Finding possible sgRNAs in a specified regions

Timothy Daley

August 9, 2016

Suppose that we have a list of genes and we wish to design short guide RNAs (sgRNAs) for CRISPR knockout, activation, or inhibition (ko/a/i). The regions for each type are different. For CRISPRko we typically want sgRNAs that are located in first exon of the transcript; in CRISPRa we want to target the promoter region of the gene, typically 400 to 50 bases upstream of the transcription start site (TSS); and for CRISPRi we want to target around the TSS and slightly downstream (e.g. -50 to +300 from the TSS).

We will split up the design into the following steps:

1. From the gene names, obtain the sequences of the targeted regions.
2. Given the DNA sequence of the targeted region, find sgRNA's that are acceptable by a given set of rules.

Step 1 can be done using R and biomaRt (`https://bioconductor.org/packages/release/bioc/html/biomaRt.html`) if the rules for finding the target region are specified.

Step 2 is where the meat of the problem lies. Typically one wishes to find guide sequences that are sufficiently different from all other similar length sequences in the genome. Current tools look for sequences that are at least a prespecified edit distance (e.g. > 2) using existing tools. This ignores the varying importance of positions on the guide RNA. For example, the PAM distal bases are known to be most critical to binding specificity. Accordingly these bases are known as the seed region are defined as between 7 and 12 base pairs closest to the PAM. If all other subsequences of the genome are at least one edit distance away from the seed region, then binding will occur with few off-target effects. Ignoring this can reduce the set of possible target sequences. Our goal here is to find the set of sgRNAs within the input sequences with either no other sequences matching the seed region or at least 2 mismatches from the total sequence in a high throughput manner so that we can handle a large number of target sequences and a large reference genome.

Given input:

1. DNA sequences of target region in FASTA format,
2. Reference genome in FASTA format,

we do the following:

1. Identify all possible sgRNAs in the target regions using PAM,
2. Construct a hash table using seed regions of possible sgRNAs,

3. Hash the genome to find matches to the seed region, if there's a match then calculate edit distance, only one match means no subsequences in the genome match the seed region.

Constructing the hash table

We will use the seed regions of the possible sgRNAs as the hash values. To make hashing efficient, we want to iteratively hash the genome using the Rabin-Karp.

Let S denote the seed region. For each letter $S_i, i = 1, \ldots, |S|$ in the seed region we can map each base to a unique integer

$$\phi(S_i) = \begin{cases} 0 & \text{if } S_i = \text{A} \\ 1 & \text{if } S_i = \text{C} \\ 2 & \text{if } S_i = \text{G} \\ 3 & \text{if } S_i = \text{T}. \end{cases}$$

Each unique nucleotide sequence of length $|S|$ can be represented as a unique number

$$\phi(S) = \sum_{i=1}^{|S|} 4^{i-1} \cdot \phi(S_i) \tag{1}$$

that is between 0 and $4^{|S|}$. This number will be the hash value of the seed sequence.

Rabin-Karp shifting

Suppose we are traversing the genome G iteratively. Given a position j in the genome with sequence $G_j G_{j+1} \cdots G_{j+|S|-1}$, the corresponding hash value of this sequence is

$$\phi(G_j G_{j+1} \cdots G_{j+|S|-1}) = \sum_{i=0}^{|S|-1} 4^i \cdot \phi(G_{j+i}).$$

The next position in the genome, $j+1$ has hash value

$$\phi(G_{j+1} G_{j+2} \cdots G_{j+|S|}) = \sum_{i=0}^{|S|-1} 4^i \cdot \phi(G_{j+1+i}).$$

By definition, the last $|S|-1$ bases of the first sequence are the first $|S|-1$ bases of the second sequence. Therefore we can obtain $\phi(G_{j+1} G_{j+2} \cdots G_{j+|S|})$ from $\phi(G_j G_{j+1} \cdots G_{j+|S|-1})$ by subtracting the first base in the sequence, shifting the numerical base, and then adding the last base.

