Proceedings of Locarna-Scan

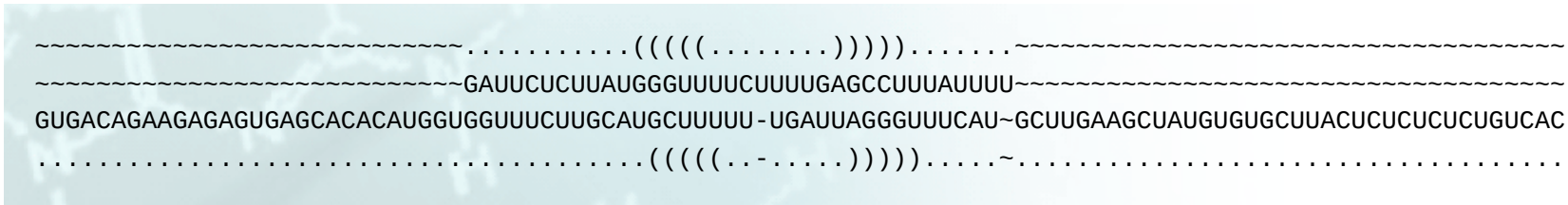
Michael Siebauer

MPI EVA

February 17, 2011

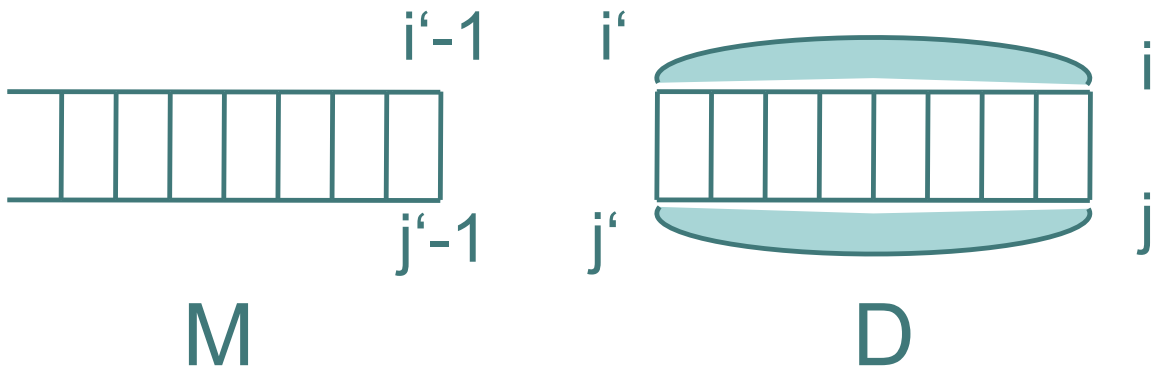
Introduction

- Locarna-Scan is a branch of Locarna in terms of homology search
- Homology search is the process of finding sequences homologous to a certain *query* sequence within a much longer *target* sequence.
- in this work by means of semi-global sequence-structure alignments



Locarna-algorithm

- base-pairing probabilities for all combinations are calculated in advance using RNAplfold
- The idea is to split the alignment into an unstructured (M) and a arc-enclosed alignment part (D).
- this arc-enclosed alignment consists again of unstructured and arc enclosed alignment



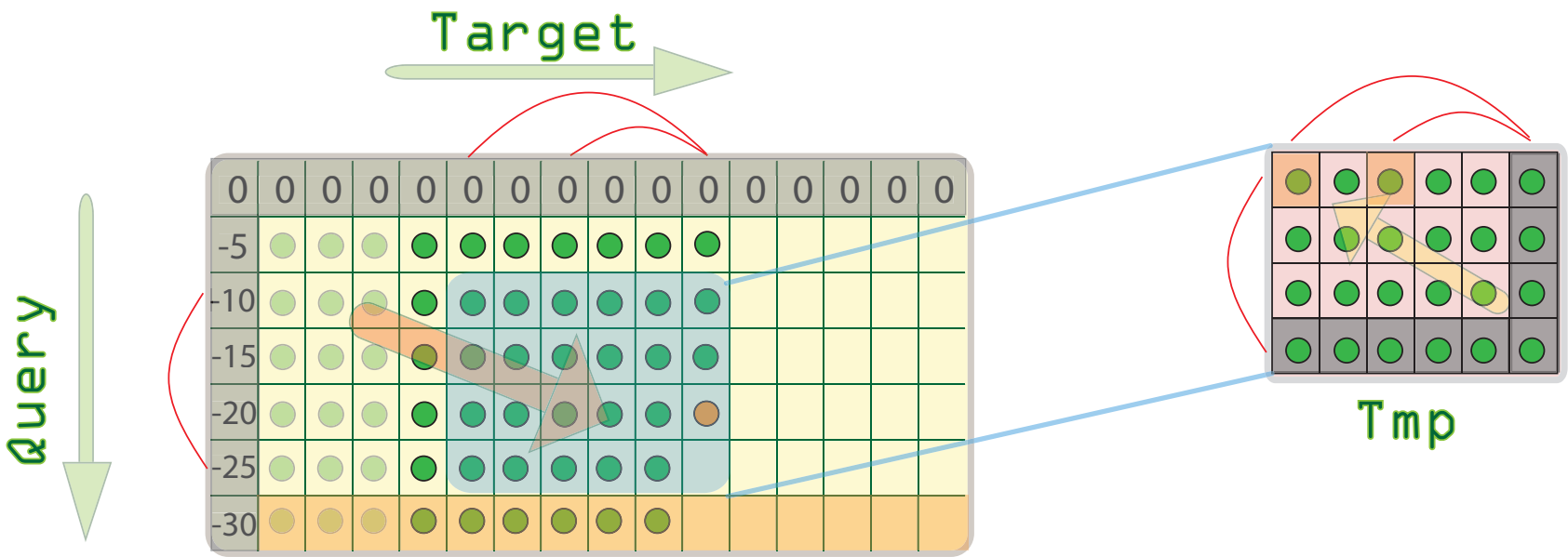
Optimization

There are several ways to reduce this last recursion step:

1. calculate and store all arc enclosed alignments on-the-fly only when needed(!)
2. Minimal arc probability cutoff (p_{min}) limits arcs per position to linear number $\frac{1}{p_{\text{min}}}$
3. Every length difference between two arcs has to be paid by gaps. Thus calculate arc combinations (ij;kl) only if: $|(j-i)-(l-k)| \leq \Delta$ cutoff

