# Clarifying the amino-acid sequence of coding VNTRs

CSE 280A, Winter 2019

Sara Rahiminejad, Milad Mortazavi

## 1 Introduction

Variable Number Tandem Repeats (VNTRs) are Repeating Units (RUs) of length 6-100 bp and span 3% of human genome. Approximately, 2,000 VNTRs are found in coding sequence, where variation in the number of RUs can change the protein sequence, including shortening of the sequence due to premature stop-codons. Multiple studies have linked variation in VNTRs with Mendelian diseases and complex disorders such as bipolar disorders. Hence, their investigation is of great importance.

In this project, we assume that we are given short single-end reads (Illumina reads) from a diploid individual who has different number of RUs for a specific VNTR in each haplotype. We use adVNTR, a Hidden Markov Model (HMM) based software, to find the number of RUs in each haplotype. Based on this information, we classify the reads into two classes corresponding to two haplotypes. Then we align the reads in each class and find the consensus DNA sequence. We correct sequencing errors by taking into account the read qualities. Afterwards, we translate the consensus sequence and compare the protein sequence with the reference protein. We consider a non-coding VNTR corresponding to CSTB gene, and a coding VNTR corresponding to GP1BA gene. For the non-coding VNTR (CSTB gene), we find the consensus DNA sequence and compare it to the reference DNA sequence to assess accuracy of the method. For the coding VNTR, we compare the amino-acid sequence resulted from translation of the consensus DNA to the reference protein sequence.

The rest of this report is organized as follows: In section 2, we describe how we simulate our data. In section 3, we explain the methods that are use in this project. In section 4, we present our result, and in section 5, we provide the conclusion of our study.

## 2 Data simulation

We simulate the reads with ART [1] which is a simulation tool to generate synthetic reads assuming different technologies. We choose Illumina single-end reads with sequencing system MSv3 which can generate up to 250 base pairs per read.

We choose two different VNTRs as follows:

- CSTB gene VNTR: A non-coding VNTR in chromosome 21, start location: 45196323 and end location: 45196359. It proceeds the gene CSTB with motif ”GCGCGGGGCGGG” and the default RU count is 3. We change the number of RU counts for our haplotypes to 4 and 6 in our samples. Our reads are 150 base pair long.
- GP1BA gene VNTR: A coding VNTR in chromosome 17, start location: 4837118 and end location 4837278. This VNTR is part of GP1BA and is translated during protein generation. Its motif is ”AGCCCGACCACCCCAGAGCCCACCTCAGAGCCCGCCCCC” with length 39 and default RU count 4. We again choose RU counts 4 and 6 for our haplotype reads. Our reads are 250 base pair long.

We try different coverage, 5×, 10×, 20× and 30× to see its effect on accuracy. After getting single-end reads with different RU counts with ART, we mix them and map them to hg19 with BWA [2] to generate SAM files. We then use samtools [3] to generate BAM files which are the inputs to adVNTR.

# 3 Methods

To compute the RU counts of our reads, we use adVNTR [4] which uses an HMM algorithm. The HMM diagram composes of three sections, one for left flanking region, one for right flanking region and one to match the repeating patters. The design of the HMM is shown in Figure 1.

The transition and emission probabilities in the HMM are defined as follows:

$$T(i,j) = \frac{h(i, j + b_0)}{\sum_{i \to l}(h(i,l) + b_0)} \tag{1}$$

$$E_i(\alpha) = \frac{h_i(\alpha + b_1)}{\sum_{\alpha'} h_i(\alpha') + Nb_1} \quad \text{for} \ \ \alpha, \alpha' \in \{A, C, G, T\} \tag{2}$$

where $h(i, j)$ denotes the number of observed transitions from state $i$ to state $j$ in hidden path of each sequence in multiple alignment, and $h_i(\alpha)$ denotes the number of emissions of $\alpha$ in state $i$.

After finding a sequence which matches the HMM profile, the start and end unit counts indicate the number of repeats for the read. In other words, when the structure of $U_s$ and $U_e$ units found are as $U_e^{k_1}(U_sU_e)^{k_2}U_s^{k_3}$, $k_1 + k_2 + k_3$ is reported as the minimum number of RU count. We keep the reads which span the whole VNTR region and discard the ones which only have part of that region.

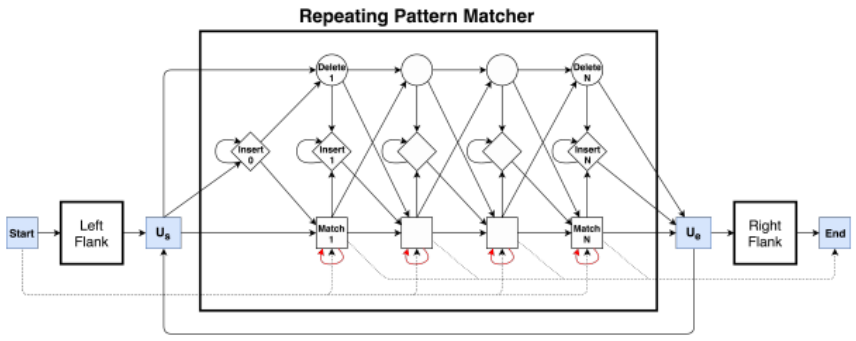


*Figure 1: The VNTR HMM.*

## 3.1 adVNTR modification

To classify the reads, we augmented the adVNTR algorithm as follows. Given a consensus genotype (i.e., $c_i, c_j$), the probability of read $c_k$ to be in class $c_i$ is as follows:

$$Pr(c_k \ in \ class \ i) = \begin{cases} \frac{1}{2} & \text{if } c_k = c_i = c_j \\ \frac{(1-r)}{(1-r)+r_\epsilon^{|c_k - c_j|}} & \text{if } c_k = c_i \\ \frac{r_\epsilon^{|c_k - c_i|}}{(1-r)+r_\epsilon^{|c_k - c_i|}} & \text{if } c_k = c_j \\ \frac{r_\epsilon^{|c_k - c_i|}}{r_\epsilon^{|c_k - c_i|}+r_\epsilon^{|c_k - c_j|}} & \text{if } c_k \neq c_i, c_k \neq c_j \end{cases} \tag{3}$$

Reads can then be classified based on the mentioned probability distribution. We write each class in a separate fastq file with the quality of reads included in order to align them later. We use the quality of reads to identify the sequencing errors in the reads.

## 3.2 Multiple alignment

Now that we have two classes of reads, we can find a multiple alignment of the each class reads. Since we have fastq files, we know the quality of reads in every locus. The quality of each read is defined as $Q = -10log_{10}e$ where $e$ is the probability of having a sequencing error for that locus. We can compute the probability of seeing a collection of reads ($S$) in a particular locus given that a particular base pair is the correct base pair by knowing the number of observations for each base pair and the sequencing errors for each base pair as follows:

$$Pr(S|A\ is\ correct) = (1-e_A)^{n_A}.e_C^{n_C}.e_G^{n_G}.e_T^{n_T} \tag{4}$$
$$Pr(S|C\ is\ correct) = e_A^{n_A}.(1-e_C)^{n_C}.e_G^{n_G}.e_T^{n_T} \tag{5}$$
$$Pr(S|G\ is\ correct) = e_A^{n_A}.e_C^{n_C}.(1-e_G)^{n_G}.e_T^{n_T} \tag{6}$$
$$Pr(S|T\ is\ correct) = e_A^{n_A}.e_C^{n_C}.e_G^{n_G}.(1-e_T)^{n_T} \tag{7}$$

where the sequencing errors are defined as $e = 10^{-Q/10}$. $S$ is the collection of base pairs at a location. Values of $n$ are the number of observation of each base pair. For example, when $S = (A, A, A, C, A, C)$, $n_A = 3$, $n_C = 2$, and $n_T = n_G = 0$. Usually there are only two variations per base pair and the other two are not present. The consensus base pair in a locus is the one having the largest probability among all.

Besides multiple alignment, we use SPAdes to find the consensus DNA. SPAdes [5] is an algorithm based on De-Bruijin Garph (DBG) assembly where a k-mer graph is constructed from reads. Then, a directed edge is drawn from each left (k-1) mer to the corresponding right (k-1) mer. Following the path in the graph yields the sequence assembly of the original reads. Figure 2 shows how DBG assembly works. In this figure, size of k-mers are 4.

