lastz

In order to generate 3' UTR alignments across primate genomes, the following procedure was used:

1. The full list of human protein coding genes (spliced mRNA complete with 5' UTR and 3' UTR) was downloaded from UCSC genome browser
2. This list was filtered to remove genes with no 3' UTR, genes with multi-exon 3' UTRs, and genes with the same identifier assigned to multiple locations in the genome.
3. 5 primate genomes (hg19, gorGor3, panTro3, nomLeu1, and ponAbe2) were downloaded from the UCSC genome browser
4. Lastz was run with each primate genome (including humans) as the query, the set of filtered human genes as the subject, a match score of 1, mismatch score of 3, and a step size of 200.
5. Orthologs across primate species were called using the following criteria:
    1. Lastz hits with less than 95% identity to any human gene were discarded.
    2. Lastz hits to a single human gene were sorted with respect to where they fell on the human gene and with respect to where they fell relative to each other, and those which fell in a linear sequence (with respect to both the human gene and the query genome) were grouped together as putative transcripts. At this stage, many individual human genes had multiple putative transcripts, which were sorted by percentage of the human gene covered by the transcript.
    3. Cases in which the best 'transcript' covered less than 80% of the human gene were discarded.
    4. Of the human genes with a best hit transcript covering more than 80% of the human gene, cases in which a second best hit transcript was also present covering more than 20% of the best hit transcript were also discarded as putatively duplicated genes.
    5. Of the remaining genes, cases in which more than 20% of the nucleotides covered the human gene redundantly were also discarded. to avoid genes which have undergone excessive internal duplications.
    6. Of the genes remaining at this stage, those in which any of the putative transcript covered an annotated human 3' UTR more than once were discarded, to ensure proper alignment of 3' UTRs.
6. After calling orthologs using the above procedure within individual primate genomes, the lists of orthologs were cross referenced, and genes with high confidence nonduplicated orthologs present in all primate species were kept.
7. Using the 11way primate multiz alignment maf file for the gorilla genome, maf alignments with overlap to 3' UTRs with non-redundant genome coverage in any species were truncated to match the boundaries of 3' UTRs and assembled into complete 3' UTR alignments (with an N inserted in cases involving 3' UTRs composed of multiple maf alignments).

sfs_code

The parameters used for sfs code were based on the following estimates and assumptions:

- 15 year generation times for all primates
- A scaled theta value of 0.00002 to reproduce observed levels of sequence divergence between human and chimpanzee
- Divergence times (and therefore split times) based on timetree
- A simulated UTR space of 30,000 nucleotides

These values were used to produce parameters for sfs_code using the following table:

Using this table and the above estimates, the final sfs_code command was as follows:

sfs_code 9 1 -t 0.00002 -L 1 30000 -TS 0 0 1 -TE 0 0 -TS 0 1 2 -TS 313.333 2 3 -TE 313.333 2 -TS 313.333 3 4 -TS 773.333 4 5 -TE 773.333 4 -TS 773.333 5 6 -TS 940 6 7 -TE 940 6 -TS 940 7 8 -TE 1360

sfs output was parsed into fasta format, assuming that mutations present in greater than half of any population would have been sampled if a reference genome of that species had been produced.

probability calculations for non-canonical miRNAs:

While most miRNAs begin with a 'U' at nucleotide one of the mature miRNA, there are 32 well-conserved miRNAs from our study that do not follow this tendency. We used our dataset to evaluate whether in these cases an 'A' is preferred opposite the first nucleotide, or whether the reverse complement of this first nucleotide is preferred, and used turnover ranks as a readout for reverse complement vs. 'A' preference. We used the following steps to test this theory:

(1) We summed the gain rank and loss rank associated with each miRNA binding site to assign an overall 'turnover rank' to each non-canonical miRNA binding site.
(2) To assign a probability of a reduced turnover rank occurring by chance, we considered that an eightmer ending in 'A' could have one of 25 gain ranks and one of 25 loss ranks, for 25^2 possible rank combinations, and enumerated all such combinations that would lead to an improved overall turnover rank relative to the turnover rank of the unmodified eightmer. This gave each of our 32 miRNAs its own probability of a reduced turnover rank occurring by chance.
(3) We counted the number of miRNAs that had a reduced turnover rank after replacing the reverse complement of the first nucleotide with an 'A' and observed that 21 out of 32 miRNA binding sites exhibited reduced turnover ranks when this substitution was made, suggesting that the reverse complement of nucleotide '1' may not be as strongly constrained as an 'A' opposite this nucleotide.
(4) To assign significance, we modeled a reduced turnover rank as a success (1), and a non-reduced turnover rank as a failure (0). Each possible outcome could therefore be converted into a string. For example, successes at the 6^{th} and 22^{nd} input eightmers could be represented as follows:

00000100000000000000010000000000

We next matched the position of each '0' or '1' to its corresponding probability of success (or the additive inverse of the success probability for a failure) from the list of probabilities calculated in (2) and took the product of these probabilities for the cumulative probability of a particular combination of successes and failures.

By repeating this process over all strings with more than 21 1's, and summing the resulting probabilities, we obtained a probability of 21 or more 'successes' occurring by chance. This process can be viewed as a joint distribution of binomial distributions, where each binomial distribution consists of a single trial with a different underlying probability. In practice, we used cumulative probabilities superimposed on taxicab geometry with a dynamic programming approach to dramatically reduce the computation of these probabilities (see the script ‘multiple_binomial_cumulative_dist.py’ for the implementation)