**Design and Specification Barcodes:**

**Definitions:**

**MAPseq:** MAPseq is atechnique for high-throughput mapping of connections in the brain. Neurons are virally induced to produce random RNA sequences which are known as barcodes. These barcodes fill both the somas and their axons. The cells are then sequenced in both the injection region and any number of putative regions. If a neuron’s (from an identified from the injection region) unique barcode shows up in a putative region, then that neuron sends outgoing signals to the target region. MAPseq is then used to interrogate the signaling structure of many neurons simultaneously.

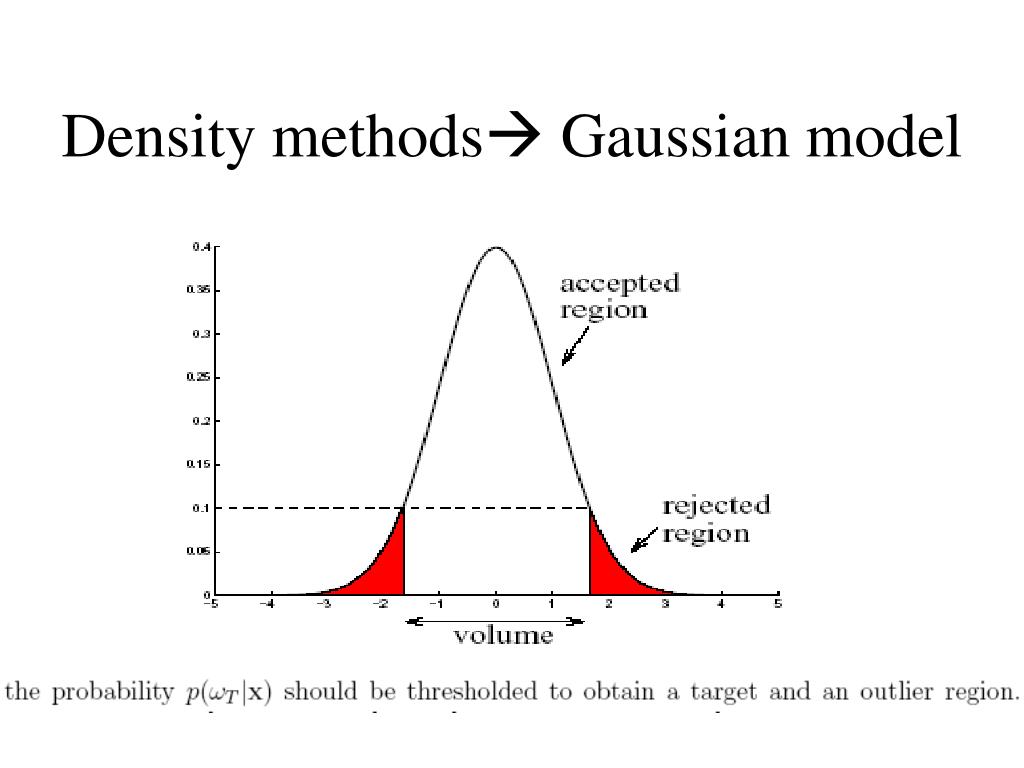
**Viral RNA Sequence:** A viral sequence a cell is infected with that contains an anchor sequence and a unique barcode that precedes this anchor sequence. This sequence is used to map neural projections.

**Anchor Sequence:** The anchor sequence is a known sequence within the viral RNA sequence. In this case it is: *GTACTGCGGCCGCTACCTA*

**Barcodes:** A unique sequence of bases that is produced by a biological process. In this instance it is a 30 base barcode.

**Whitelisted Barcodes:** A set of barcodes that pass the filtering process (identified true barcodes). In this program a list of whitelisted barcodes has not been provided and therefor they are determined in one of two ways. The first way in which true barcodes are determined is by setting a threshold frequency based on the number of cells sequenced and removing any barcodes that are below that frequency threshold. The second was is by creating a gaussian distribution in order to determine the most common barcodes (true barcodes) by finding the kernel-density estimate via Gaussian kernels. The second method is very conservative and therefor would likely rule out certain barcodes that would be true.

**Gaussian distribution (normal distribution):** a type of continuous probability distribution that is symmetrical about its mean. Most observations cluster around the mean and the further away an observation is from the mean, the lower its probability of occurring. The graph of a Gaussian function forms the characteristic bell shape of a Gaussian/normal distribution.



**Kernel Density Estimation:** a way to estimate the probability density function of a random variable in a non-parametric way.

**Quality Score (Q Score):** “The sequencing quality score of a given base, Q, is defined by the following equation:

Q = -10log10(e)

where e is the estimated probability of the base call being wrong.