More simply,

$$\begin{aligned}
\phi(G_{j+1}G_{j+2}\cdots G_{j+|S|}) &= 4^{|S|-1}\phi(G_{j+|S|}) + \sum_{i=1}^{|S|-1} 4^{i-1}\cdot\phi(G_{j+i}) \\
&= 4^{|S|-1}\phi(G_{j+|S|}) + 4^{-1}\sum_{i=1}^{|S|-1} 4^{i}\cdot\phi(G_{j+i}) \\
&= 4^{|S|-1}\phi(G_{j+|S|}) + 4^{-1}\big(\sum_{i=0}^{|S|-1} 4^{i}\cdot\phi(G_{j+i}) - \phi(G_j)\big) \\
&= 4^{|S|-1}\phi(G_{j+|S|}) + 4^{-1}\big(\phi(G_jG_{j+1}\cdots G_{j+|S|-1}) - \phi(G_j)\big).
\end{aligned}$$

This gives a way to iteratively compute the hash values of the seed sequences. One issue is that integer division is not fast, so we would like to avoid this. (Never mind. Since we are dividing by 2^2 we can use two bit shifts to do the division and this is very efficient) One way is to define $\psi(S) = \sum_{i=1}^{|S|} 4^{|S|-i}\cdot\phi(S_i)$. In this case the hash value of the sequence $G_{j+1}G_{j+2}\cdots G_{j+|S|}$ is equal to

$$\begin{aligned}
\psi(G_{j+1}G_{j+2}\cdots G_{j+|S|}) &= \sum_{i=1}^{|S|} 4^{|S|-i}\cdot\phi(G_{j+i}) \\
&= \phi(G_{j+|S|}) + \sum_{i=1}^{|S|-1} 4^{|S|-i}\phi(G_{j+i}) \\
&= \phi(G_{j+|S|}) + 4\sum_{i=2}^{|S|} 4^{|S|-i}\phi(G_{j+i-1}) \\
&= \phi(G_{j+|S|}) + 4\big(\sum_{i=1}^{|S|} 4^{|S|-i}\phi(G_{j+i-1}) - 4^{|S|-1}\phi(G_j)\big) \\
&= \phi(G_{j+|S|}) + 4\big(\psi(G_jG_{j+1}\cdots G_{j+|S|-1}) - 4^{|S|-1}\phi(G_j)\big).
\end{aligned}$$

This recursive formula uses multiplication instead of division and should be more efficient than the previous recursion.

Note that the largest value of a size_t in C++ is $2^{64}-1$ and that the largest value of the hash value is equal to $4^{|S|}$. As long as the seed sequence is less than 32 bases long we will never run into overflow issues. And typically the seed sequence is between 6 and 12 bases long, so we expect no issues.

Now consider the reverse complement of the seed sequence. Let $\bar{S} = \bar{S}_{|S|}\bar{S}_{|S|-1}\cdots\bar{S}_1$ denote the reverse complement of the sequence S. We must also hash the seed sequence of the reverse complement of the proposed sgRNA. Since the sequence is reversed, it should be more efficient to use the first function $\phi(\bar{S})$ as the hash value.

$$
\begin{aligned}
\phi(\bar{G}_{j+|S|}\bar{G}_{j+|S|-1}\cdot\bar{G}_{j+1}) &= \sum_{i=1}^{|S|} 4^{i-1}\cdot\phi(\bar{G}_{j+|S|+1-i}) \\
&= \phi(\bar{G}_{j+|S|}) + \sum_{i=1}^{|S|-1} 4^{i}\cdot\phi(\bar{G}_{j+|S|-i}) \\
&= \phi(\bar{G}_{j+|S|}) + 4\big(\sum_{i=1}^{|S|} 4^{i-1}\cdot\phi(\bar{G}_{j+|S|-i}) - 4^{|S|-1}\phi(\bar{G}_j)\big) \\
&= \phi(\bar{G}_{j+|S|}) + 4\big(\phi(\bar{G}_j\cdots\bar{G}_{j+|S|-1}) - 4^{|S|-1}\phi(\bar{G}_j)\big)
\end{aligned}
$$