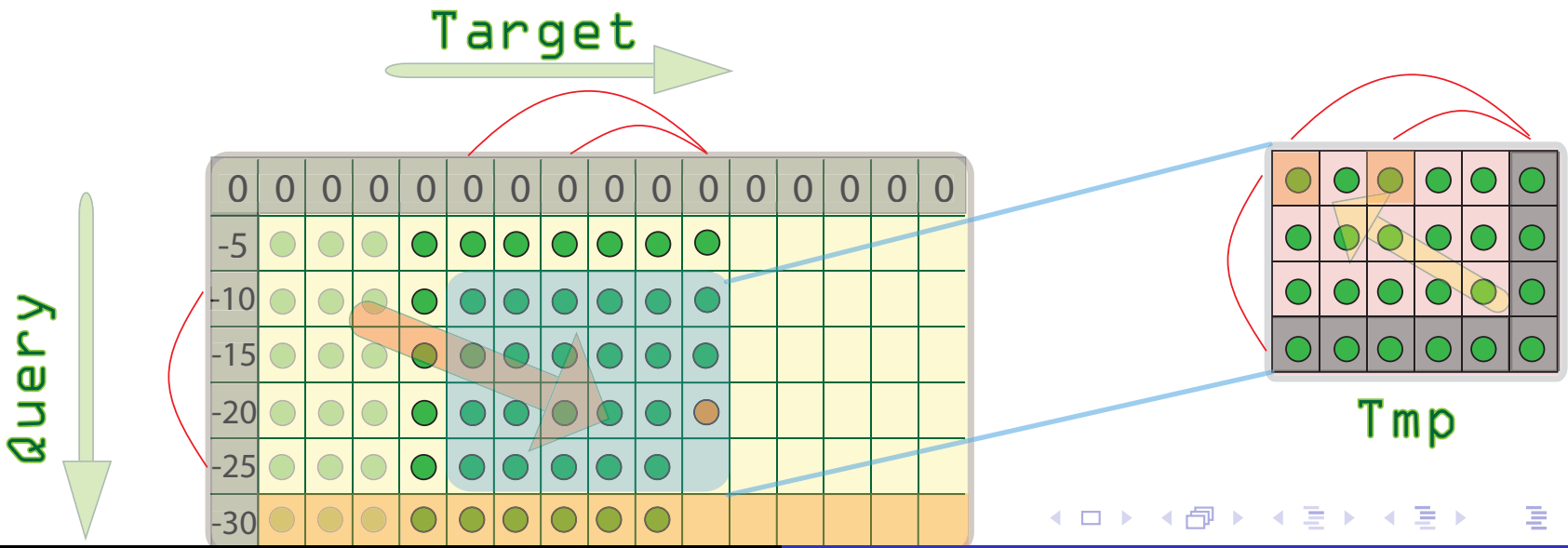
Locarna-Scan

- tmp-matrix is a global suffix alignment
- all eventual arc-enclosed alignments WITHIN have already been calculated
- calculation only for the longest arcs $\longrightarrow$ all shorter are contained in the matrix



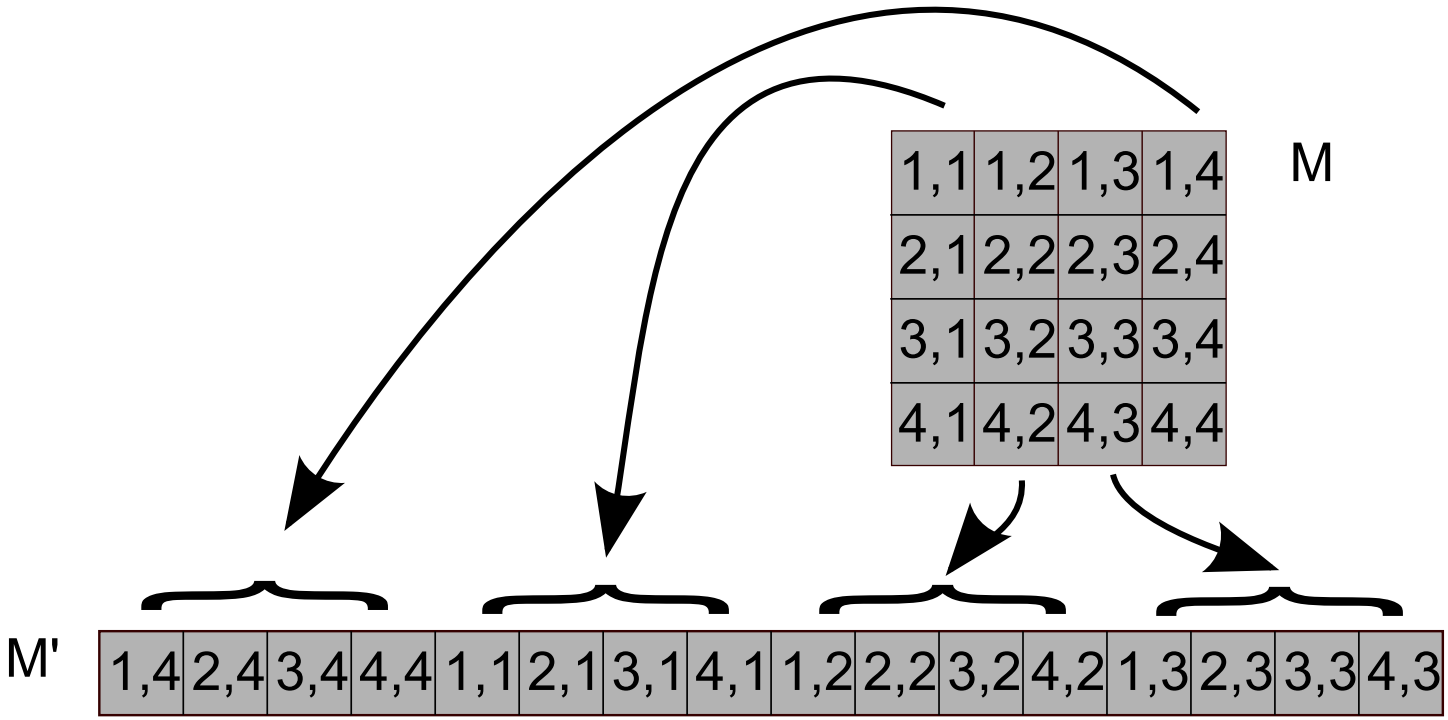
Optimizations

- D-Matrix is very sparse → using of a sparse matrix
- multiple (profile) queries → combined into PSSM
- preprocessing of the arc-probability matrix (dotplot) → arc sorting and determination of important values (like maximal arc span)
- only very small part of the scoring matrix has to remain in memory → query-length × longest-arc-span + 1



Matrix tricks

- scoring matrix is a rotation matrix $\longrightarrow$ overwrites outdated (no longer needed) values with new ones
- mapping of the matrix structure onto a linear array avoids costly memory jumps



Output

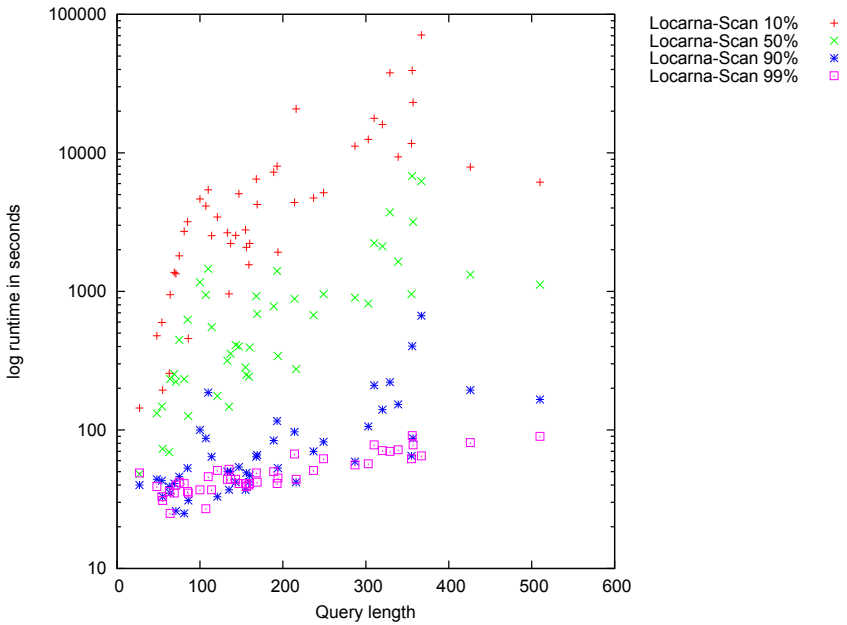
- Locarna-Scan gives you a score for each position along the target
- scores should increase along a hidden query, reach its maxima at the end and decrease afterwards $\rightarrow$ local maxima point to "good" alignment ends
- starting from the global(!) maxima, other local maxima within a certain distance are discarded (at least one query length distance in both directions)

