is also very similar and reported to contribute only 2% in total.
- Objective is convex.
- Results reported with using plain Viterbi are very close.

$$\Rightarrow$$ However there is still a gap in terms of sensitivity and specificity.

Possible explanations:
- It was reported that maximum expected accuracy outperforms Viterbi (1, ).
- Rfam consists of consensus structures we instead assume that there is one correct label.

Among the things we tried are:
- Different regularization terms max $\pm 5\%$ sensitivity / specificity.
- Margin / Rescaling workarounds showed no benefit.
- Insensitive structural loss function

**References**

- Bateman. Rfam: annotating non-coding RNAs in complete genomes.
- Contrafold. V.151.