Statistics for matching runs

- Statistics of matching runs:
  \[ E(l) \approx mn^{l} \]
- Length versus score?
  - Consider all mismatches receive a negative score of \(-\infty\) and
    \(a_i b_j\) match receives a positive score of \(s_{i,j}\).
- What is the expected number of matching runs with a score \(x\) or higher?
  \[ E(S \geq x) \propto mn^{x} \]
  - Using this theory of matching runs, Karlin and Altschul
    developed a theory for statistics of local alignments
    without gaps (extended this theory to allow for mismatches).

Statistics of local alignments
without gaps

\[ E(S \geq x) = Kmne^{-\lambda x} \]

- \(K < 1\) is a proportionality constant that corrects the \(mn\) “space
factor” for the fact that there are not really \(mn\) independent
places that could have produced score \(S \geq x\).
- \(K\) has little effect on the statistical significance of a similarity
score
- \(\lambda\) is closely related to the scoring matrix used and it takes into
account that the scoring matrices do not contain actual
probabilities of co-occurrence, but instead a scaled version
of those values. To understand how \(\lambda\) is computed, we have to
revisit the construction of scoring matrices.

Scoring Matrices

- In 1970s there were few protein sequences available. Dayhoff used
a limited set of families of protein sequences multiply aligned
inferred mutation likelihoods.

Example

- If \(M\) occurs in the sequences with 0.01
frequency and \(L\) occurs with 0.1 frequency. By
random pairing, you expect 0.001 amino acid
pairs to be \(M-L\). If the observed frequency of
\(M-L\) is actually 0.003, score of matching \(M-L\)
will be
  \[ -\log_2(3) = 1.585 \text{ bits} \]
  \[ -\log_e(3) = \ln(3) = 1.1 \text{ nats} \]
- Since, scoring matrices are usually provided
as integer matrices, these values are scaled
by a constant factor. \(\lambda\) is approximately
the inverse of the original scaling factor.
How to compute \( \lambda \)

- Recall that:
  \[
  \lambda_{ij} = \log\left(\frac{q_{ij}}{p_ip_j}\right)
  \]
  \[
  \Rightarrow q_{ij} = p_ip_j e^{\lambda_{ij}}
  \]

and:
\[
\sum_{i,j} q_{ij} = 1 \quad \text{Sum of observed frequencies is 1.}
\]

\[
\Rightarrow \sum_{p=i}^{j} \sum_{p=q}=1 \quad \text{Given the frequencies of individual amino acids and the scores in the matrix, } \lambda \text{ can be estimated.}
\]

Extreme value distribution

- Consider an experiment that obtains the maximum value of locally aligning a random string with query string (without gaps). Repeat with another random string and so on. Plot the distribution of these maximum values.
- The resulting distribution is an extreme value distribution, called a **Gumbel distribution**.

Normal vs. Extreme Value Distribution

![Diagram showing Normal and Extreme Value distributions](image)

- Normal distribution: \( y = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \)
- Extreme value distribution: \( y = e^{-x} - e^{-x} \)

Local alignments with gaps

- The EVD distribution is not always observed. Theory of local alignments with gaps is not well studied as in without gaps. Mostly empirical results. For example, BLAST allows only a certain range of gap penalties.

BLAST statistics

- Pre-computed \( \lambda \) and K values for different scoring matrices and gap penalties are used for faster computation.
- Raw score is converted to bit score:
  \[
  S_{\text{bit}} = \frac{\lambda S - \ln K}{\ln 2}
  \]
- E-value is computed using
  \[
  E = \text{srs} \cdot 2^{-S_{\text{bit}}}
  \]
  \[
  \text{srs} = (m-L)(n-N\cdot L)
  \]
- \( m \) is query size, \( n \) is database size and \( L \) is the typical length of maximal scoring alignment.

FASTA Statistics

- FASTA tries to estimate the probability distribution of alignments for every query.
- For any query sequence, a large collection of scores is gathered during the search of the database.
- They estimate the parameters of the EVD distribution based on the histogram of scores.
- Advantages:
  - reliable statistics for different parameters
  - different databases, different gap penalties, different scoring matrices, queries with different amino acid compositions.
Statistical significance
another example

• Suppose, we have a huge graph with weighted edges and I want to find strongly connected clusters of nodes.
• Suppose, an algorithm for this task is given.
• The algorithms gives you the best hundred clusters in this graph.
• How do you define best?
• Cluster size?
• Total weight of edges?

Statistical significance

• How different is a found cluster of size N from a random cluster of the same size?
• This measure will enable comparison of clusters of different sizes.

Statistical significance of a cluster

• Use maximum spanning tree weight of a cluster as a quantitative representation of that cluster.
• And see what values random clusters get. (sample many random clusters)

Statistical significance of a cluster

Looks like an exponential decay. We may fit an exponential distribution on this histogram.

\[ y = \lambda e^{-\lambda x} \]

Fitting an exponential

\[ y = \lambda e^{-\lambda x} \]

Statistical significance of a cluster

After we fit an exponential distribution, we compute the probability that another random cluster gets a higher score than the score of found cluster.

\[ P(x \geq w) = e^{-\lambda w} \]
Examples

• $\lambda_5 = 1.7$ for clusters of size 5 and $\lambda_{20} = 0.36$ for clusters of size 20.
• Suppose you have found a cluster of size 5 with weights of its edges sum up to 15 and you have found a cluster of size 20 with weight 45 which one would you prefer?

\[
P(x \geq 15) = e^{-\lambda_{15}} = 8.42 \times 10^{-12}
\]
\[
P(x \geq 45) = e^{-\lambda_{45}} = 9.21 \times 10^{-8}
\]