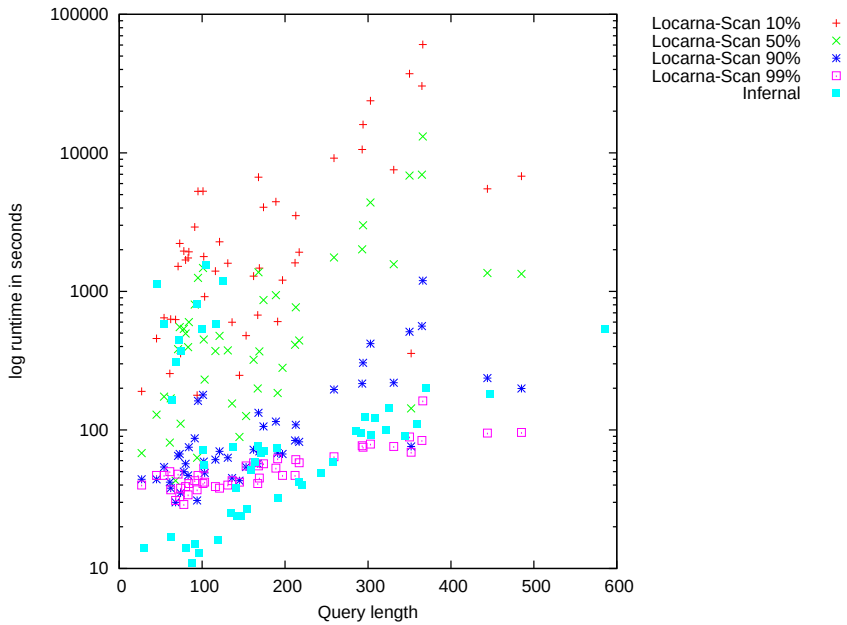
Verification

- For verification a comparison against Infernal and RSearch was made.
- 450 ncRNA sequences from 51 RNA families were hidden in a 20×500kb artificial genome.
- The aim was to retrieve those sequences again.

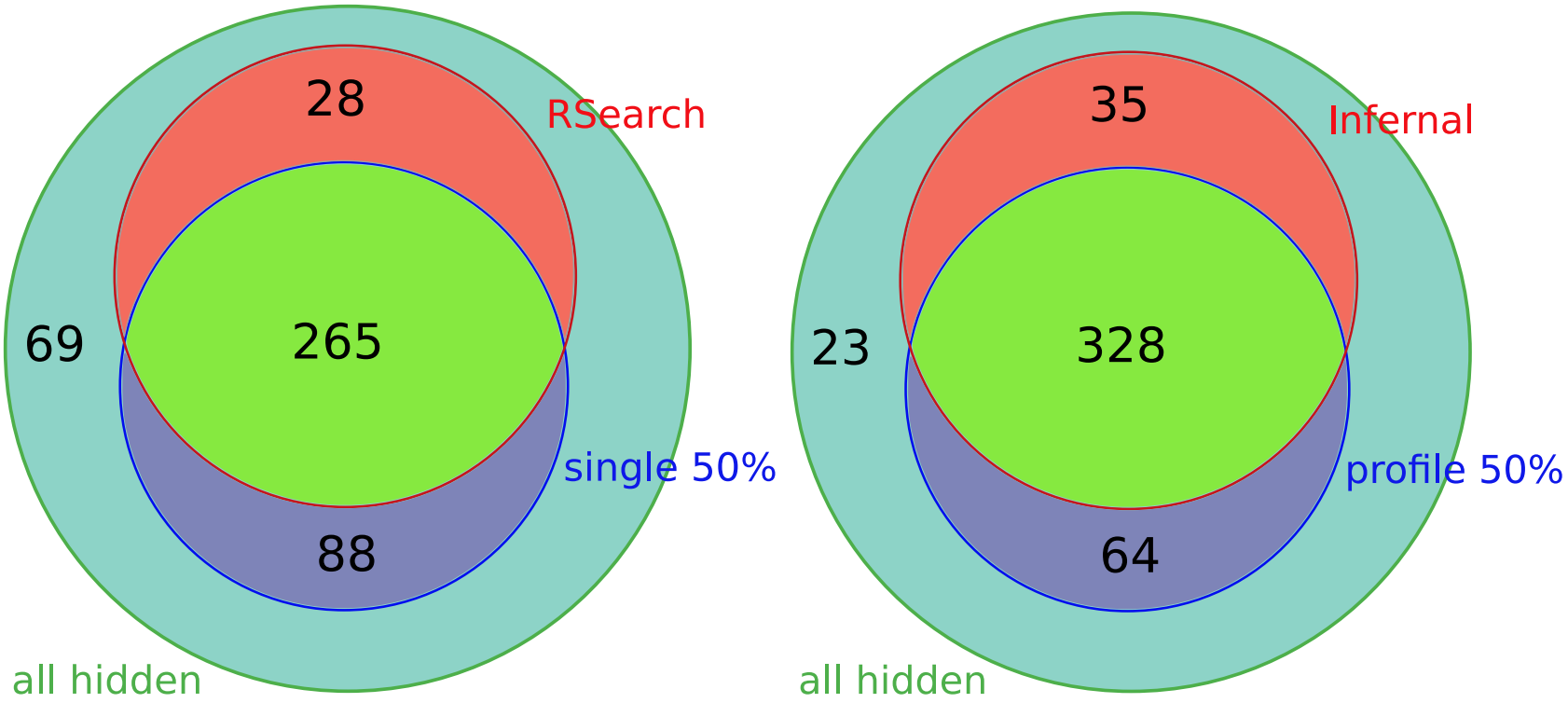
Runtime

- 2 different query sets (single query against RSearch, Profile query (alignment of several) against Infernal)
- different minimum arc-probabilities (10%, 50%, 90%, 99%)
- scanning of fwd. and rev. compl. strand separate





