**High Throughput Nanoparticle Screening using JOint Rapid DNA Analysis of Nanoparticles (JORDAN)**

Christopher M. Monaco1,2, TBD, James E. Dahlman1

1Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology and Emory University School of Medicine, Atlanta, Georgia, USA

2School of Biological Sciences, Georgia Institute of Technology, Atlanta, Georgia, USA

**Abstract**

# Introduction

The transport of foreign nucleic acids is carefully regulated *in vivo*, making systemic drug delivery inefficient1-3. Thousands of LNPs have been designed to deliver genetic drugs to on-target cells, and minimize delivery to off-target cells. LNP chemical diversity is imparted 2 ways. First, thousands of distinct biomaterials can be created using many chemistries4-9. Second, each biomaterial can be formulated into hundreds of different LNPs by adding poly(ethylene glycol) (PEG), cholesterol, 1,2-dioleoyl-*sn*-glycero-3-phosphoethanolamine (DOPE), and other constituents, using different mole ratios.

The field can currently synthesize nanomaterials at a rate several orders of magnitude higher than the rate at which we can test nanomaterials for drug delivery *in vivo.* Recently, we reported a nanoparticle DNA barcoding system18 to increase the number of LNPs we could study at once *in vivo*. We used a microfluidic device to barcode LNPs (**Fig. 1A**)19; each LNP was formulated to carry a unique DNA barcode. We pooled stable LNPs, administered them to animals, and deep sequenced the barcodes to quantify the delivery of up to 30 LNPs simultaneously18. This original paper focused exclusively on control experiments designed to characterize the system. Specifically, we demonstrated that barcoded LNPs can be made so they do not mix in solution, that DNA sequencing readouts were linear with respect to the amount of injected DNA, that DNA barcode delivery recapitulates the behavior of previously characterized LNPs, that delivery does not change with DNA sequence, and that delivery of DNA barcodes to hepatocytes *in vivo* modeled siRNA delivery to hepatocytes *in vivo*18.

We now report that the same LNP barcoding system, herein named JOint Rapid DNA Analysis of Nanoparticles (JORDAN), can elucidate fundamental questions about nanoparticle delivery. *[talk about number screens than can be performed]*

# Experimental Design

## Rational Design of Barcodes

The DNA barcodes were designed rationally with several characteristics, as we previously described10. We purchased 56 nucleotide single stranded DNA sequences from IDT (**Fig. 1c, Supplementary Fig. 1a**). We included 2 universal 21 and 20 nucleotide primer regions in addition to a random 7 nucleotide (‘7N’) region that is unique to each piece of DNA (**Supplementary Fig. 1b**). Barcodes were distinguished using an 8 basepair (bp) sequence in the middle of the barcode. An 8 bp sequence can generate over 1,000,000 (48) unique barcodes; we selected 240 barcodes to prevent sequence bleaching on the Illumina MiniSeqTM machine. The 2 nucleotides on the 5’ and 3’ ends of the 56-nucleotide ssDNA sequence were modified with phosphorothioate linkages to reduce exonuclease degradation and improve DNA barcode stability.