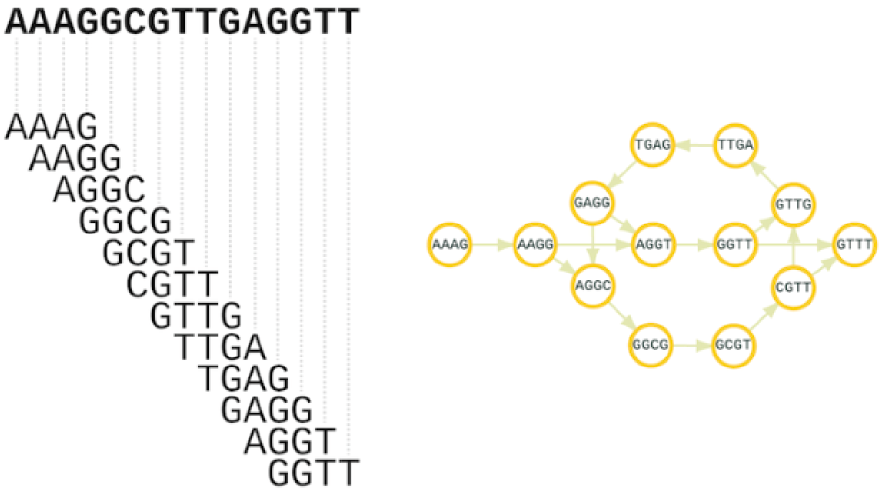
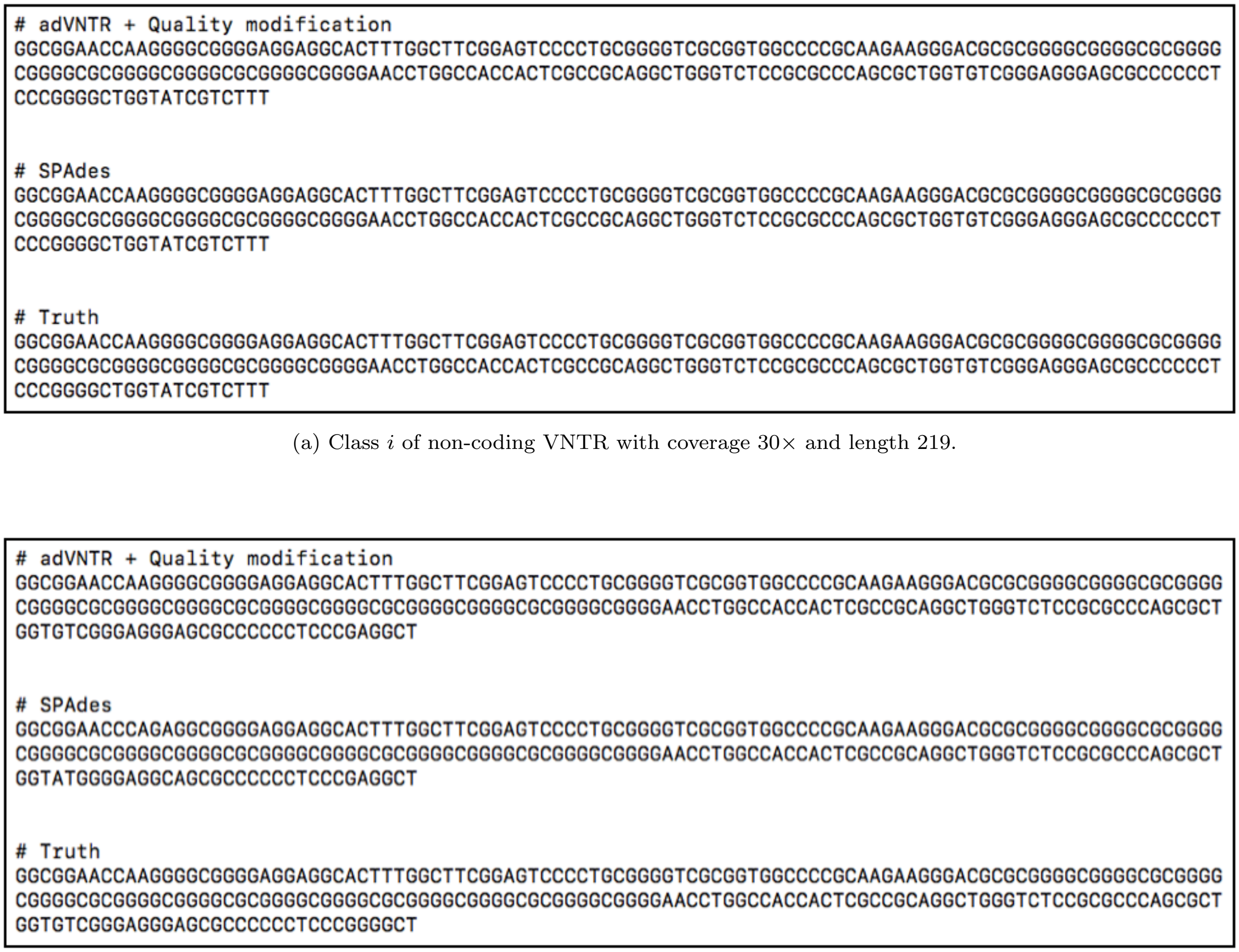


*Figure 2: De-Brujin Graph assembly for AAAGGCGTTGAGGTT.*

# 4 Results and Discussion

We use adVNTR (section 3.1) to classify our simulated reads into two classes. Class $i$ for reads having RU count 4, and class $j$ for reads having RU count 6. Then we use multiple alignments with sequencing error correction (will be explained more in section 4.1) and SPAdes for each class separately to obtain the consensus sequence. Figure 3 shows the comparison of the consensus DNA sequence with the reference DNA sequence. This example is for the non-coding VNTR with coverage 30×.

As seen from Figure 3, both multiple alignment and SPAdes are able to find the right sequence. In part (b), multiple alignment works better than SPAdes. Out of 219 base pairs, there is only one base pair which does not match to the reference DNA sequence and it occurs due to low quality of the base pair read with no other read to correct it at that position. For SPAdes, the number of errors are 6 and this is related to non-uniformity of coverage.



```
# adVNTR + Quality modification
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGG
CGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCT
CCCGGGGCTGGTATCGTCTTT


# SPAdes
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGG
CGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCT
CCCGGGGCTGGTATCGTCTTT


# Truth
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGG
CGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCT
CCCGGGGCTGGTATCGTCTTT
```

(a) Class $i$ of non-coding VNTR with coverage 30× and length 219.

```
# adVNTR + Quality modification
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGG
CGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCT
GGTGTCGGGAGGGAGCGCCCCCCTCCCGAGGCT


# SPAdes
GGCGGAACCCAGAGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGG
CGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCT
GGTATGGGGAGGCAGCGCCCCCCTCCCGAGGCT


# Truth
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGG
CGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCT
GGTGTCGGGAGGGAGCGCCCCCCTCCCGGGGCT
```

(b) Class $j$ of non-coding VNTR with coverage 30× and length 231.

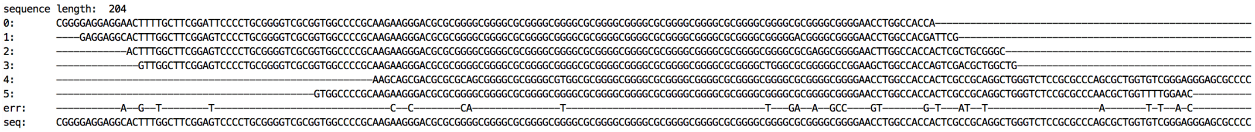
*Figure 3: Results of assembly for multiple alignment, SPAdes, and the correct assembly coming from chromosome 21.*

## 4.1 Sequencing error correction

As mentioned in section 3.2 we take into account the quality of the base pair reads. In the case where all the reads are similar in a locus we choose that base pair as consensus. In the case where multiple base pairs are present at a locus, we compute the conditional probability of seeing the observations at that locus given that the correct base pair is X. We then choose the base pair X which maximizes this conditional probability of seeing the base pairs.

A sample of alignment for our reads as well as the sequencing errors can be seen in Figure 4. The last line is the consensus sequence and the line above it is the detected sequencing errors in the sequence.

This method showed that it can return the reference sequence with high accuracy. In fact the accuracy for class $i$ of the non-coding VNTR was 100% and for class $j$ only one base pair was reported incorrectly due to lack of high quality read in that locus and presence of one low quality sequencing error. Overall, this method for error correction is found do be very powerful.



```
sequence length:  204
0:      CGGGGAGGAGGAACTTTTGCTTCGGATTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCA------------------------------------------------------
1:      ----GAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGGACGGGGCGGGGAACCTGGCCACGATTCG--------------------------------------------------
2:      ------------ACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGAGGCGGGGAACTTGGCCACCACTCGCTGCGGGC------------------------------------------
3:      --------------GTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCTGGGCGCGGGGGCCGGAAGCTGGCCACCAGTCGACGCTGGCTG----------------------------------------
4:      ------------------------------------------------------AAGCAGCGACGCGCGCAGCGGGGCGCGGGGCGTGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCC
5:      --------------------------------------------GTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAACGCTGGTTTTGGAAC----------
err:    -----------A--G--T--------T------------------------------C--C--------CA---------------T------------------------------------T---GA--A--GCC----GT--------G-T---AT--T---------------------A--------T-T--A-C----------
seq:    CGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCC
```

*Figure 4: Alignment of reads with coverage 5× for class j of non-coding VNTR. The sequence correction is performed based on the read qualities. The consensus DNA sequence is reported on the last line, and the sequence errors are shown in the line above that.*

## 4.2 Indel detection

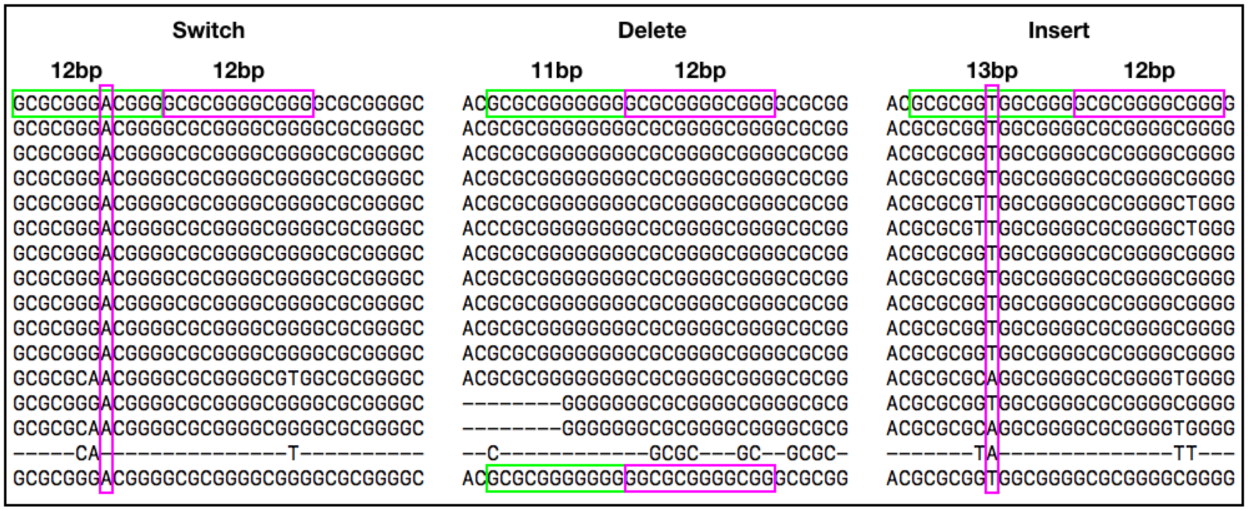
We test our method when a switch in a base pair, deletion or insertion is present in our sample. We try these three cases:

1. Switch: change a base pair from ”G” to ”A” in the first RU. The new motif is ”GCGCGGG**A**CGGG”.

2. Delete: delete a ”C” in the middle of the first RU. The new motif is ”GCGCGGGGGGG”

3. Insert: insert an extra ”T” in the middle of the first RU. The new unit is ”GCGCGG**T**GGCGGG”.

As can be see in figure 5 we can reconstruct the consensus sequence correctly for all of the conditions above. It’s also important to note that adVNTR can retrieve the correct number of repeats even with these modifications and deviations in the sample sequence.



```
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGTTGGCGGGGCGCGGGGCTGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACCCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGTTGGCGGGGCGCGGGGCTGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
GCGCGCAACGGGGCGCGGGGCGTGGCGCGGGGC    ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG    ACGCGCGCAGGCGGGGCGCGGGGTGGGG
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    --------GGGGGGGCGCGGGGCGGGGCGCG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
GCGCGCAACGGGGCGCGGGGCGGGGCGCGGGGC    --------GGGGGGGCGCGGGGCGGGGCGCG    ACGCGCGCAGGCGGGGCGCGGGGTGGGG
-----CA---------------T----------    --C------------GCGC---GC--GCGC-    -------TA---------------TT---
GCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC    ACGCGCGGGGGGGGGCGCGGGGCGGGCGCGG    ACGCGCGGTGGCGGGGCGCGGGGCGGGG
```

*Figure 5: Sequence alignment and error correction in the presence of base pair switch, deletion, or insertion in the sample. We can retrieve these small deviations from the reference genome and still assemble the correct consensus sequence.*

## 4.3 Coverage variation along sequence

As mentioned previously, adVNTR returns only the reads which cover the whole extent of the VNTR region. Therefore, we loose the reads which cover only the beginning or end parts of the VNTR. This causes our coverage to be non-uniform. This is going to be a problem in some sequencing methods such as SPAdes. Figure 6 shows how coverage varies as a function of locus. The coverage drops fast at the beginning and end regions of the consensus sequence.

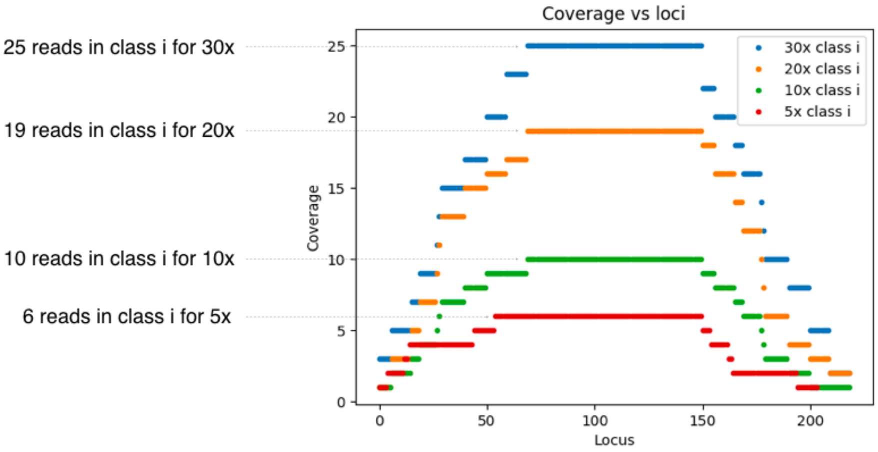


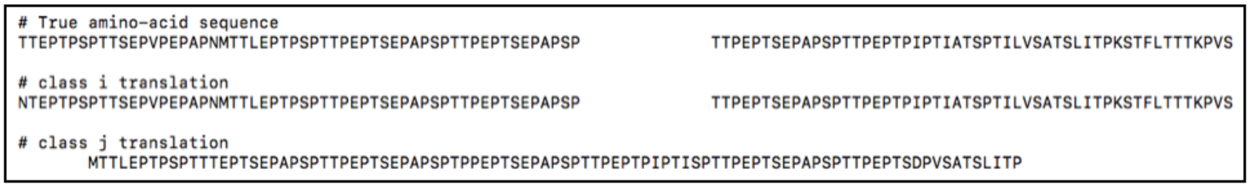
*Figure 6: Coverage vs loci for the non-coding VNTR.*

### 4.4 Translation of the consensus mRNA

To construct the amino-acid sequence, first we need to transcribe the consensus DNA to mRNA. To do transcription, we change "T"s in the consensus DNA to "U". The result of this process is mRNA. After getting the mRNA, we can translate it to a protein sequence in six ways; three ways for the forward and three for the reverse strands, both from the 5′ to 3′ end of the DNA.

Since the non-coding VNTR is not in the coding region, we do not expect to get the same protein sequence as the reference protein. Hence, we just show the results of the coding VNTR. Figure 7 shows the comparison of translation of the consensus DNA sequence for the coding VNTR (corresponding to GP1BA gene) to the reference protein. The reference protein is longer (652 amino-acids), however our reads are 250 base pairs and our resulting amino-acid has just 93 amino acids, and we can only compare these base pairs.

The gap which is evident in the middle of the reference and class $i$ sequences is to accommodate for the extra base pairs present in the class $j$ with two more RUs. The result shows that 99% of the amino-acid base pairs for class $i$ matches with the reference protein.



```
# True amino-acid sequence
TTEPTPSPTTSEPVPEPAPNMTTLEPTPSPTTPEPTSEPAPSPTTPEPTSEPAPSP              TTPEPTSEPAPSPTTPEPTPIPTIATSPTILVSATSLITPKSTFLTTTKPVS

# class i translation
NTEPTPSPTTSEPVPEPAPNMTTLEPTPSPTTPEPTSEPAPSPTTPEPTSEPAPSP              TTPEPTSEPAPSPTTPEPTPIPTIATSPTILVSATSLITPKSTFLTTTKPVS

# class j translation
        MTTLEPTPSPTTTEPTSEPAPSPTTPEPTSEPAPSPTPPEPTSEPAPSPTTPEPTPIPTISPTTPEPTSEPAPSPTTPEPTSDPVSATSLITP
```

*Figure 7: Most likely amino-acid sequence of the coding VNTR for class i and class j. The first row is a segment of the protein sequence corresponding to gene GP1BA. The second row is the translation of class i consensus DNA, and the third row is the translation of class j. The gap present in the first and second rows are to accommodate the RU increament in class j.*

## 5 Conclusion

We have applied adVNTR, an HMM based software, to genotype our sample and find the number of RU counts in each haplotype. We then classified our reads based on a probability distribution and the results of adVNTR. We applied an alignment method by taking into account the quality of our reads to come up with the consensus DNA sequence. We also tried SPAdes as a *de novo* sequencing algorithm and compared the consensus sequences with the reference to assess the accuracy of the methods. We successfully found the consensus DNA sequence when we took into account the quality of the reads to correct the sequencing errors. We also found that SPAdes sometimes has a hard time giving the consensus sequence due to non-uniformity of our reads resulting from adVNTR. We then tested our method with the presence of a switch in a base pair, deletion or insertion in the sample. We found that our method is still accurate in the presence of these modifications. For the coding VNTR, we translated the consensus DNA sequence and compared it with the reference protein. The resulting protein sequence is shorter than the total length of the reference protein since our reads are only 250 base pair long and cannot cover all the coding region of the genome. However, the accuracy of the match with the corresponding segment of the reference protein was very high.

We conclude that our alignment method coupled with adVNTR genotyping can result in an accurate haplotype separation and consensus DNA sequence for both haplotypes. This method can be applied to other VNTRs and a broader range of genes in the future.

## 6 References

[1] Huang, W., et al., *ART: a next-generation sequencing read simulator*. Bioinformatics, 2012. **28**(4): p. 593-4.

[2] Li, H. and R. Durbin, *Fast and accurate short read alignment with Burrows-Wheeler Transform*. Bioinformatics, 2009. **25**(14): p. 1754-60.

[3] Li, H., et al., *The Sequence Alignment/Map format and SAMtools*. Bioinformatics, 2009. **25**(16): p. 2078-9.

[4] Bakhtiari, M., et al., *Targeted Genotypying of Variable Number Tandem Repeats with adVNTR*. Genome Res, 2018. **28**(11): p. 1709-1719.

[5] Bankevich, A., et al., *SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing*. J Comput Biol. 2012. **19**(5): p. 455-77.