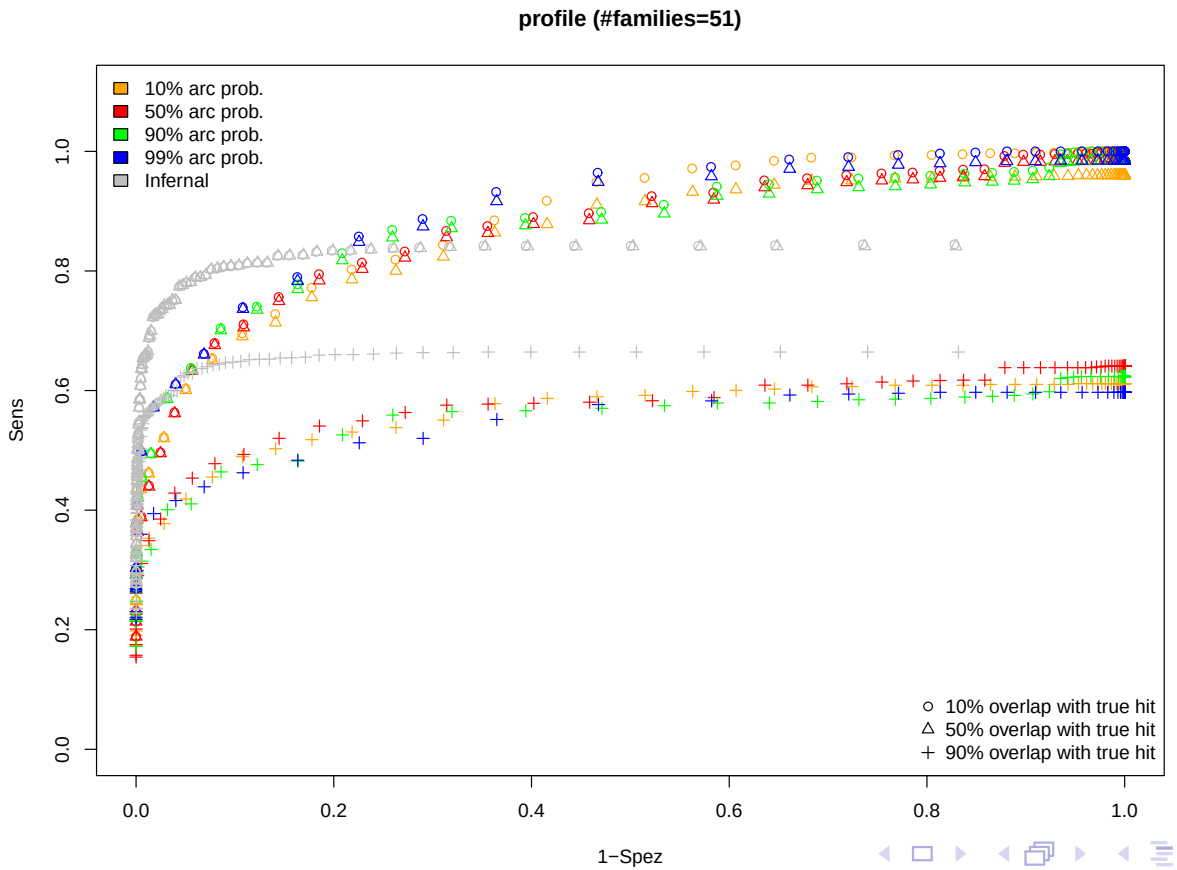
Comparison single vs. profile



- Locarna-Scan discovers more hidden RNAs
- Profile queries perform better
- higher arc probabilities increase scanning performance

ROC curves

- curves of all families are combined (averaged ROC-curves)
- a prediction is called *hit* when at least 50 or 90% of the region overlaps the hidden sequence, and vice versa!