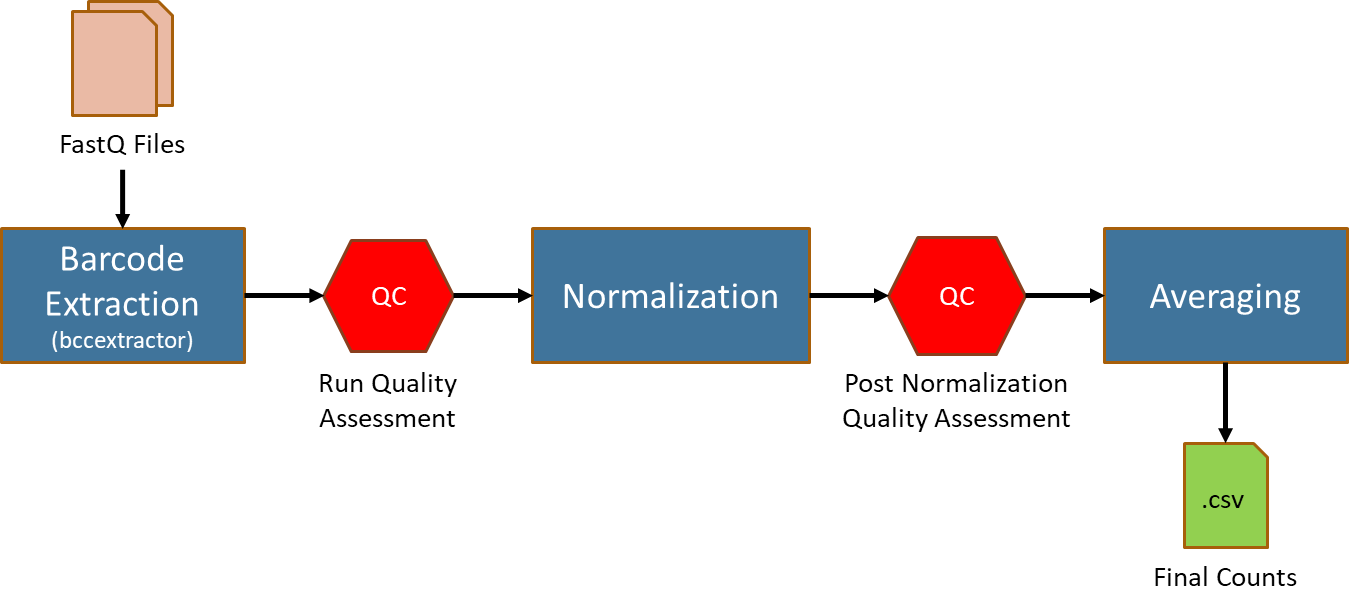
## Sequencing

Details about sequencing. Illumina MiniSeq. Barcode designed based on the Nextera XT kit. Use 1x75

## Sequencing Analysis and Counting Barcodes

### Overview

We have developed a bioinformatics pipeline to quickly and reliably extract barcode counts from sequencing data. We use custom software tools and scripts to collect counts from Illumina FastQ files and then clean and normalize those counts to result in a comma separated values file ready for downstream analysis. An overview of the pipeline is shown visually in **Figure FC**.

**

*[consider using expanded flow chart]*

A software tool called BarCode Counts Extractor (BCCE) is used to extract raw counts from a series of FastQ files using an experimental barcode library file. In addition to counts, the tool generates statistics used to assess run quality. Following a successful quality assessment, the counts are normalized and then once again assessed for quality and potential problems with individual replicates or barcodes. Finally, replicates are averaged together. All tools and scripts are available for download on our lab’s website, [www.dahlmanlab.org](http://www.dahlmanlab.org).

### Limitations

# Materials

.

# Procedures

## **Nanoparticle Formulation**

Nanoparticles were formulated in a microfluidic device by mixing DNA with lipomer, PEG, cholesterol, and a helper lipid, as previously described5,10,13-20

LNP hydrodynamic diameter was measured using high throughput dynamic light scattering (DLS) (DynaPro Plate Reader II, Wyatt). LNPs were diluted in sterile 1X PBS to a concentration of ~0.0005 mg/mL, and analyzed. LNPs were included if they met 3 criteria: diameter >20 nm, diameter <300 nm, and autocorrelation function with only 1 inflection point. Over the course of our experiments, ~60% of the LNPs we formulated met all 3 criteria (**Supplementary Fig. 1k, 1l**). Particles that met these criteria were pooled and dialyzed with 1X phosphate buffered saline (PBS, Invitrogen), and were sterile filtered with a 0.22 μm filter.

## **Pooling and Injection**

Again, can probably come from a paper.

