Algorithm of Designing Intron

1 Introduction

We adapt a previous algorithm by Perutka et al. (2014) to design our intron.

In this algorithm, a training set from the database, which contains the information of appearance frequency of nucleotide of each type in all positions, is used to score all 45-bp subsequences in the DNA and find the potential insert position. After that, the corresponding primers are designed to modify the intron.

2 Algorithm

To begin with, the probability that a 45-bp sequence, denoted by X, is a target sequence is:

$$\mathbb{P}(T|X) = \frac{\mathbb{P}(X|T)\mathbb{P}(T)}{\mathbb{P}(X)} = \frac{(\prod_p f_{i(p)})\mathbb{P}(T)}{\prod_p F_{i(p)}} \tag{1}$$

where $\mathbb{P}(X)$ is the probability that a random 45-bp sequence is X, $\mathbb{P}(T)$ is the probability that a random 45-bp sequence is a target sequence, $f_i(p)$ is the frequency that in the training set the position p has the same nucleotide $i(p)$ with X, $F_{i(p)}$ is the frequency that in a random 45-bp sequence the position p has the same nucleotide $i(p)$ with X and $\prod_p$ means multiply for all 45 positions in the 45-bp sequence. The $F_{i(p)}$ is calculated using the nucleotide frequencies in *E.coli* genome: $F_G = 0.2536$, $F_A = 0.2462$, $F_T = 0.2456$, $F_C = 0.2542$.

Formula (1) cannot be directly used to score a sequence because $\mathbb{P}(T)$ is unknown. However, $\mathbb{P}(T)$ is a constant, so it could be ignored when comparing with different sequences. Thus a log-odd score S is calculated as:

$$S = \log_2 \frac{\prod_p f_{i(p)}}{\prod_p F_i(p)} = \sum_p \log_2 \frac{f_{i(p)}}{F_{i(p)}} \tag{2}$$

The next step in previous algorithm is to calculate a E-value which is the probability that a random 45-bp sequence would have a S score more than we calculated for sequence X. In the paper by Perutka et al.(2014), the threshold set for E-value is 0.6, which would mean that a randomly generated DNA sequence of n bp would have approximately $0.6n$ potential target sequence, which is not practical here since the number is too big. Instead, we simply compare the S score and select the top 0.2% 45-bp (including both sense-strand and antisense-strand) sequences as our potential target sequence and design introns for them.

Lastly, we use Watson-Crick base pairs to design the primers to modify the intron donor plasmid's IBS1/2, EBS1/δ, and EBS2 sequences.

3 MATLAB code

Here we provide the MATLAB code that realizes the algorithm.

```
%function for Watson-Crick base-pair.
function [newSeq] = pair(seq)
syms A T C G
for k=1:length(seq)
    if seq(k) == A
        seq(k) = T;
    elseif seq(k) == G
        seq(k) = C;
    elseif seq(k) == C
       seq(k) = G;
    elseif seq(k) == T
        seq(k) = A;
    end
end
newSeq=seq;
end

%function for calculating S score
function [s] = seqScore(seq,trainSet)
syms A T C G
fg=0.2536;fa=0.2462;ft=0.2459;fc=0.2542;
s=1;
```

```
for k=1:length(seq)
    if seq(k) == A
        s = s*trainSet(2,k)/fa;
    elseif seq(k) == G
        s = s*trainSet(1,k)/fg;
    elseif seq(k) == C
        s = s*trainSet(4,k)/fc;
    elseif seq(k) == T
        s = s*trainSet(3,k)/ft;
    end
end
s=log2(s);
end

%function for designing primers
function primer(seq)
syms A T C G
disp(seq) %show the 45-bp
%non-modified primers
IBS1 = [A,A,A,A,A,A,G,C,T,T,A,T,A,A,T,T,A,T,C,C,T,T,A,G,A,...
A,A,T,C,C,T,C,G,T,C,G,T,G,C,G,C,C,C,A,G,A,T,A,G,G,G,T,G];
EBS1 = [C,A,G,A,T,T,G,T,A,C,A,A,A,T,G,T,G,G,T,G,A,T,A,A,C,A,G,A,...
T,A,A,G,T,C,C,T,C,G,T,C,C,T,T,A,A,C,T,T,A,C,C,T,T,T,C,T,T,T,G,T];
EBS2 = [T,G,A,A,C,G,C,A,A,G,T,T,T,C,T,A,A,T,T,T,C,G,G,T,...
T,A,T,T,T,C,T,C,G,A,T,A,G,A,G,G,A,A,A,G,T,G,T,C,T];
IBS1(24:28)=seq(19:23);
IBS1(30:35)=seq(25:30);
disp('IBS1/2');disp(IBS1)
EBS1(35:42)=seq(25:32);
disp('EBS1/delta');disp(EBS1)
EBS2(26:30)=pair(seq(23:-1:19));
disp('EBS2');disp(EBS2)
end

%main code
syms A T C G
trainSet=[];
DNA=[];
```

```
selectRatio=0.002 %proportion of target secquence
%calculate S score
sScore=zeros(1,2*(length(DNA)-44));
for k=1:(length(DNA)-44)
    seq=DNA(k:k+44);
    sS(k)=seqScore(seq,trainSet);
end
antiDNA=pair(DNA);
for k=1:(length(DNA)-44)
    seq=antiDNA(k:k+44);
    sS(k+length(DNA)-44)=seqScore(seq,trainSet);
end
[score,position]=sort(sS,’descend’);
%design and display primers
for k=1:floor(2*(length(DNA)-44)*selectRatio)
    if position(k)>length(DNA)-44
     disp([num2str(-position(k)+2*length(DNA)-73),’|’,...
        num2str(-position(k)+2*length(DNA)-72),’a’])
        primer(DNAs((position(k)-length(DNA)+44):...
        (position(k)+88-length(DNA))))
    else
     disp([num2str(position(k)+29),’|’,num2str(position(k)+30),’s’])
        primer(DNA(position(k):(position(k)+44)))
    end
end
```

4 Reference

Perutka, J., Wang, W., Goerlitz, D. and Lambowitz, A. (2004). Use of Computer-designed Group II Introns to Disrupt Escherichia coli DExH/D-box Protein and DNA Helicase Genes. Journal of Molecular Biology, 336(2), pp.421-439.