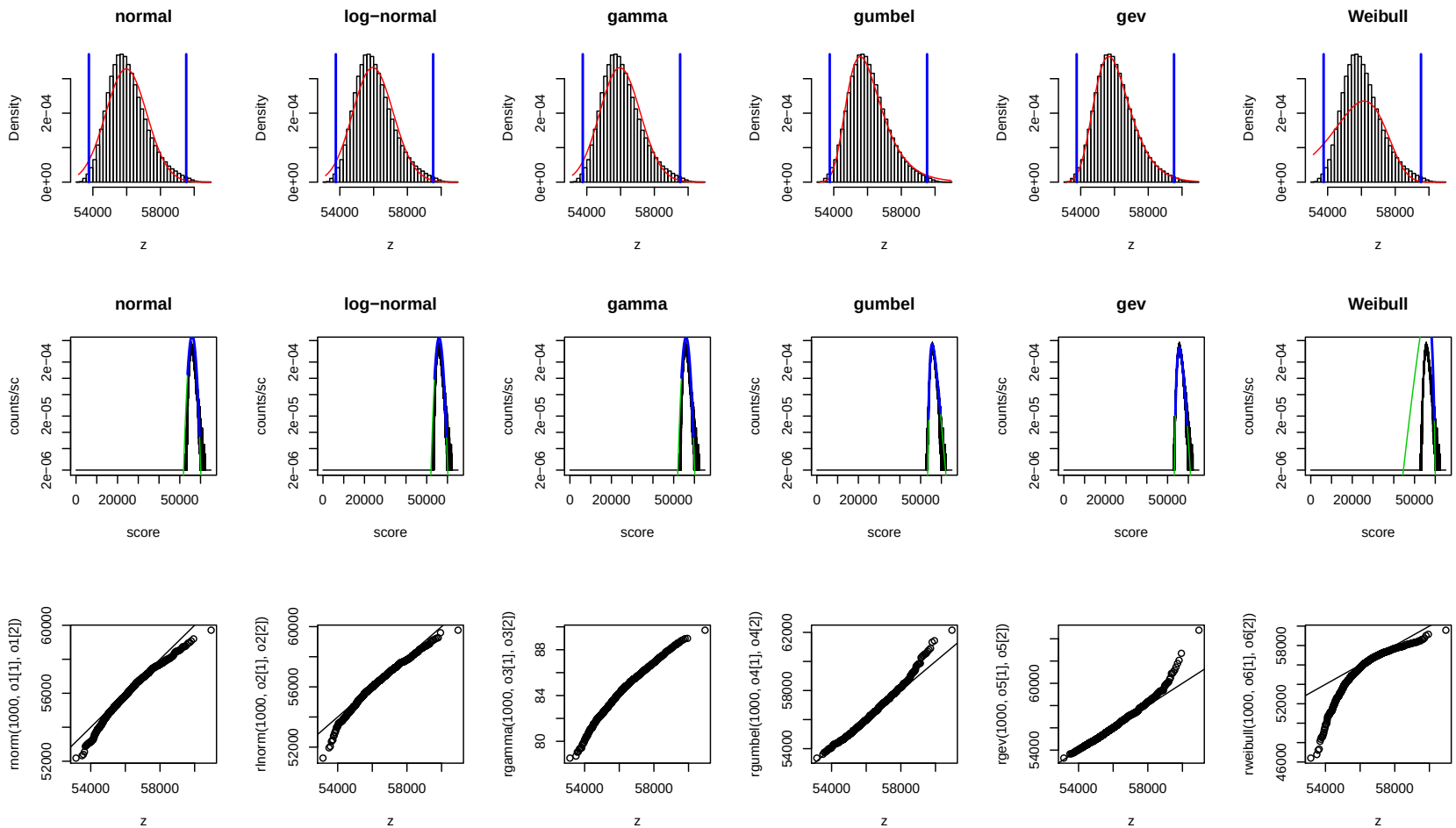


Expectation values

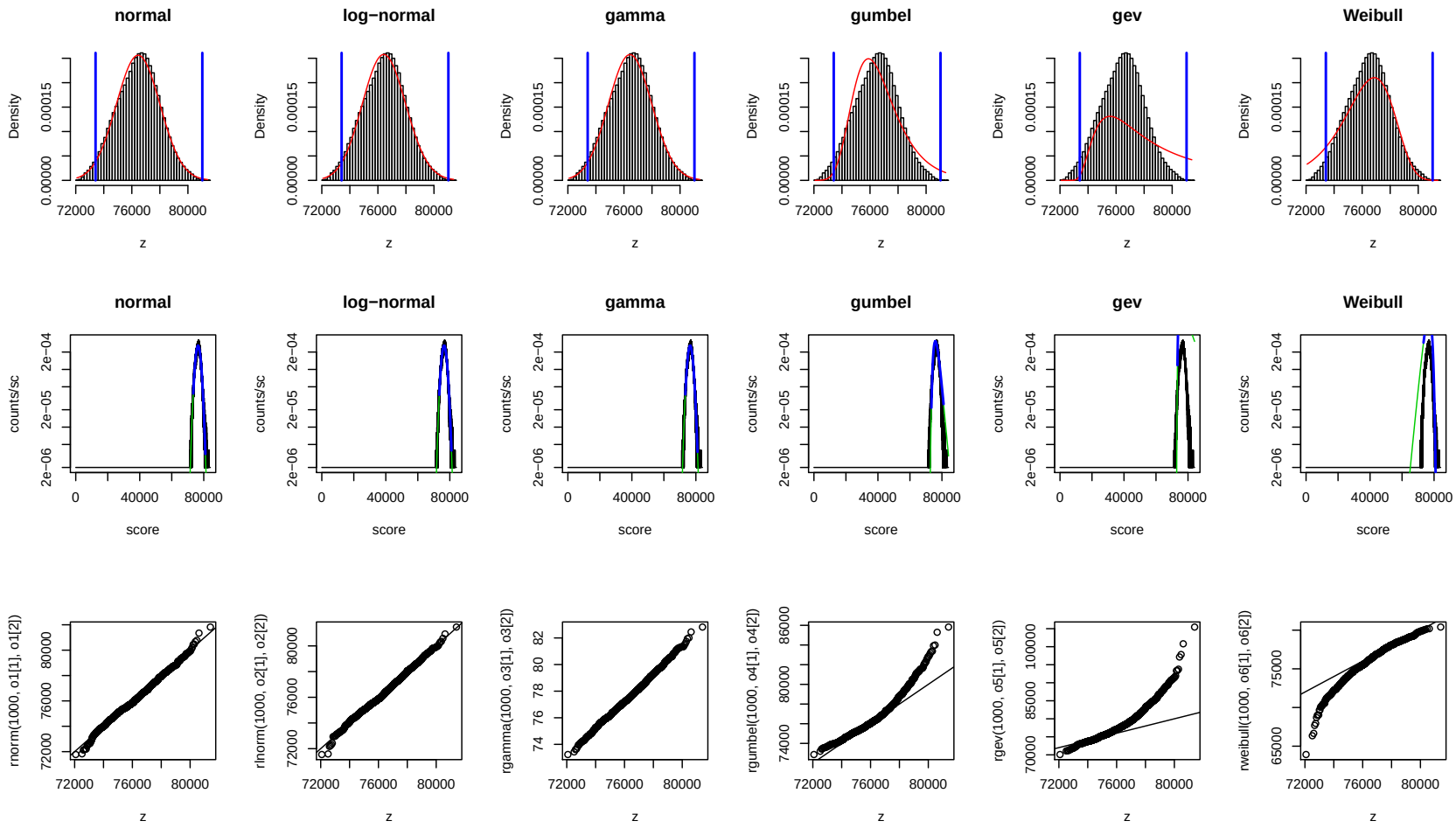
ROC-curves showed worse sensitivity compared to Infernal $\longrightarrow$ Infernal and RSearch use e-value ranked predictions

Expectation values

ROC-curves showed worse sensitivity compared to Infernal $\longrightarrow$
Infernal and RSearch use e-value ranked predictions
Finding a probability distribution to approximate our observed score distribution showed to be quite difficult!



Distributions approximating one RNA family . . .



. . . often failed for other families

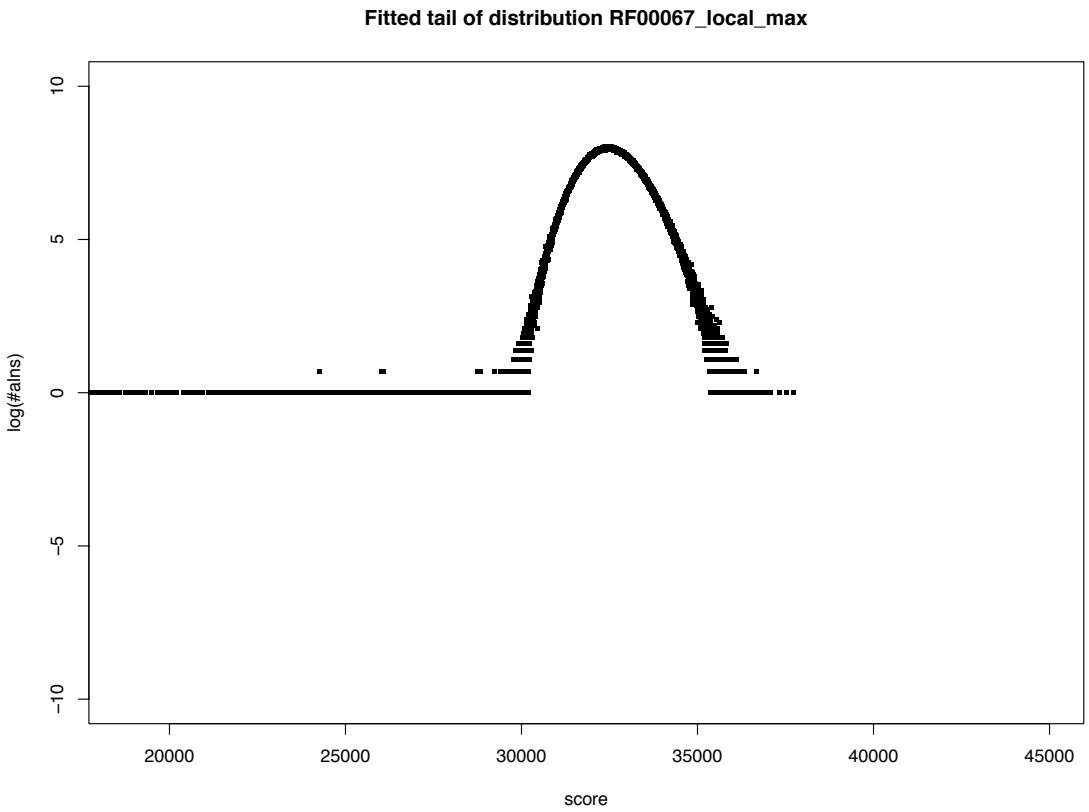
better solution

finally Jana showed us a better idea during the *Herbstseminar*.