## **Sorting/PCR**

All samples were amplified and prepared for sequencing. More specifically, 1μL of primers (5uM for Final Reverse/Forward, 0.5uM for Base Forward) were added to 5 μL of Kapa HiFi 2X master mix, 3 μL sterile H2O, and 1 μL DNA template. The reaction was run for 30 cycles. When the PCR reaction did not produce clear bands, the primer concentrations, DNA template input, PCR temperature, and number of cycles were optimized for individual samples. PCR amplicon was isolated with gel extraction.

## **Library Prep and Sequencing**

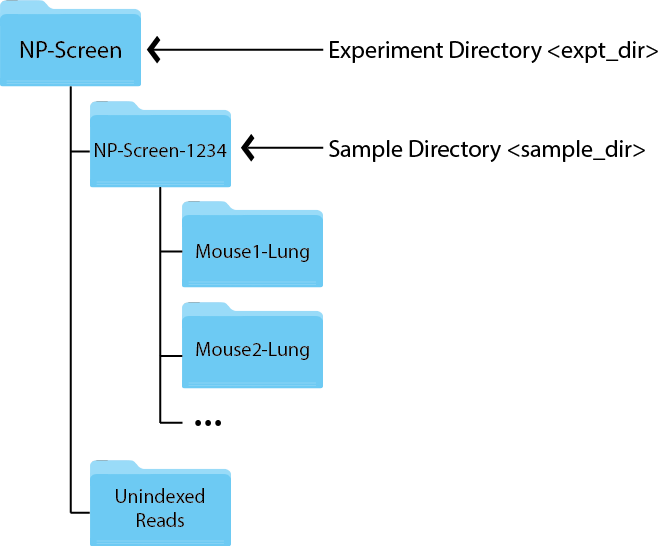
Illumina MiniSeq. Nextera XT. 1x75

## **Sequencing Analysis and Counting Barcodes**

### 1. Sequencing Data Acquisition and Organization

Your lab or sequencing core will have procedures in place for acquiring raw sequencing results. This document assumes that sequencing data has already been acquired or has been made accessible.

Illumina stores sequencing read files in a similar directory structure as shown in **Figure F**. The Experiment Directory contains all files and subdirectories associated with a given experiment. Within the experiment directory, a sample directory contains directories for each indexed sample on your sequencing run. In the case of paired-end reads, both the forward (R1) and reverse (R2) reads are stored within the indexed subdirectory. The Unindexed-Reads directory contains a directory of FASTQ files with sequences corresponding to reads that were unable to be indexed, either due to read errors or controls placed on the sequencing run.



### 2. Generating Raw Barcode Counts

Raw barcode counts are extracted with the BarCode Counts Extractor (BCCE) tool available on our website at http://dahlmanlab.org. This tool is written in Python and can be run from the command line on any computer running Python 2.7+.

#### 2.1 Barcode Library Preparation

BCCE requires a library of barcodes used in the experiment to be provided in a two column, tab-delimited text file. The first column of this file should contain a barcode name or unique barcode identifier and he second column should contain the barcode sequence written 5'-3', capitalized, and without whitespace. There should be no headers. An example of the file contents would be as follows

BC1 GACACAGT

BC2 ACAGAGGT

BC3 CCACTAAG

BC4 GATACCTG

BC5 AGCCGTAA

BC6 CTCCTGAA

...

This file can be located anywhere, however, it is recommended the file is stored in the Experiment Directory.

#### 2.2 The Barcode Counts Extractor Tool

BCCE takes several arguments as shown below:

bccextractor [-h] [-v] -l <lib.txt> -i <indir> -o <out.csv> [-s <int>] [-b <int>] [-p <str>] [-t –threads]

* Required Arguments:

|  |  |
| --- | --- |
| -l, --lib | Experimental barcode library file. This is the tab-delimited file discussed in section 3.1 of this document. |
| -i, --inDir | Input directory containing FastQ files to be processed. Typically this will be the Sample Directory. |
| -o, --outFile | Output file name ending with a CSV extension. |