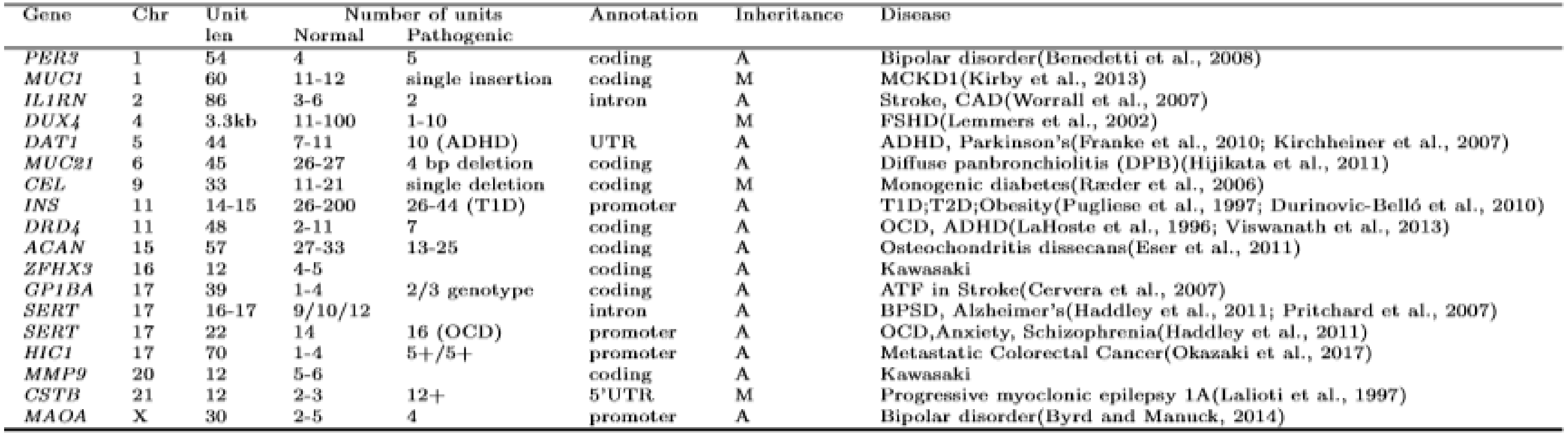
# Clarifying the amino-acid sequence of coding VNTRs

Sara Rahiminejad
Milad Mortazavi

CSE 280A

# Variable Number Tandem Repeats (VNTRs)

- VNTRs: repeating units (RUs) of length 6-100 bp and span 3% of human genome.
- 2000 VNTRs in coding sequence.
- Variation in RU counts of VNTRs will change the protein sequence.



| Gene | Chr | Unit | Number of units | | Annotation | Inheritance | Disease |
|---|---|---|---|---|---|---|---|
| | | len | Normal | Pathogenic | | | |
| *PER3* | 1 | 54 | 4 | 5 | coding | A | Bipolar disorder(Benedetti et al., 2008) |
| *MUC1* | 1 | 60 | 11-12 | single insertion | coding | M | MCKD1(Kirby et al., 2013) |
| *IL1RN* | 2 | 86 | 3-6 | 2 | intron | A | Stroke, CAD(Worrall et al., 2007) |
| *DUX4* | 4 | 3.3kb | 11-100 | 1-10 | | M | FSHD(Lemmers et al., 2002) |
| *DAT1* | 5 | 44 | 7-11 | 10 (ADHD) | UTR | A | ADHD, Parkinson's(Franke et al., 2010; Kirchheiner et al., 2007) |
| *MUC21* | 6 | 45 | 26-27 | 4 bp deletion | coding | A | Diffuse panbronchiolitis (DPB)(Hijikata et al., 2011) |
| *CEL* | 9 | 33 | 11-21 | single deletion | coding | M | Monogenic diabetes(Ræder et al., 2006) |
| *INS* | 11 | 14-15 | 26-200 | 26-44 (T1D) | promoter | A | T1D;T2D;Obesity(Pugliese et al., 1997; Durinovic-Belló et al., 2010) |
| *DRD4* | 11 | 48 | 2-11 | 7 | coding | A | OCD, ADHD(LaHoste et al., 1996; Viswanath et al., 2013) |
| *ACAN* | 15 | 57 | 27-33 | 13-25 | coding | A | Osteochondritis dissecans(Eser et al., 2011) |
| *ZFHX3* | 16 | 12 | 4-5 | | coding | A | Kawasaki |
| *GP1BA* | 17 | 39 | 1-4 | 2/3 genotype | coding | A | ATF in Stroke(Cervera et al., 2007) |
| *SERT* | 17 | 16-17 | 9/10/12 | | intron | A | BPSD, Alzheimer's(Haddley et al., 2011; Pritchard et al., 2007) |
| *SERT* | 17 | 22 | 14 | 16 (OCD) | promoter | A | OCD,Anxiety, Schizophrenia(Haddley et al., 2011) |
| *HIC1* | 17 | 70 | 1-4 | 5+/5+ | promoter | A | Metastatic Colorectal Cancer(Okazaki et al., 2017) |
| *MMP9* | 20 | 12 | 5-6 | | coding | A | Kawasaki |
| *CSTB* | 21 | 12 | 2-3 | 12+ | 5'UTR | M | Progressive myoclonic epilepsy 1A(Lalioti et al., 1997) |
| *MAOA* | X | 30 | 2-5 | 4 | promoter | A | Bipolar disorder(Byrd and Manuck, 2014) |

*Bakhtiari et al., Targeted Genotyping of Variable Number Tandem Repeats with adVNTR, 2018.*

# Project outline

- Input:
  - Collection of reads from a coding VNTR.
  - RU count.
  - Reference protein.

- Output:
  - Multiple alignment of the reads
  - Consensus DNA sequence.
  - A predicted protein sequence by translating the RNA.

# Data simulation

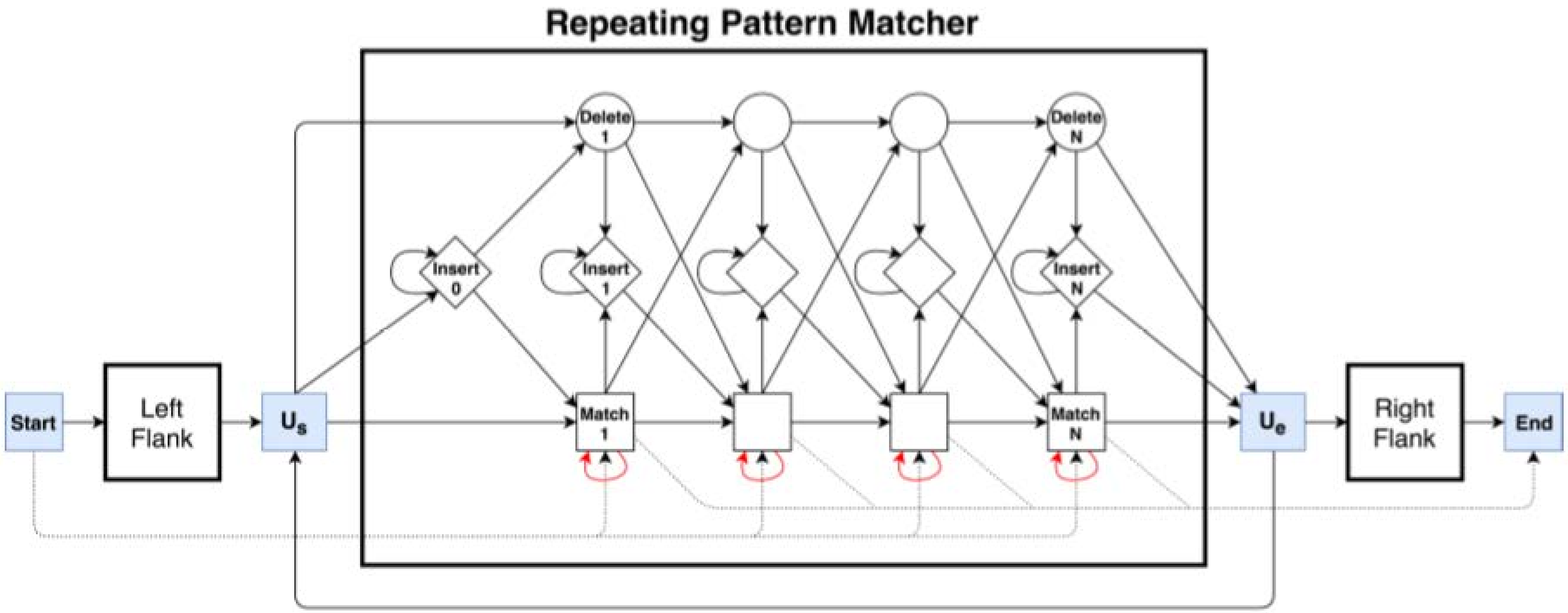
- VNTR chosen:

| Chromosome | Start | End | VNTR-id | Gene | Motif | Def RU |
|---|---|---|---|---|---|---|
| ch21 | 45196323 | 45196359 | 301645 | CSTB | GCGCGGGGCGGG | 3 |

- Build **RU= 4** and **RU=6** by modifying ***chr21.fa***
- Obtain reads from ART with these two RU counts
- Mix the two type reads
- Map the reads to ***hg19*** and generate SAM files with bwa
- Generate BAM files from SAM files with samtools

# Methods

# Finding VNTRs using HMM[1]



$$T(i,j) = \frac{h(i,j)+b_0}{\sum_{i\to l}(h(i,l)+b_0)}, \quad E_i(\alpha) = \frac{h_i(\alpha)+b_1}{\sum_{\alpha'}(h_i(\alpha')+b_1)} \quad \text{for } \alpha, \alpha' \in \{A, C, G, T\}.$$

*Bakhtiari et al., Targeted Genotyping of Variable Number Tandem Repeats with adVNTR, 2018.*

# Using adVNTR to detect RU-counts and classify

- Modify adVNTR to classify the reads based on RU-counts:
  Given a consensus genotype ($c_i$, $c_j$),
  Compute the probability of read $c_k$ to be in class i:

$$\Pr(c_k\ in\ class\ i)\begin{cases} \frac{1}{2} & c_k = c_i = c_j \\ \frac{(1-r)}{(1-r)+r_\epsilon^{|c_k-c_i|}} & c_k = c_i \\ \frac{r_\epsilon^{|c_k-c_i|}}{(1-r)+r_\epsilon^{|c_k-c_i|}} & c_k = c_j \\ \frac{r_\epsilon^{|c_k-c_i|}}{r_\epsilon^{|c_k-c_i|}+r_\epsilon^{|c_k-c_j|}} & c_k \neq c_i\ , c_k \neq c_j \end{cases}$$

$r_\epsilon$ is defined s.t. $r_\epsilon^\Delta$ is the probability of RU counting error by $\pm\Delta$ in the estimation of the true count.

$$r = \frac{2r_\epsilon}{1-r_\epsilon}$$

- Classify the reads based on this probability distribution.