better solution

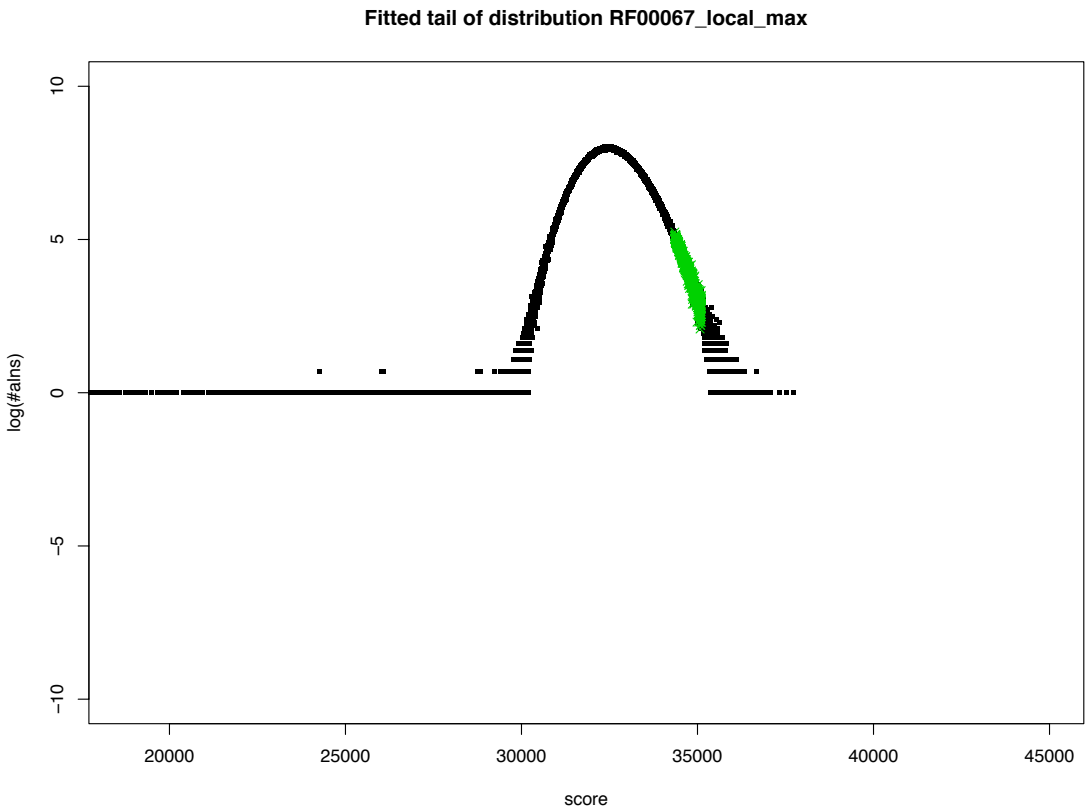
finally Jana showed us a better idea during the *Herbstseminar*. idea is not to fit against a certain probability function, but to approximate the score distribution slope using a polynomial

score fitting



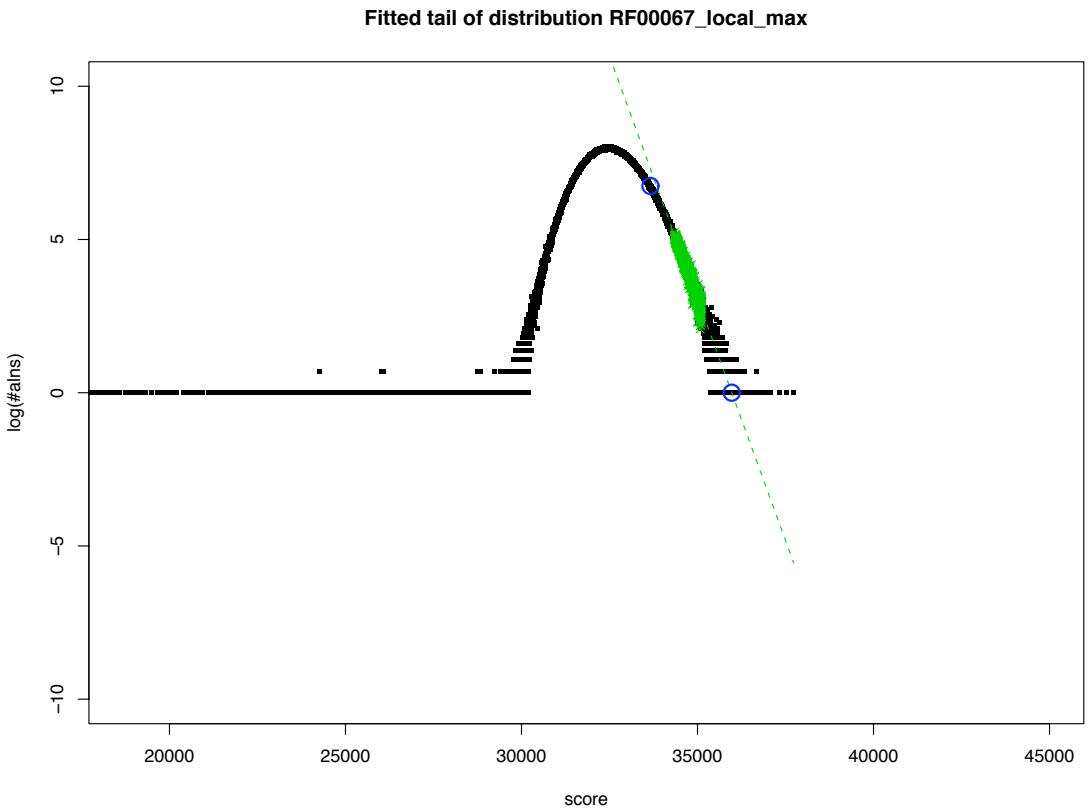
- converting distribution to log-scale

score fitting



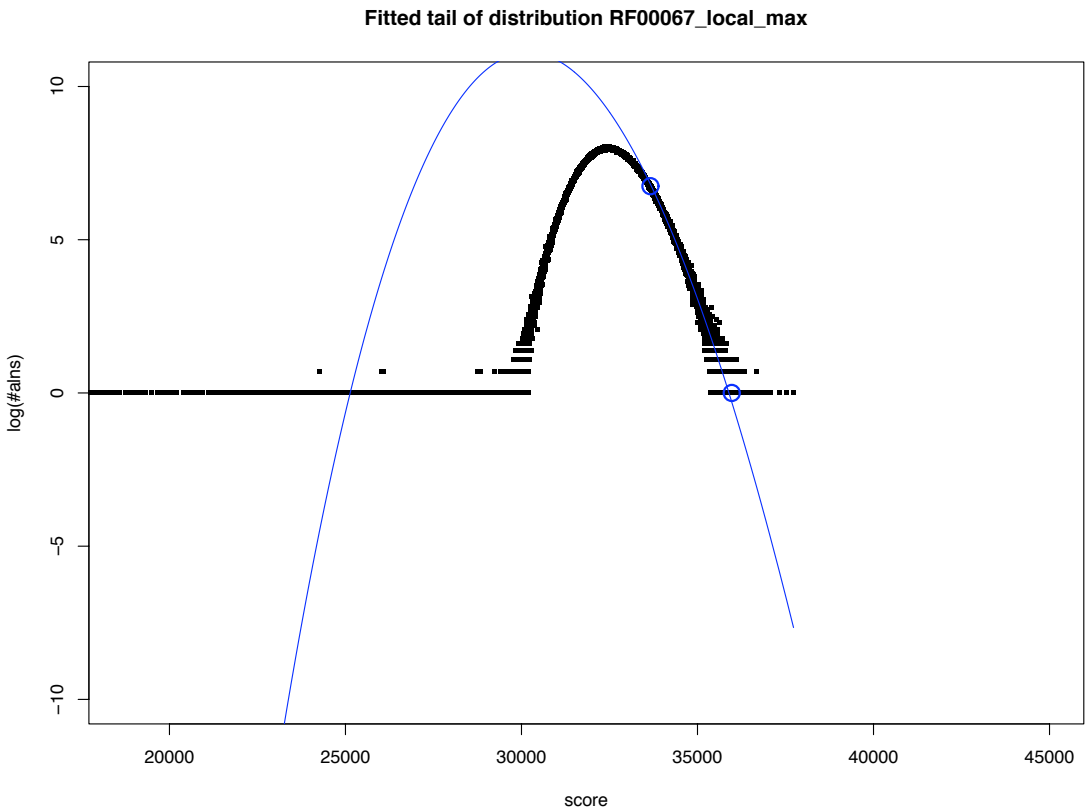
- converting distribution to log-scale
- select a subset of the slope