* Optional Arguments:

|  |  |
| --- | --- |
| -s, --start | Barcode start location. Describes which nucleotide, starting from the 5' end as 1, the barcode is expected to start on. The default values is set to 55. |
| -b, --length | Barcode length. The default value is set to 8. |
| -p, --prefix | File name prefix to drop in output file sample names. This option can be used if the sample names in a sequencing run all contain an identical prefix, i.e. 170514\_MySample\_Hw-64.fastq.gz contains the prefix "170514\_MySample\_", however, in our output sample naming we are only interested in "Hw-64". |
| -t, --threads | Number of processing threads used during execution. The default values is set to 8. |

* Information Arguments:

|  |  |
| --- | --- |
| -v, --version | Show program’s version and exit. |
| -h, --help | Show help message and exit. |

The tool will generate four files in the directory to which the output file is to be written. The first is a CSV file containing raw counts where columns are sample names extracted from the FASTQ files and rows are the barcode sequences provided in the barcode library file. The second is a log file showing the number of sequences processed versus the number of barcodes identified and the mean quality score in the barcode region per sample. The remaining two files are plots of the log file providing the user with a quick way of visualizing the quality of the run. Representative plots are shown in **Figure Y**.

#### 2.3 Running the Barcode Counts Extractor Tool

Once prerequisites are in place, running BCCE is accomplished by simply specifying the correct arguments. For the directory structure in Figure 1, running BCEE would take the following form:

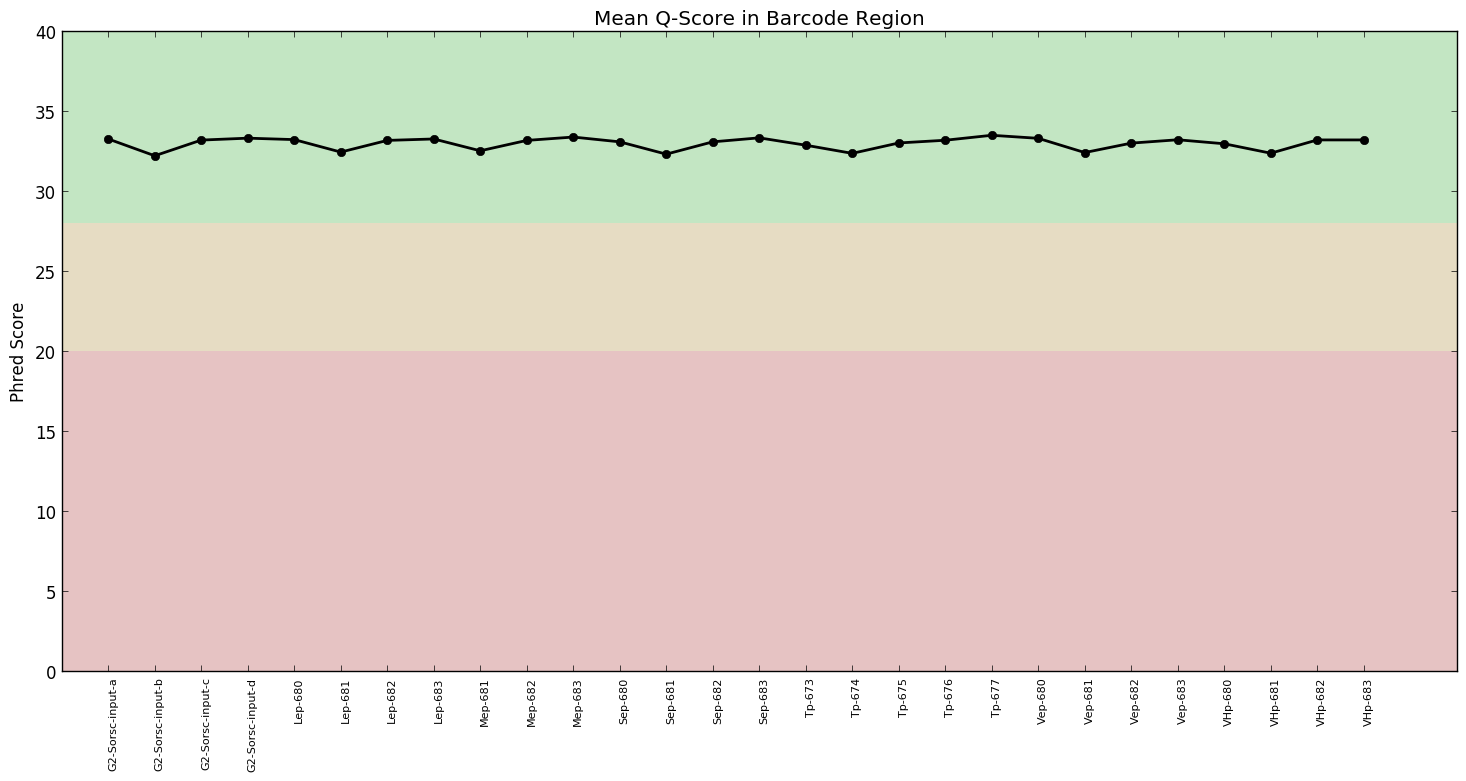
bccextractor -l bclib.txt -i NP-Screen-1234/ -o raw\_counts.csv