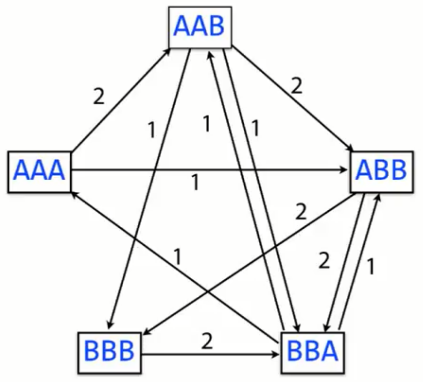
# Two approaches to short read assembly

- **OLG**: Overlap-Layout-Consensus assembly
- **DBG**: De Brujin Graph assembly
    - Constructing k-mer graph from reads (left and right (k-1)mers).
    - Draw a directed edge from each left (k-1)mer to the corresponding right (k-1)mer.
    - Follow the path in the graph to assemble.

AAABBBA

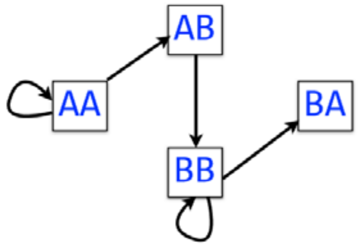
AAA, AAB, ABB, BBB, BBA

Overlap graph



Overlap-Layout-Consensus (OLC) assembly

De Bruijn graph



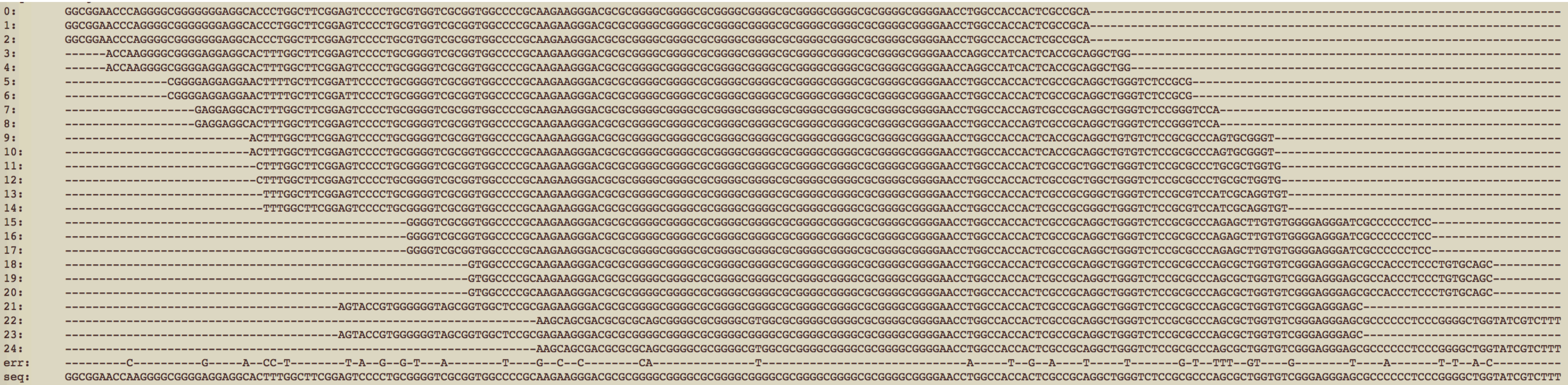
De Bruijn Graph based (DBG) assembly

# Results

# Alignment of multiple reads

- Sequence length: 219
- Decide consensus base pair based on read quality:
    - $Prob\ A\ with\ S = (1 - e_A)^{n_A} e_B^{n_B}$
    - $Prob\ B\ with\ S = e_A^{n_A} (1 - e_B)^{n_B}$

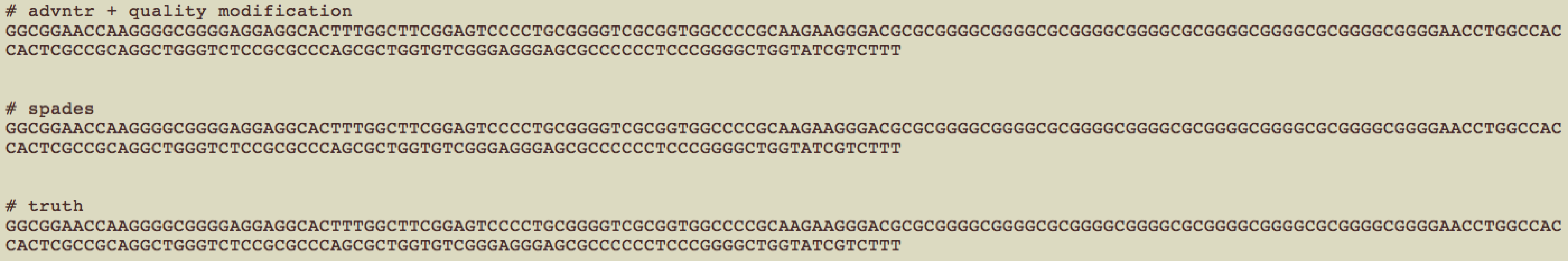
    $where\ Q = -\log_{10} e$
- Observed exact reconstruction of consensus sequence.
- Of 219 base pairs, 1 base pair did not match because of low quality and coverage.



```
0:      GGCGGAACCCAGGGGCGGGGGGGAGGCACCCTGGCTTCGGAGTCCCCTGCGTGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCA---------------------------------------------------------------------
1:      GGCGGAACCCAGGGGCGGGGGGGAGGCACCCTGGCTTCGGAGTCCCCTGCGTGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCA---------------------------------------------------------------------
2:      GGCGGAACCCAGGGGCGGGGGGGAGGCACCCTGGCTTCGGAGTCCCCTGCGTGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCA---------------------------------------------------------------------
3:      ------ACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCAGGCCATCACTCACCGCAGGCTGG---------------------------------------------------------------
4:      ------ACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCAGGCCATCACTCACCGCAGGCTGG---------------------------------------------------------------
5:      ---------------CGGGGAGGAGGAACTTTTGCTTCGGATTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCG------------------------------------------------------
6:      ---------------CGGGGAGGAGGAACTTTTGCTTCGGATTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCG------------------------------------------------------
7:      -------------------GAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCAGTCGCCGCAGGCTGGGTCTCCGGGTCCA--------------------------------------------------
8:      -------------------GAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCAGTCGCCGCAGGCTGGGTCTCCGGGTCCA--------------------------------------------------
9:      ---------------------------ACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCACCGCAGGCTGTGTCTCCGCGCCCAGTGCGGGT------------------------------------------
10:     ---------------------------ACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCACCGCAGGCTGTGTCTCCGCGCCCAGTGCGGGT------------------------------------------
11:     ----------------------------CTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCTGGCTGGGTCTCCGCGCCCTGCGCTGGTG-----------------------------------------
12:     ----------------------------CTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCTGGCTGGGTCTCCGCGCCCTGCGCTGGTG-----------------------------------------
13:     -----------------------------TTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCGGGCTGGGTCTCCGCGTCCATCGCAGGTGT----------------------------------------
14:     -----------------------------TTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCGGGCTGGGTCTCCGCGTCCATCGCAGGTGT----------------------------------------
15:     --------------------------------------------------GGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGAGCTTGTGTGGGGAGGGATCGCCCCCCTCC-------------------
16:     --------------------------------------------------GGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGAGCTTGTGTGGGGAGGGATCGCCCCCCTCC-------------------
17:     --------------------------------------------------GGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGAGCTTGTGTGGGGAGGGATCGCCCCCCTCC-------------------
18:     -----------------------------------------------------------GTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCACCCTCCCTGTGCAGC----------
19:     -----------------------------------------------------------GTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCACCCTCCCTGTGCAGC----------
20:     -----------------------------------------------------------GTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCACCCTCCCTGTGCAGC----------
21:     ----------------------------------------AGTACCGTGGGGGGTAGCGGTGGCTCCGCGAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGC-----------------------------
22:     ---------------------------------------------------------------------AAGCAGCGACGCGCGCAGCGGGGCGCGGGGCGTGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCTCCCGGGGCTGGTATCGTCTTT
23:     ----------------------------------------AGTACCGTGGGGGGTAGCGGTGGCTCCGCGAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGC-----------------------------
24:     ---------------------------------------------------------------------AAGCAGCGACGCGCGCAGCGGGGCGCGGGGCGTGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCTCCCGGGGCTGGTATCGTCTTT
err:    ---------C----------G-----A--CC-T--------T-A--G--G-T---A--------T----G--C--C--------CA---------------T------------------------------A-----T--G--A----T-----T-------G-T--TTT--GT----G--------T----A-------T-T--A-C----------
seq:    GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCTCCCGGGGCTGGTATCGTCTTT
```

# Results of SPAdes and adVNTR comparison

class i, 30x, length 219:

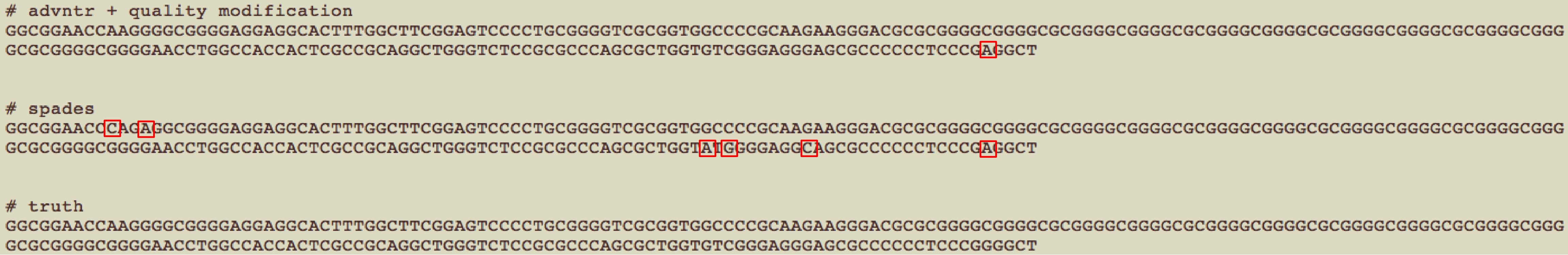


```
# advntr + quality modification
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCAC
CACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCTCCCGGGGCTGGTATCGTCTTT


# spades
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCAC
CACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCTCCCGGGGCTGGTATCGTCTTT


# truth
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGAACCTGGCCAC
CACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCTCCCGGGGCTGGTATCGTCTTT
```

class j, 30x, length 231:



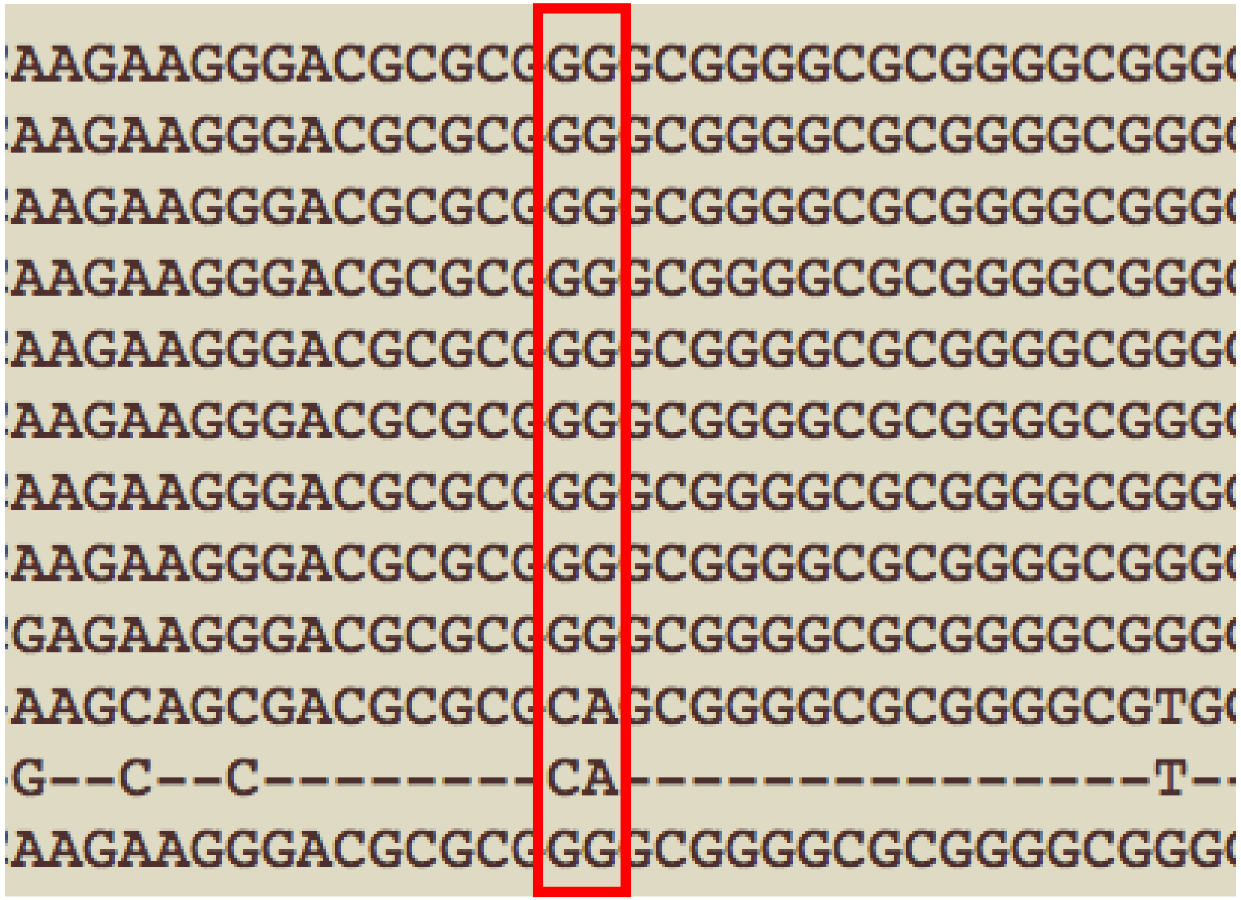
```
# advntr + quality modification
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGG
GCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCTCCCGAGGCT


# spades
GGCGGAACCCAGAGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGG
GCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTATGGGGAGGCAGCGCCCCCCTCCCGAGGCT


# truth
GGCGGAACCAAGGGGCGGGGAGGAGGCACTTTGGCTTCGGAGTCCCCTGCGGGGTCGCGGTGGCCCCGCAAGAAGGGACGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGGGCGCGGGGCGGG
GCGCGGGGCGGGGAACCTGGCCACCACTCGCCGCAGGCTGGGTCTCCGCGCCCAGCGCTGGTGTCGGGAGGGAGCGCCCCCCTCCCGGGGCT
```

# Sequencing error correction

- Sequence errors are low quality reads and will be eliminated.



Reads

```
AAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
AAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
AAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
AAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
AAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
AAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
AAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
AAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
GAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
AAGCAGCGACGCGCCGCAGCGGGGCGCGGGGCGTG
```

Errors

```
G--C--C---------CA---------------T--
```

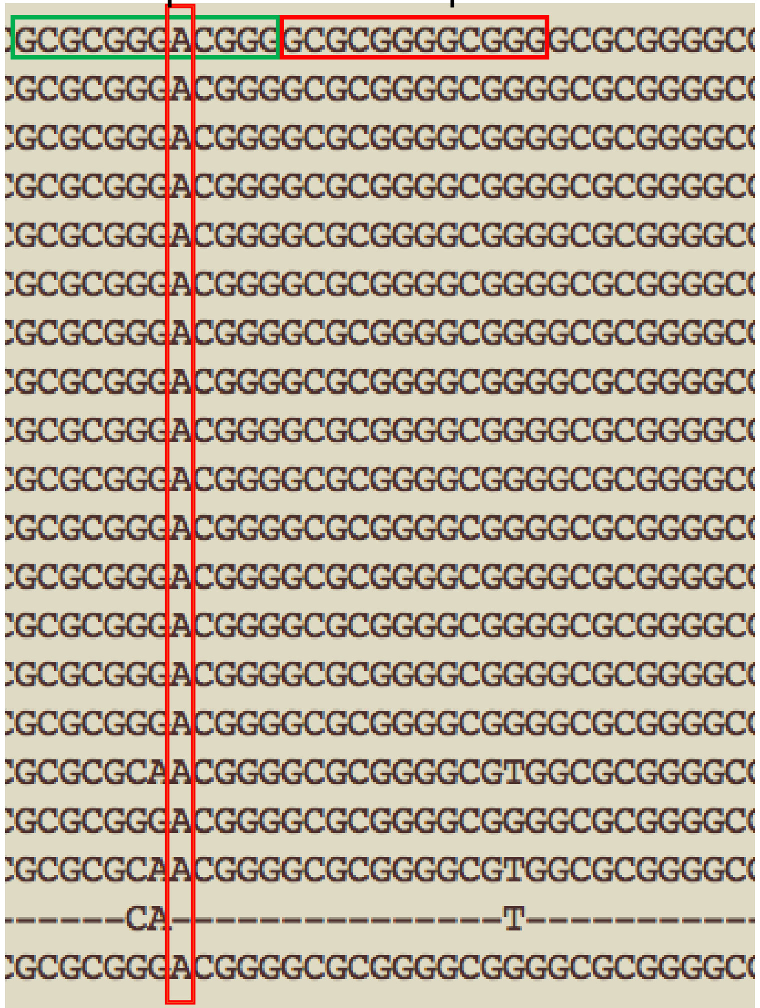
Sequence alignment

```
AAGAAGGGACGCGCCGGGGCGGGGCGCGGGGCGGG
```

# Indel detection vs sequencing errors

**Switch**

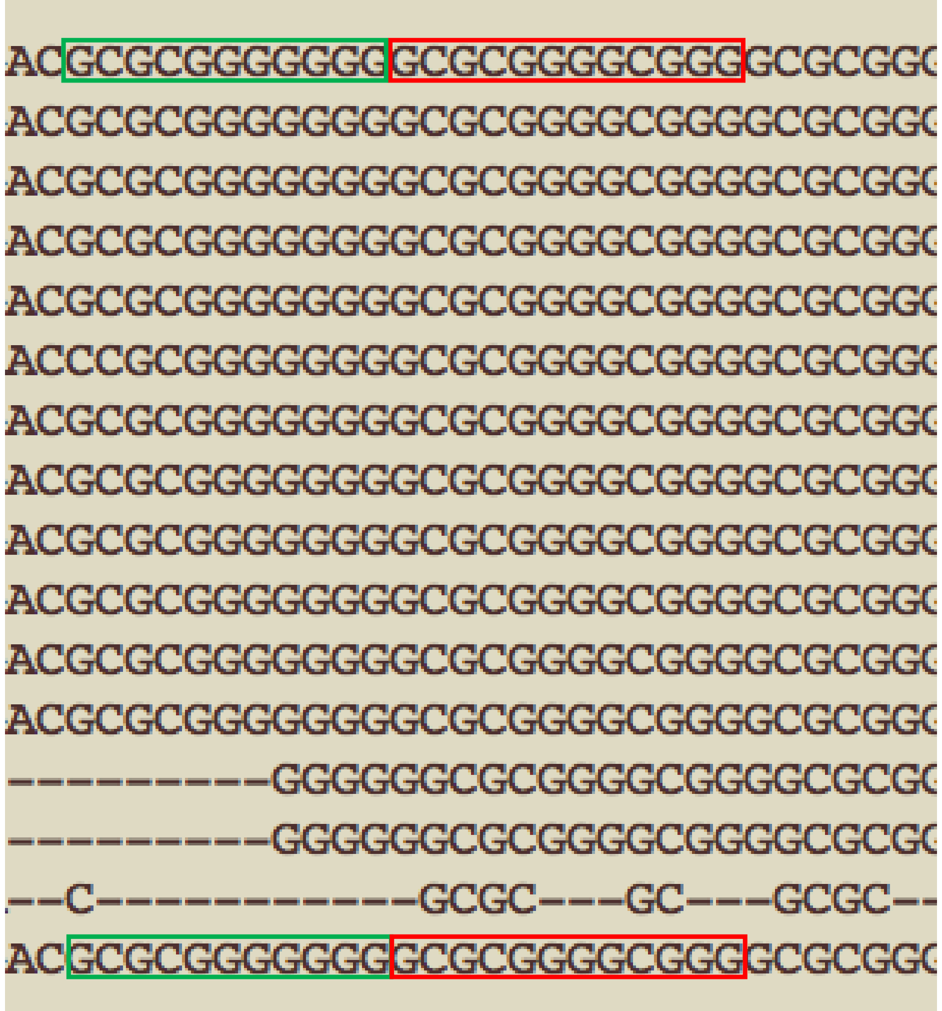
12 bp 12 bp



```
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGCAACGGGGCGCGGGGCGTGGCGCGGGGC
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
CGCGCGCAACGGGGCGCGGGGCGTGGCGCGGGGC
------CA---------------T----------
CGCGCGGGACGGGGCGCGGGGCGGGGCGCGGGGC
```

