[Re] Local alignment statistics - Introduction

Nathan Brouwer

11/6/2019

Introduction

The code below can be used to replicate Table 1 on page 465 of Altschul and Gish (1996) "Local Alignment Statistics." This table is the result of a **computational experiment** to understand the statistical basis for **E values** and **P values** of **un-gapped local alignments** like those used in **BLAST searches**.

The goal of this project is primarily to replicate this table, while also explaining key steps in the process. For additional information on the stistical concepts discussed in Altschul and Gish (1996) see "The Statistics of Sequence Similarity Scores" at https://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html

Each row of Table 1 represents the result of 10,000 simulations. I refer to each row of the table as a "subexperiment" (Though this terminology is not used by Altschul and Gish). During each **iteration** of a subexperiment, two separate but equal length random sequences of amino acids are generated. The sequences are then run through a local alignment algorithm and scored using a BLOSUM 62 scoring matrix. The score (s) and length of the alignment was then stored. Once all 10,000 iterations were finished the mean length of the alignments (l) was calculated. The distribution of scores was then analyzed using statistical methods which can estimate the parameters **mu** and **lambda** of the Gumbel **extreme value distribution (EVD)**. The parameters mu and lambda were then plugged into the following equation to solve for the parameter K:

Plain text version of equation:

mu = (ln(m x n x K))/lambda

Rendered version of equation.

$$\mu = \frac{\ln(m * n * K)}{\lambda}$$

The scripts below takes a very basic and inefficient approach to replicating this table.

1. **Part 1** reproduces the original table.

2. **Part 2** reproduces the regression results discussed in the paper
3. **Part 3** sets up the amino acid frequencies from which random sequences are produced
4. **Part 4** introduces how to simulate random polypeptides using the sample() function
5. **Part 5** of the script outlines a basic workflow for a single iteration of the experiment (a single alignment).
6. **Part 6** demonstrates how to use a for() to carry out the necessary computations for a full run of simulations necessary to create one row of the table (one subexperiment).
7. **Part 6** demonstrates how to calculate the numbers that appears in the table, using the first row as an example.

8. **Part 7** contains additional copies of the code to run the computations for *all* of the lines of the table.
   Part 5 processes the data produced by Part 4 to calculate mu and lambda.
   Part 6 compiles the final table and calculates K.

Other scripts in this project folder take a more advanced approach to this problem.

To do

check that aa 1 letter codes are correct :) add section introducing seq alignment