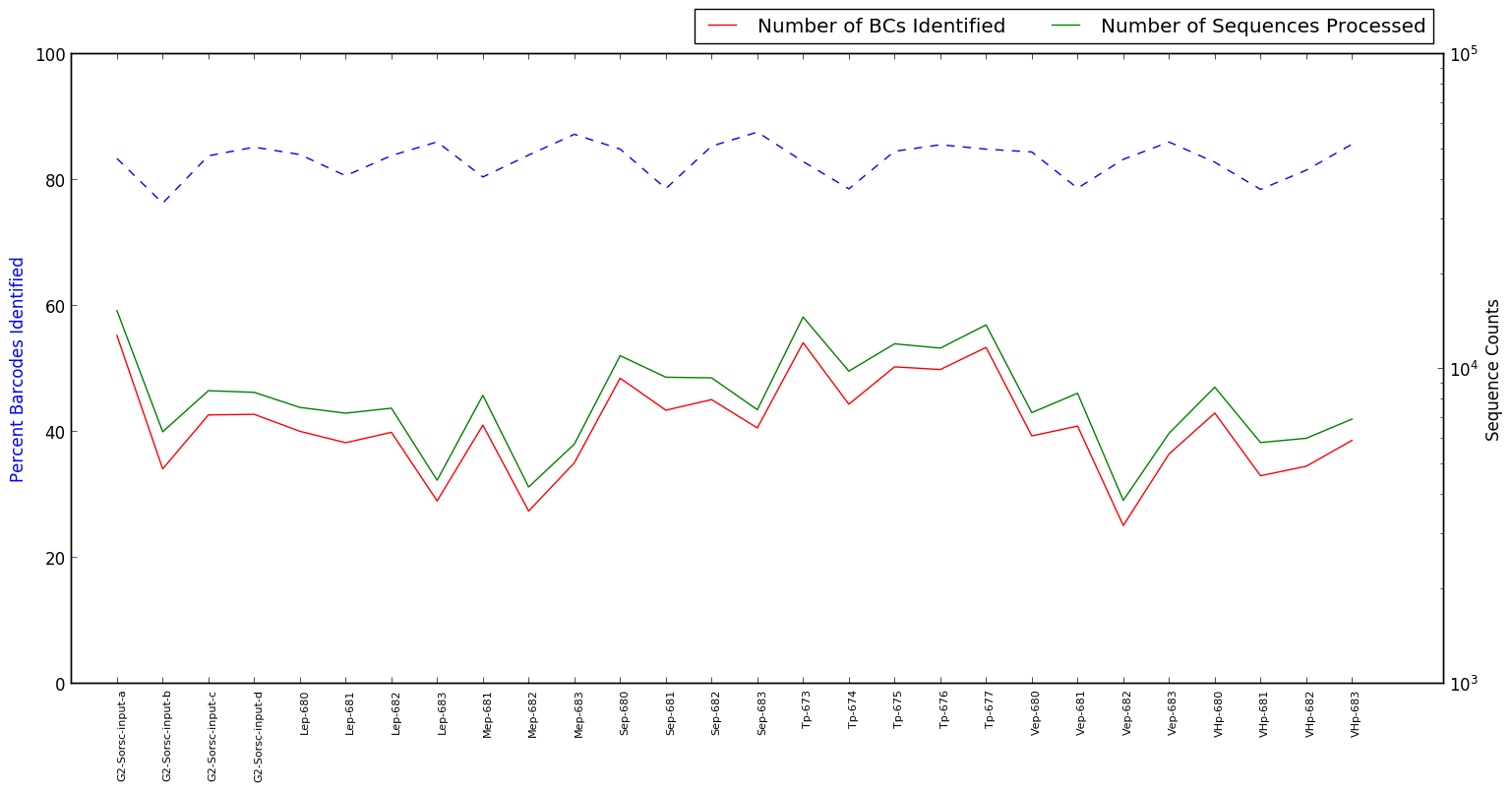
Since the current barcode design will yield reads with barcode location and length of 55 and 8, respectively, no optional arguments are required. We have designed to take location and length arguments optionally as to be flexible enough to use with any barcode design.

