Generating Validation Library for Base Editors

CompBio coding challenge
Sasha M, Schwank Lab, 2024

In this exercise you will infer PAM preferences of a novel genome editor (e.g. identified from a metagenomic mining pipeline).

Background

Base editors are CRSIPR-based gene editing tools that allow to make single nucleotide changes in the genome. For example, adenosine base editors (ABEs) contain 1) a Cas9 enzyme that uses gRNA to find the target site and 2) a deaminase to convert A-T to G-C at that site (Fig. 1). Editing efficiency of an ABE depends on the relative position of Cas9 binding site to the target A base. In turn, positions of where a Cas9 can bind is limited by the PAM requirement. For example, *Sp*Cas9 requires an -NGGN- (N stands for any of A,C,T,G nucleotides) PAM in the vicinity of the edit site, limiting the number of mutations that can be corrected with SpCas9-based ABEs. One solution is to find natural orthologs of *Sp*Cas9 with alternative PAM requirements. Once found, such an ortholog must be characterized and validated in the lab by i.e. testing its base editing efficiency on a wide range of genomic sequences.

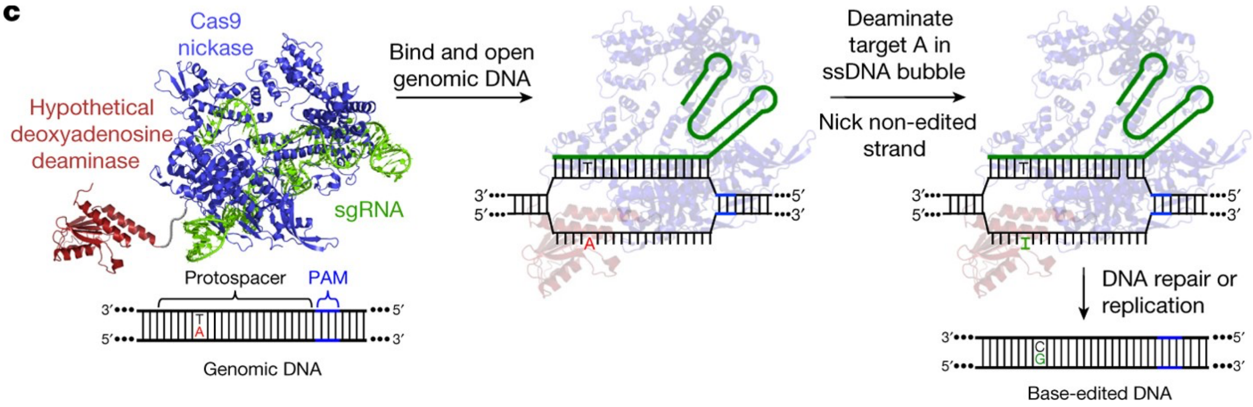

c
Cas9 nickase
Hypothetical deoxyadenosine deaminase
sgRNA
Protospacer
PAM
3′
5′
5′
3′
T
A
Genomic DNA
Bind and open genomic DNA
Deaminate target A in ssDNA bubble
Nick non-edited strand
I
DNA repair or replication
C
G
Base-edited DNA

Fig1. Schematic of the ABE. Source: https://doi.org/10.1038/nature24644

One such validation is called PAM profiling. There, ABE is tested on a library comprised of a static target site flanked with all possible 4-nt PAM combinations. After adding the ABE to the library, editing outcomes within the individual library members are assessed by deep amplicon sequencing.

Task

You need to write a script that to calculate the PAM profile of a given editor. You're given a fasta file that contains sequences of the library members. Each sequence has the following format:

```
  nnnnnnnn yyyyyytatc atgtctgctc gaagcggccg
tacctctaga gccatttgtc tcgctgaagt acaagtggta gactagnnnn ncagcatacc
tAtggtttca tccgXXXXgg ccgcttgtgg atgaatactg ccatttgtct caagatctag
ttacgccaag cttaaaaaag caccgactcg gtgccacttt ttcaagttga taacggacta
gccttatttc aacttgctat gctgtttcca gcatagctct gaaaccyyyy yynnnnnnnn
```

In the sequence above:
XXXX are the 4 randomized nucleotides corresponding to a PAM
catacctAtggtttcatccg is the 20 bp target sequence within which every A nucleotide will be potentially converted to G by an editor. *Note: for editing efficiency, we're interested in the A at position 8 (underscored).*
yyyyyy are the replicate barcodes (note: left and right barcrodes are identical in sequence)
nnnnnnnn are the sequencing primers, can be ignored for this task.

Apart from these regions, the rest of the amplicon should remain constant, modulo sequencing and sample prep errors.

For each replicate and PAM compute the A->G conversion rate. Present results as a heat map (rows == PAM[1, 2], column == PAM[3, 4]). See example below.

Input details

You're given a fastq file `library.fastq` with full-length reads.

Note: amplicons are sequenced from 3' strand you need to compute the reverse complement as the first step.

Barcode table:

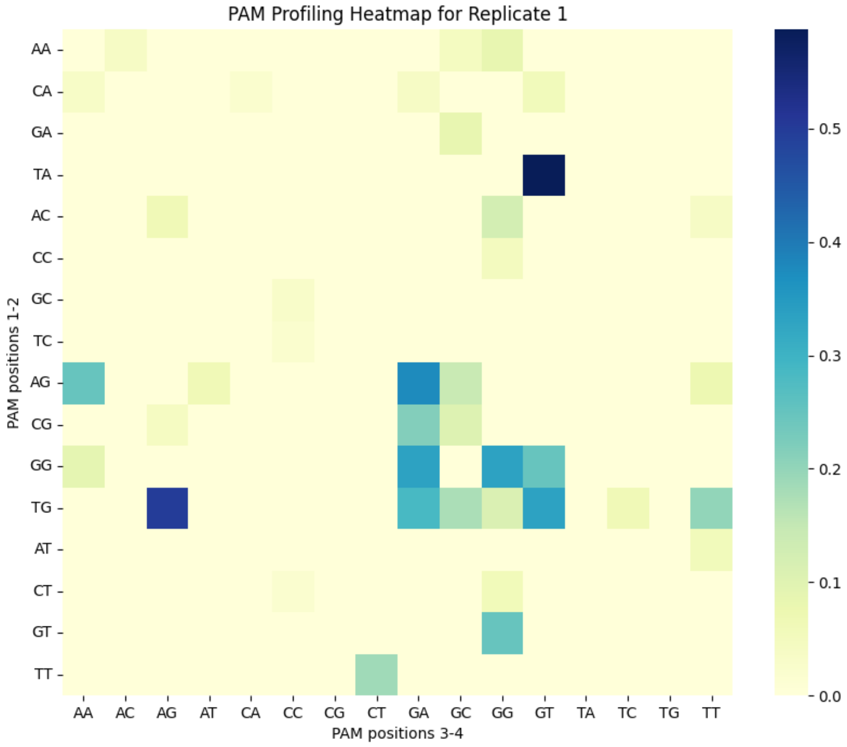
Replicate	Barcode Forward/Reverse
1	GTCAGT
2	TAGCCT
3	CGGTAC

Submission

Upload your code to github and share it with sasha.melkonyan@gmail.com. Repository should contain

1. *src* folder with all the source files

2. *heatmap_rep{1,2,3}.png* - heatmap visualizing the editing rates.
3. Any files with intermediate results.



Reference heatmap for replicate 1.