Of the importance of being random

Structural clustering

We wish to test the hypothesis that the native structure is well represented in the Boltzmann ensemble by a set of similar structures. As mentioned in the second lecture, one way to test such an hypothesis consists in:

- Sampling a set $\mathscr{S}$ of suboptimal structures;
- Cluster $\mathscr{S}$ with respect to structural similarity;
- Consider the heaviest cluster;
- Return the centroid structure as the most likely candidate for the native structure.

Stochastic backtrack

So far, the backtrack that has been implemented produces (one of) the minimal free energy secondary structure(s) for a given sequence.

Starting from the code that has been provided in the corrected version of the first session, implement a function `stochasticBacktrack(tab,seq,T=37.)`, which takes as input the matrix produced by the computation of the partition function and a temperature in Celsius, and generates a secondary structure, represented by its well-parenthesized expression. Each compatible structure must be generated with probability equal to the Boltzmann probability.

Example:

Input		Output
`tab = partitionFunction("CCC")`		
`stochasticBacktrack(tab,seq)`	$\longrightarrow$	`"..."`

Input		Output
`tab = partitionFunction("CCG",T=sys.maxint)`		
`stochasticBacktrack(tab,seq,T=sys.maxint)`	$\longrightarrow$	`"..."` with prob. 1/2
	$\longrightarrow$	`"(.)"` with prob. 1/2

Basic structural clustering

Given a set $\mathscr{S}$ of secondary structure, we wish to cluster them, to partition them into sets $\mathscr{S}_1, \mathscr{S}_2, \ldots, \mathscr{S}_k$ such that

$$\forall i \neq j \in [1,k], \mathscr{S}_i \cap \mathscr{S}_j = \varnothing \qquad \text{and} \qquad \bigcup_{i=1}^{k} \mathscr{S}_i = \mathscr{S},$$

and such that the differences between two structure in $\mathscr{S}_i$ are minimal.

Given a similarity threshold $\delta \in \mathbb{N}$, our algorithm produces a set $\mathscr{C}$ of clusters, initially set to $\varnothing$, in the following way:

1. Consider the minimum free-energy structure $S_0 \in \mathscr{S}$;
2. Take the set $\mathscr{S}' \subset \mathscr{S}$ of structures in $\mathscr{S}$ at maximal distance δ of S;
3. $\mathscr{S}'$ becomes a cluster ($\mathscr{C} \leftarrow \mathscr{C} \cup \{\mathscr{S}'\}$), and is removed from $\mathscr{S}$ ($\mathscr{S} \leftarrow \mathscr{S} - \{\mathscr{S}'\}$);
4. If $\mathscr{S} \neq \varnothing$, iterate from step 1. Otherwise, return $\mathscr{C}$.

Implement a function `cluster(structs,delta=3)` which, given a sorte list of secondary structures `structs` and a similarity threshold `delta`, returns a set of sets of secondary structures produced by the algorithm above.
Example:

Input	Output
`cluster([("..)","(.)","..."],1)`	`set(set(["(..)","..."]),set(["(.)."]))`

Weight of a cluster

We now need to order the clusters with respect to their respective Boltzmann weight. Implement a function `BoltzmannWeight(clust, T=37)` which, given a set of structures `clust` and a temperature `T`, returns the sum of Boltzmann factors

$$\sum_{S \in \texttt{clust}} e^{\frac{-E_S}{RT}}.$$

Remark: The free-energy E_S of a structure S will be computed using the `runRNAEval` function from the `ViennaWrappers` library, and the value of RT will be sought in the corrected version of the first session.

Centroid structure

Now we are able to decide which is the most promising (i.e. heaviest) cluster, but we still need to associate a single representative structure to the cluster. To that purpose, we introduce a function `centroid(structs,T=37)`, which returns the secondary structure composed of base-pairs which consist of base-pairs having probability greater than 1/2 in the cluster.

In other words, one must associate with each base-pair (i, j), occurring in at least one of the structures in `clust`, its total Boltzmann probability $\mathscr{P}_{i,j}$, defined as:

$$\mathscr{P}_{i,j} = \sum_{\substack{S \in \texttt{clust} \\ \text{s.t. } (i,j) \in S}} \frac{e^{\frac{-E_S}{RT}}}{\mathscr{Z}}$$

where $\mathscr{Z}$ is the partition function. The secondary structure returned by the `centroid` function is the set of base-pairs (i, j) such that $\mathscr{P}_{i,j} > 1/2$.

Benchmarking the centroid structure hypothesis

Adapt the benchmarking script, previously used to test the predictive power of `RNAfold` and your implementation of the Nussinov algorithm, and test the whole pipeline based on structural clustering.

Probability of a structural motif in the Boltzmann ensemble

In non-coding RNA families, it is somehow classic to postulate the responsibility of a structure motif for some function of RNA. Such a motif can in turn be used to detect novel instances of such an RNA. This begs for the following question: *How to detect the significantly enriched presence of a motif in the Boltzmann ensemble for a given RNA?* A a very basic level, and for a given motif m, this requires us to be able to compute the Boltzmann probability of the motif:

$$\mathscr{P}_m = \frac{\sum_{S \text{ s.t. } m \subset S} e^{\frac{-E_S}{RT}}}{\mathscr{Z}} \tag{1}$$

Through sampling

Let us first estimate the value of $\mathscr{P}_m$ through sampling. To that purpose, you must code a function `sampleProb(seq, motif, K=1000,T=37)` which takes a sequence `seq`, a set of base-pairs `motif`, an integer `K` and a temperature `T`. It generates `K` structures at random using stochastic backtrack, and then simply returns the proportion of sampled structures which contain all of the base-pairs in `motif`.

From dot-plot

`RNAfold` accepts an option `-t`, which induces the computation of the partition function, and the derivation of various thermodynamic quantities (entropy, ensemble diversity...). Among these quantities, the software can produce a base-pair probability matrix, also called *dot-plot*, which associates a Boltzmann probability $p_{i,j}$ to each base-pair (i, j). It is then possible to postulate the independence of these probabilities, and estimate the probability of a motif as the product of its base-pair probabilities.

$$\mathscr{P}_m \approx \sum_{(i,j) \in m} p_{i,j}$$

Implement a function `BPProb(seq, motif, T=37)` which estimates the probability of a motif based on the above (biased) approximation.

As a constrained partition function

Partition function computation software can be easily adapted to *force* some structure elements to appear, disallowing any conflicting alternative. `RNAfold` can be invoked using the `-c` option to define a mask/motif m, and provide a partial characterization of

the structure. In combination with the `-p` option, this allows to compute the *restricted partition function*

$$\mathcal{Z}_m \approx \sum_{\substack{S \\ \text{s.t. } m \subset S}} e^{\frac{-E_S}{RT}}.$$

Write a wrapper for this mode of `RNAfold`, and use it to code a function `exactProb(seq, motif, T=37)` which computes exactly the probability of a motif $\mathscr{P}_m$, as defined in Eq. 1.

Compare the precision of the two estimates obtained using `sampleProb` and to the result of `exactProb` for the first sequences of the Benchmark. Comment the output.