score fitting



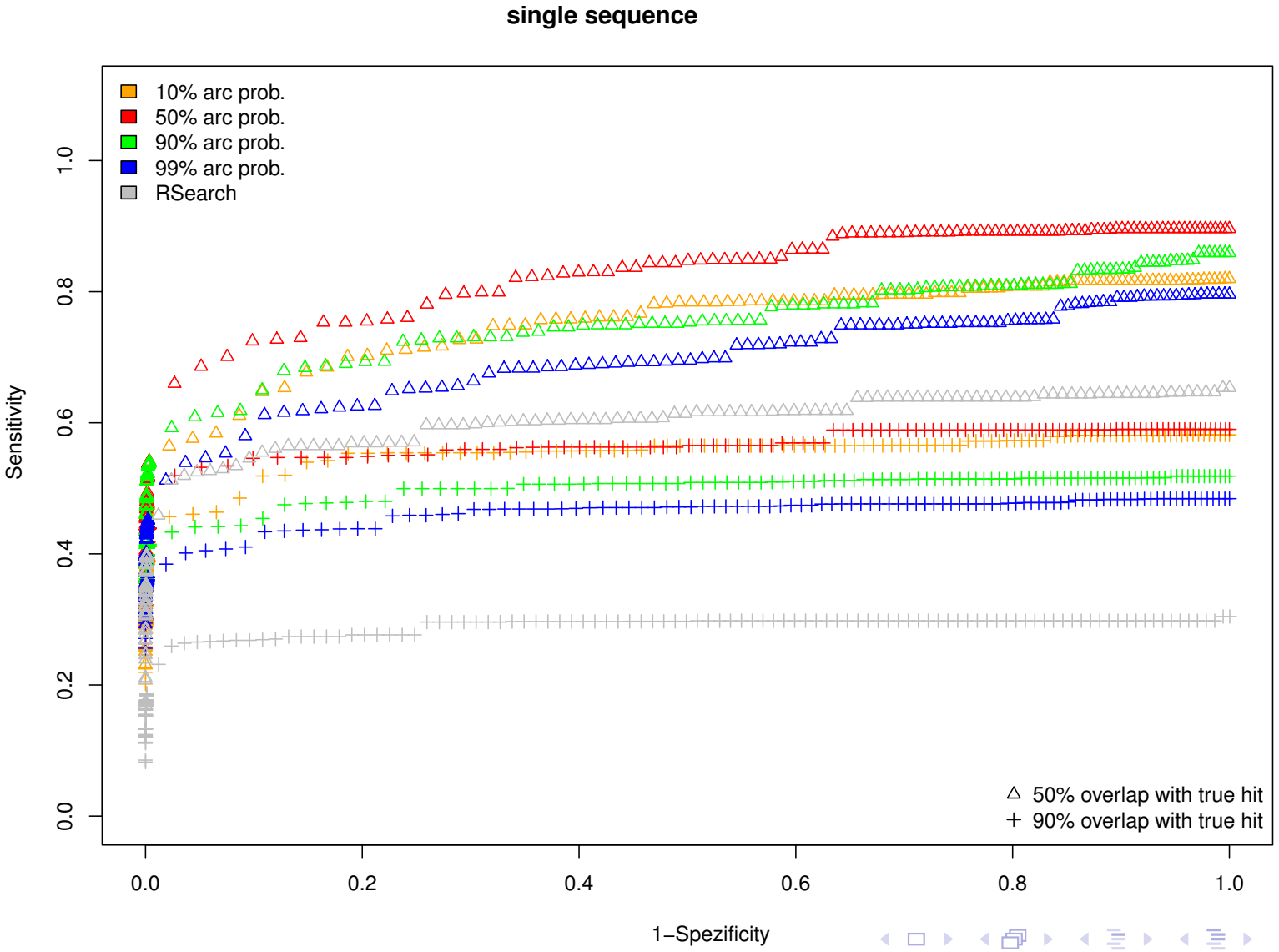
- converting distribution to log-scale
- select a subset of the slope
- fit a tangent along this subset $\longrightarrow$ intersections determine score fitting range