* **Higher Q scores** indicate a smaller probability of error.
* **Lower Q scores** can result in a significant portion of the reads being unusable. They may also lead to increased false-positive variant calls, resulting in inaccurate conclusions.

A quality score of 20 (Q20) represents an error rate of 1 in 100 (meaning every 100 bp sequencing read may contain an error), with a corresponding call accuracy of 99%.

When sequencing quality reaches Q30, virtually all of the reads will be perfect, with no errors or ambiguities. This is why Q30 is considered a benchmark for quality in next-generation sequencing (NGS).”

**Hamming Distance:** Hamming code is a linear code that is useful for error detection. A hamming distance refers to the number of points at which two lines of code differ. In this case it is a way to determine how many bases are mismatched in an RNA sequence when comparing two sequences. In this program barcodes with a hamming distance of 1 from a whitelisted barcode will be determined in order to better filter the data.

**Specification**

**Purpose:** The purpose of the program is to take in a fastq file which contains RNA sequences and output a shorter list of true underlying barcodes.

**Description:** The program will take in a fastq file which contains RNA sequences. The sequences will be searched to see if they contain the anchor sequence *GTACTGCGGCCGCTACCTA* and if the sequence contains this anchor sequence, the 30 bases preceding this anchor sequence will be stored in a list of barcodes. Additionally, RNA sequences want to be filtered by quality score, the user can select the minimum quality score of a base they would like to ensure. The list of barcodes will then be analyzed to determine which barcodes meet the filtering criteria for whitelisted barcodes. Whitelisted barcodes will be determined in one of two ways. (1) Setting a threshold frequency based on the number of cells sequenced and removing any barcodes that are below that frequency threshold. (2) Performing a Gaussian density distribution on the frequency of barcodes within the barcodes list. If a barcode’s frequency is above the threshold determined by the Gaussian density distribution, then it is deemed very likely a true barcode. The first method is far more accurate, but the program will include the second in case the cell number is not known. Once the true barcodes (whitelist) have been determined based on statistical analysis of the frequencies of the barcodes, the barcodes that are hamming-1 from a whitelist barcode will be determined and stored within a dictionary along with the whitelist barcode it has diverged from. The output is either simply the list of true barcodes or both the list of true barcodes along with the dictionary storing the hamming-1 barcodes from the whitelisted barcodes.

**Assumptions:** All RNA sequences in the fastq file are of the same size and the barcodes have a length of 30 bases. Input will be a fastq file.

**Input/Output:** The input is a fastq file along with multiple parameters that must be set in order for the program to determine what type of analysis to perform. These parameters consist of:

1. file\_name: a string representing the file name of fastq file (required)
2. anchor\_seq: a string used to find the barcodes (required)
3. expected\_number\_cells: an integer representing the known number of cells sequenced (optional, default = 0)
4. quality\_score: an integer between 0 and 30 used to determine if a base meets the set quality score (0-30) (optional, default = -1)
5. *get*\_hamming\_one\_off\_codes: a boolean value that if set to true will have the program return a dictionary of barcodes that are hamming of one of a whitelisted barcode (optional, default = False)

The output is either both a list of true barcodes (whitelist) and a dictionary of all apparent barcodes that are hamming-1 paired with the whitelisted barcode they most likely diverged from or just the list of true barcodes depending on the *get*\_hamming\_one\_off\_codes input parameter.

**Testing:** The program is tested via taking in fastq input files and determining if the number of cells roughly matches the number of true barcodes. For example, when running the attached fastq file, the number of cells entered is 1000 and the output for number of true barcodes is 975. Additionally, print statements were laid throughout the program during development to ensure the frequency of barcodes was correct and the hamming-1 distance algorithm worked.

**Error Handling:** Error handling for file format will be included along with error handling when the start index for an anchor sequence can not be found. Additionally, error checking is performed on inputs as well.

**Scalability:** In the future this program can be made more extensible by running on multiple machines via AWS Fargate. DDB could be used for a distributed lock system in order to synchronize processes across multiple machines. Additionally, the hamming-1 function would benefit from implementing multi-processing in order to process larger files more quickly. I would do this by implementing the maps function and splitting up the barcodes array into a list of lists and passing that in. The number of processes would be set to the CPU count. Numpy was used for arrays in order to increase efficacity as well as a Counter object was used to determine the frequency of particular barcodes.

**Data Changes:** Different data will have different statistics and this program takes this issue into account as it determines the distribution of barcode frequencies each time for different data.

**Automation:** As long as the anchor sequence and fastq file are uploaded, automation should not be an issue.