Supplemental Material

Search and clustering orders of magnitude faster than BLAST

Robert C. Edgar
robert@drive5.com
http://www.drive5.com/usearch

UBLAST and USEARCH algorithms

Let T_1, T_2 ... be the database sequences (targets), Q be the query sequence and w be the word size (by default, w=8 for nucleotides and w=5 for amino acids). Let $W(S)$ be the set of words of length w in sequence S and $sim(S,T)$ be the similarity of sequences S and T. Similarity is defined either as fractional identity computed as the number of identical letters in a global alignment divided by the length of the shorter sequence or as the E-value computed from a local alignment. A target meeting or exceeding a pre-set similarity threshold (t) is an *accept*, a failed match is a *reject*. Targets are compared to Q in order of decreasing $U_i = |W(Q) \bigcap W(T_i)|$, i.e. the number of unique words in common between Q and the ith database sequence. Since U correlates well with similarity, (i) if an accept exists, one is likely to be found in the first few targets tested, (ii) the first accept is likely to have the highest possible similarity, or close to it, and (iii) the probability that a high-similarity hit exists in the database decreases rapidly with the number of failed attempts. A search is terminated after a pre-determined number of accepts (a, default a=1) or rejects (r, default r=8) have occurred. Increasing r tends to improve sensitivity at the cost of slower execution time, increasing a allows multiple hits to be reported for a given query and increases the probability of finding the best possible accept. Words are extracted from the query at intervals of μ letters (the *step size*), so μ=1 yields all overlapping words and with μ=w a complete, non-overlapping coverage is obtained. As μ increases, U can be computed more quickly, but sensitivity tends to be reduced due to reduced correlation between U_i and $sim(T_i,Q)$. This reduced correlation tends to increase the number of rejections before a match is found, to reduce the probability that an accept is found if one is present in the database, and to reduce the probability that the best possible accept is found first. The choice of μ is made given e, a user-specified parameter giving the desired expected value of U for a target with similarity t. The value of μ given e is estimated with the help of tables derived from empirical data as biological sequences tend to have highly constrained regions, causing common word counts to be higher than theoretical lower bounds for a given similarity. By default e=8, meaning that the step size is chosen so that an accept is expected to have at least eight words in common with the subset extracted from query.

Explicit sequence comparisons start by finding gapless high-scoring segment pairs (HSPs). In the case of UBLAST, exact word matches of length *k* (default *k*=3 for amino acids, *k*=4 for nucleotides) are extended using a BLAST-like algorithm with X-drop termination. In the case of USEARCH, HSPs are identified using spaced pairs of matching words of length *k*. The similarity of the HSPs is computed, if it is <*t* then the target is rejected without further processing as any remaining regions will almost always have lower similarity. Otherwise, remaining regions are aligned using banded dynamic programming (Chao, et al., 1992) and similarity is computed from the final alignment. E-values are computed for HSPs using Karlin-Altschul statistics (Karlin and Altschul, 1990).

UCLUST algorithm

The UCLUST algorithm employs USEARCH to seek a matching cluster for a given sequence. Many variants of this approach are possible; the most computationally efficient is described here. The high-level strategy follows that of ICATools (Parsons, 1995). An initially empty database is created, which is extended as input sequences are processed. The database contains exactly one representative sequence for each cluster, known as its *seed*. Each input sequence (*Q*) is compared to the current database using USEARCH. If a matching seed is found, *Q* is assigned to the corresponding cluster and the database remains unchanged; otherwise, *Q* is added to the database, becoming the seed of a new cluster. A match is defined as a global alignment of the query to a seed with an identity exceeding a pre-set threshold. Using a single representative sequence per cluster minimizes the database size and total number of sequence comparisons and hence the time and memory required, but may not be biologically optimal as there is no enforced lower bound on pair-wise similarity when neither sequence is a seed. CD-HIT uses the same clustering criteria (match to a single representative sequence per cluster) and a similar definition of identity to UCLUST, allowing a direct comparison between the two programs. As UCLUST processes sequences in input file order, they should be sorted so that the most appropriate seed sequence for a cluster tends to be placed before other members. CD-HIT processes sequences in order of decreasing length. In the clustering experiments reported in the main manuscript, sequences were therefore sorted by decreasing length as a pre-processing step before running UCLUST in order to facilitate a direct comparison.

Program options

Default options were used for all programs, with the following exceptions. The --maxtargets option of UBLAST was set to 1000 for Pfam and 100 for Rfam. The similarity threshold for USEARCH was specified as --id 0 --evalue 1.0 for Pfam and --id 0.0 for Rfam. For BLASTN, BLASTP and MEGABLAST, the -outfmt 6 option was used. The -M option of CD-HIT was used to indicate the

amount of available memory (2 Gb). The identity threshold for clustering was specified using the -c option of CD-HIT and the --id option for UCLUST.

References

Chao, K.M., Pearson, W.R. and Miller, W. (1992) Aligning two sequences within a specified diagonal band, *Comput Appl Biosci*, **8**, 481-487.

Karlin, S. and Altschul, S.F. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes, *Proc Natl Acad Sci U S A*, **87**, 2264-2268.

Parsons, J.D. (1995) Improved tools for DNA comparison and clustering, *Comput Appl Biosci*, **11**, 603-613.