**Delete**

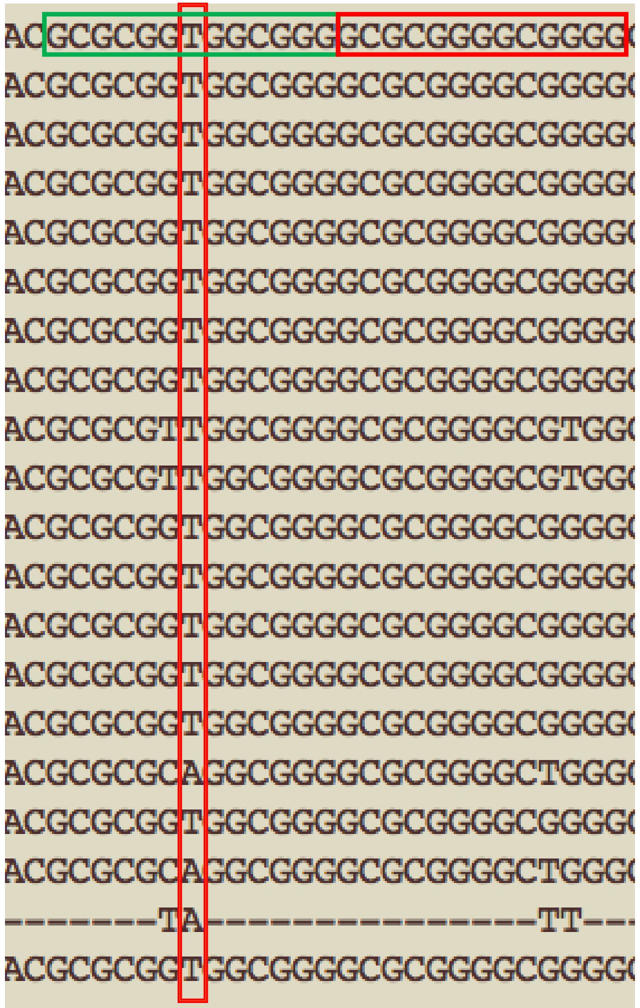
11 bp 12 bp



```
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACCCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
---------GGGGGGCGCGGGGCGGGGCGCG
---------GGGGGGCGCGGGGCGGGGCGCG
--C-----------GCGC---GC---GCGC--
ACGCGCGGGGGGGGCGCGGGGCGGGGCGCGG
```

**Insert**

13 bp 12 bp



```
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGTTGGCGGGGCGCGGGGCGTGG
ACGCGCGTTGGCGGGGCGCGGGGCGTGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGCAGGCGGGGCGCGGGGCTGGG
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
ACGCGCGCAGGCGGGGCGCGGGGCTGGG
-------TA---------------TT---
ACGCGCGGTGGCGGGGCGCGGGGCGGGG
```