score fitting

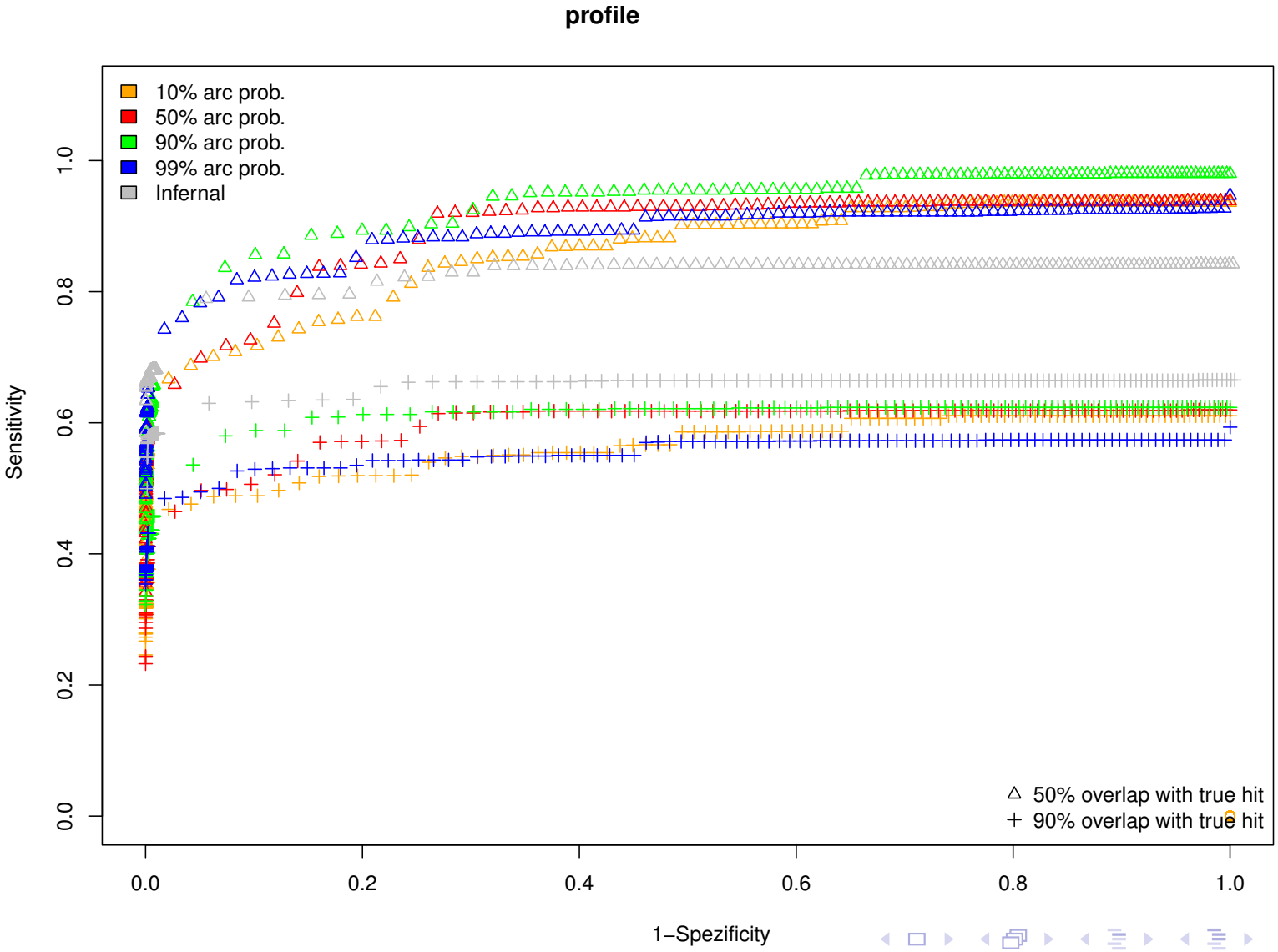


- converting distribution to log-scale
- select a subset of the slope
- fit a tangent along this subset → intersections determine score fitting range
- calculate a polynomial distribution through these scores → deviation from this distribution determines e-values

ROC curves - single query



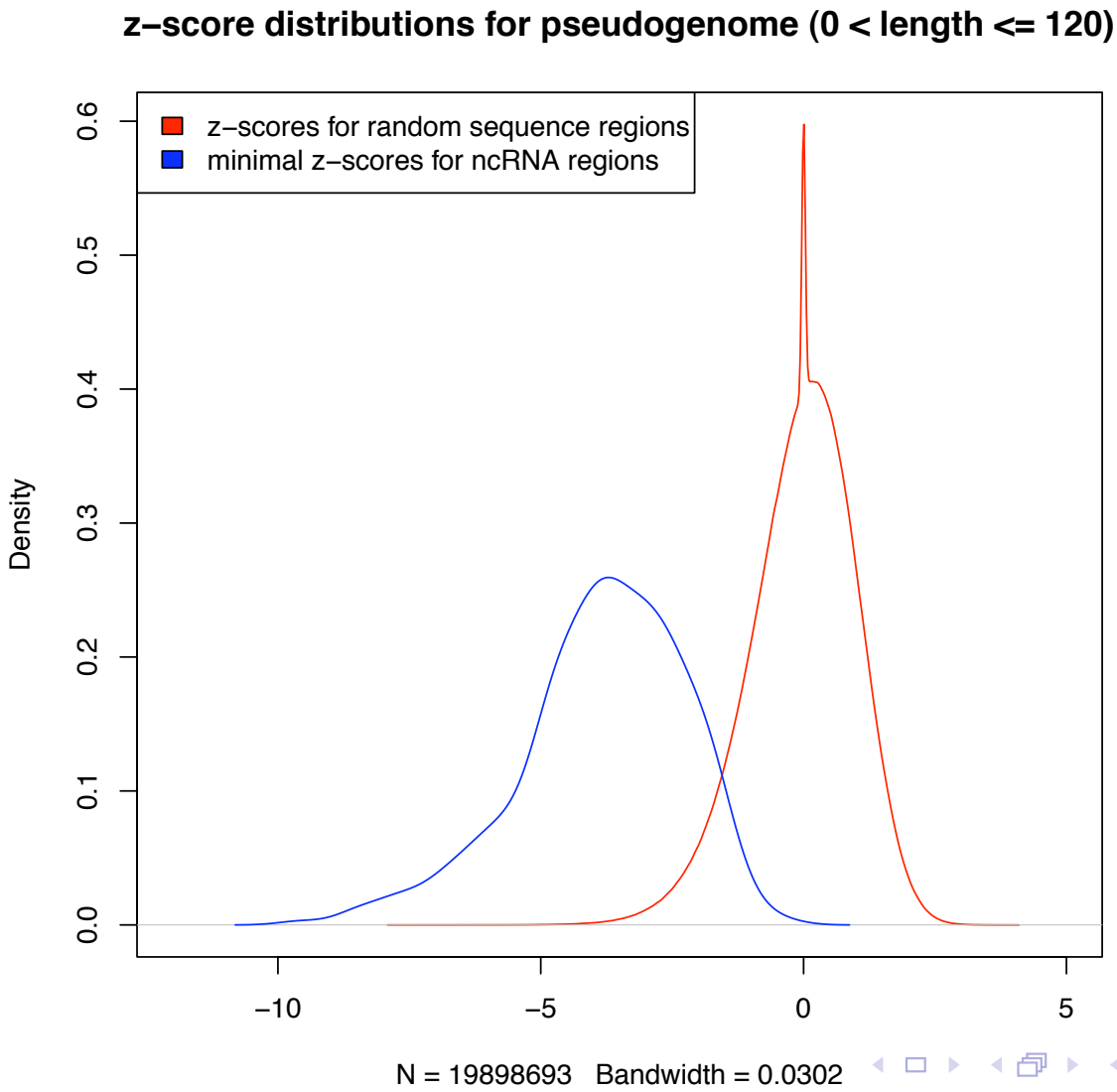
ROC curves - profile query



Z-Scores

- give an estimation about the deviation of the Gibbs free energy for a structure within the window from the expected energy for structures from random sequences of that size and base composition
- are calculated for each window of a certain size along the target sequence (using RNAplfold2 and GetZofPL)
- Z-scores below zero indicate better energetics

Z-Scores


z–score distributions for pseudogenome (0 < length <= 120)
z–scores for random sequence regions
minimal z–scores for ncRNA regions
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
–10
–5
0
5
N = 19898693 Bandwidth = 0.0302

problems

- Our current problem is how to integrate z-scores into our alignment scores
- $\longrightarrow$ simply removing local maxima whose z-score > 0 also reduced sensitivity

problems

- Our current problem is how to integrate z-scores into our alignment scores
- $\longrightarrow$ simply removing local maxima whose z-score > 0 also reduced sensitivity
- If only few local maxima remained after filtering, the score distribution fitting failes $\longrightarrow$ edges of the polynomial bend upwards $\longrightarrow$ "better" scores get higher e-values

problems

- Our current problem is how to integrate z-scores into our alignment scores
- $\longrightarrow$ simply removing local maxima whose z-score > 0 also reduced sensitivity
- If only few local maxima remained after filtering, the score distribution fitting failes $\longrightarrow$ edges of the polynomial bend upwards $\longrightarrow$ "better" scores get higher e-values
- Time!

Thank you for your attention!

Special thanks goes to Kristin Reiche, Jana Hertel, Stephan Bernhart, Sven Findeiß and Steffen Heyne

Latest version:

svn co `https://yaseto.svn.sourceforge.net/svnroot/yaseto`

michael_siebauer@eva.mpg.de