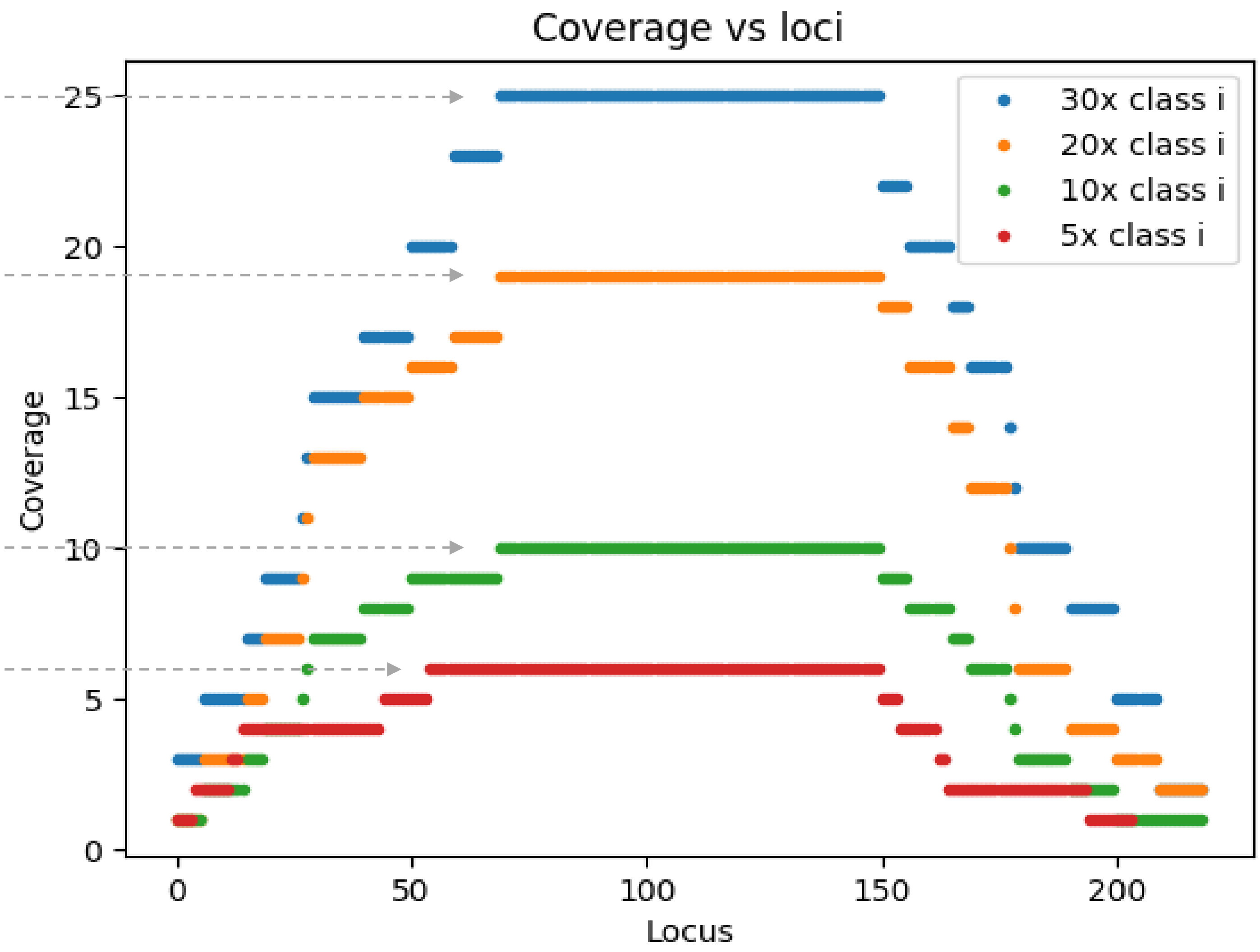
# Coverage variation along sequence

25 reads in class i for 30x

19 reads in class i for 20x

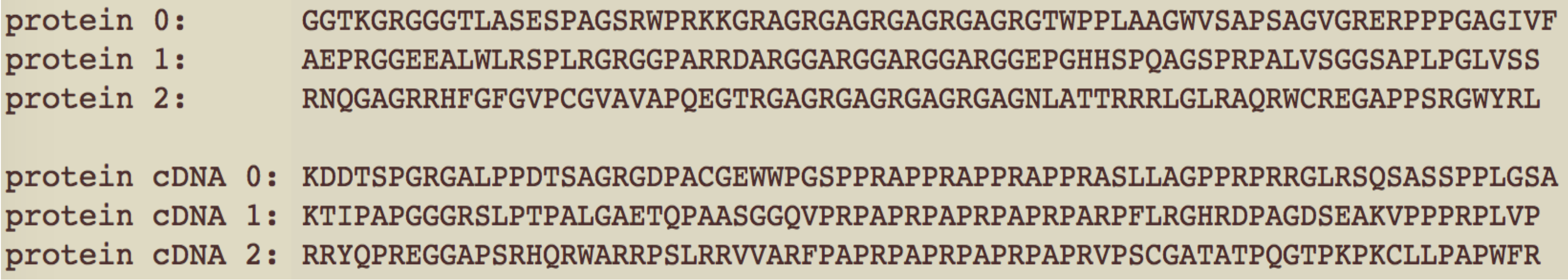
10 reads in class i for 10x

6 reads in class i for 5x



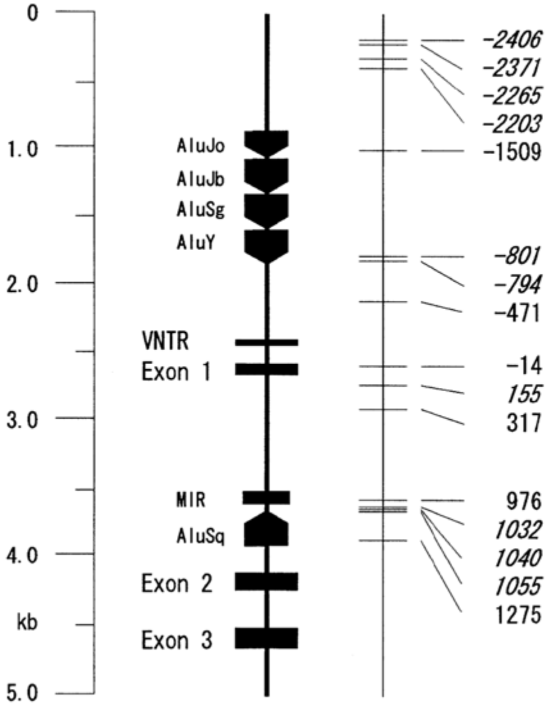
# Translation of CSTB gene

- Find all possible protein sequences:
  - 3 protein sequences for translating in the 5’ to 3’ direction.
  - 3 protein sequences for translating the reverse strand.



```
protein 0:      GGTKGRGGGTLASESPAGSRWPRKKGRAGRGAGRGAGRGAGRGTWPPLAAGWVSAPSAGVGRERPPPGAGIVF
protein 1:      AEPRGGEEALWLRSPLRGRGGPARRDARGGARGGARGGARGGEPGHHSPQAGSPRPALVSGGSAPLPGLVSS
protein 2:      RNQGAGRRHFGFGVPCGVAVAPQEGTRGAGRGAGRGAGRGAGNLATTRRRLGLRAQRWCREGAPPSRGWYRL

protein cDNA 0: KDDTSPGRGALPPDTSAGRGDPACGEWWPGSPPRAPPRAPPRAPPRASLLAGPPRPRRGLRSQSASSPPLGSA
protein cDNA 1: KTIPAPGGGRSLPTPALGAETQPAASGGQVPRPAPRPAPRPAPRPARPFLRGHRDPAGDSEAKVPPPRPLVP
protein cDNA 2: RRYQPREGGAPSRHQRWARRPSLRRVVARFPAPRPAPRPAPRPAPRVPSCGATATPQGTPKPKCLLPAPWFR
```

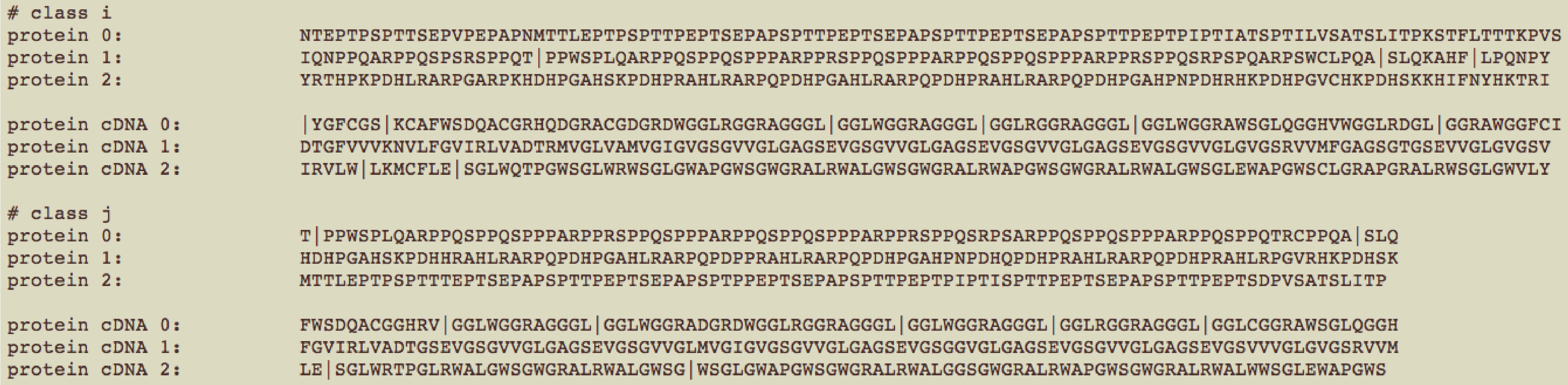


CSTB protein sequence:

MMCGAPSATQPATAETQHIADQVRSQLEEKENKKFPVFKAVSFKSQVVAGTNYFIKVHVGDEDFVHLRVFQSLPHENKPLTLSNYQTNKA KHDELTYF

*Osava et al., Evolution of the cystatin B gene: implications for the origin of its variable dodecamer tandem repeat in humans, 2002*

# Translation of GP1BA gene

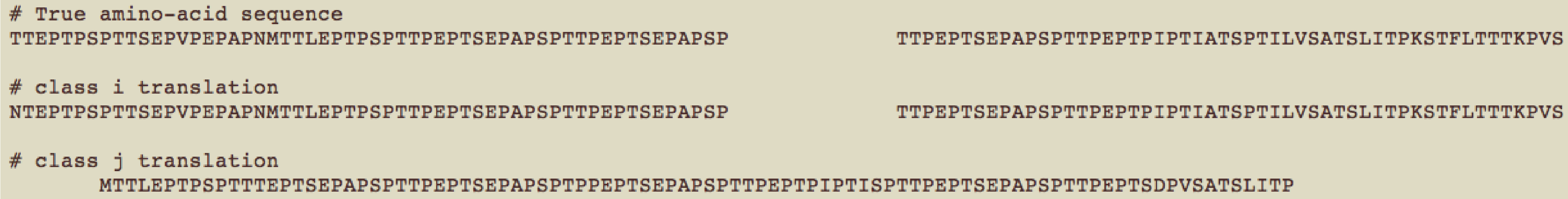


```
# class i
protein 0:               NTEPTPSPTTSEPVPEPAPNMTTLEPTPSPTTPEPTSEPAPSPTTPEPTSEPAPSPTTPEPTSEPAPSPTTPEPTPIPTIATSPTILVSATSLITPKSTFLTTTKPVS
protein 1:               IQNPPQARPPQSPSRSPPQT|PPWSPLQARPPQSPPQSPPPARPPRSPPQSPPPARPPQSPPQSPPPARPPRSPPQSRPSPQARPSWCLPQA|SLQKAHF|LPQNPY
protein 2:               YRTHPKPDHLRARPGARPKHDHPGAHSKPDHPRAHLRARPQPDHPGAHLRARPQPDHPRAHLRARPQPDHPGAHPNPDHRHKPDHPGVCHKPDHSKKHIFNYHKTRI

protein cDNA 0:          |YGFCGS|KCAFWSDQACGRHQDGRACGDGRDWGGLRGGRAGGGL|GGLWGGRAGGGL|GGLRGGRAGGGL|GGLWGGRAWSGLQGGHVWGGLRDGL|GGRAWGGFCI
protein cDNA 1:          DTGFVVVKNVLFGVIRLVADTRMVGLVAMVGIGVGSGVVGLGAGSEVGSGVVGLGAGSEVGSGVVGLGAGSEVGSGVVGLGVGSRVVMFGAGSGTGSEVVGLGVGSV
protein cDNA 2:          IRVLW|LKMCFLE|SGLWQTPGWSGLWRWSGLGWAPGWSGWGRALRWALGWSGWGRALRWAPGWSGWGRALRWALGWSGLEWAPGWSCLGRAPGRALRWSGLGWVLY

# class j
protein 0:               T|PPWSPLQARPPQSPPQSPPPARPPRSPPQSPPPARPPQSPPQSPPPARPPRSPPQSRPSARPPQSPPQSPPPARPPQSPPQTRCPPQA|SLQ
protein 1:               HDHPGAHSKPDHHRAHLRARPQPDHPGAHLRARPQPDPPRAHLRARPQPDHPGAHPNPDHQPDHPRAHLRARPQPDHPRAHLRPGVRHKPDHSK
protein 2:               MTTLEPTPSPTTTEPTSEPAPSPTTPEPTSEPAPSPTPPEPTSEPAPSPTTPEPTPIPTISPTTPEPTSEPAPSPTTPEPTSDPVSATSLITP

protein cDNA 0:          FWSDQACGGHRV|GGLWGGRAGGGL|GGLWGGRADGRDWGGLRGGRAGGGL|GGLWGGRAGGGL|GGLRGGRAGGGL|GGLCGGRAWSGLQGGH
protein cDNA 1:          FGVIRLVADTGSEVGSGVVGLGAGSEVGSGVVGLMVGIGVGSGVVGLGAGSEVGSGGVGLGAGSEVGSGVVGLGAGSEVGSVVVGLGVGSRVVM
protein cDNA 2:          LE|SGLWRTPGLRWALGWSGWGRALRWALGWSG|WSGLGWAPGWSGWGRALRWALGGSGWGRALRWAPGWSGWGRALRWALWWSGLEWAPGWS
```



```
# True amino-acid sequence
TTEPTPSPTTSEPVPEPAPNMTTLEPTPSPTTPEPTSEPAPSPTTPEPTSEPAPSP              TTPEPTSEPAPSPTTPEPTPIPTIATSPTILVSATSLITPKSTFLTTTKPVS

# class i translation
NTEPTPSPTTSEPVPEPAPNMTTLEPTPSPTTPEPTSEPAPSPTTPEPTSEPAPSP              TTPEPTSEPAPSPTTPEPTPIPTIATSPTILVSATSLITPKSTFLTTTKPVS

# class j translation
        MTTLEPTPSPTTTEPTSEPAPSPTTPEPTSEPAPSPTPPEPTSEPAPSPTTPEPTPIPTISPTTPEPTSEPAPSPTTPEPTSDPVSATSLITP
```

# Thank you!