As the tool processes each FASTQ file, the file name will be displayed in the terminal. Once processing is complete for all samples, the tool will report Complete! and two new files should appear in the location specified by the output directory argument: the counts file and log file.

### 3. Assessing Run Quality

This is the first of two quality control steps. Here, we visually examine the output of the BCCE tool to ensure we have a high average quality score (Q-Score) in the barcode region and that there were a high percentage of barcodes identified. Figure Ya shows a representative plot of the mean quality score in the barcode region of the sequence. As with most Illumina sequencing, we desire a Phred quality score of at least 28 to consider the data to be of high quality. Our extraction method is robust to low quality data so if a sample’s mean q-score in the barcode region is low, a low number of counts will be returned and this sample is likely to be discarded in later quality control steps. We also look at percentage of barcodes identified from total sequences in each sample as well as the absolute number of identified barcodes and sequences. In this way, we can further assure we have a good sequencing run by seeing that both our sequencing depth and the number of identified barcodes is high.

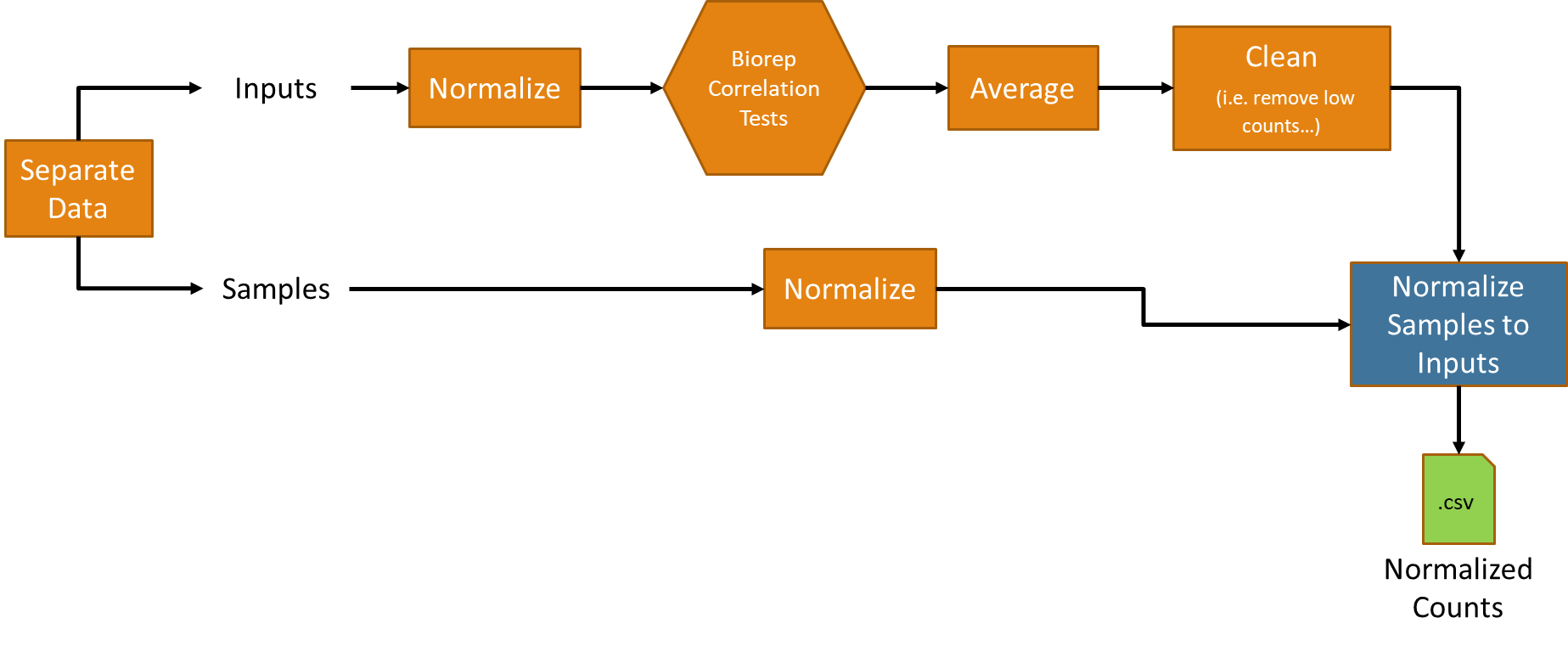




### 4. Normalizing Counts

Once extracted, raw barcode counts must be normalized across samples and back to the injection inputs.

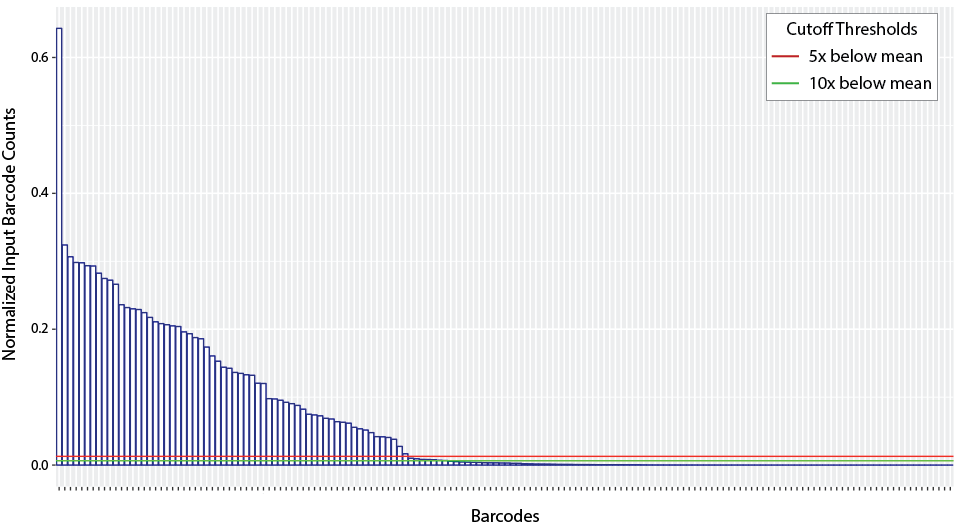
#### 4.1 The Normalization Process



**4.1.1 Normalization of Injection Inputs**

The first step in normalizing raw counts data is to normalize the injection inputs. There are typically three to four injection input biological replicates. Each replicate is normalized to unity by dividing each barcode's count by the sum of the counts per replicate. We then conduct a pairwise Pearson Correlation test to identify any inputs not highly correlated with each other. We expect high correlation between biological replicates and so inputs showing low correlation will be removed from the experiment. The normalized replicates are then averaged together to give a single normalized injection input.

Following injection input normalization and averaging, relative average input barcode counts are plotted on a bar chart. We typically see a distribution similar to what is shown in **Figure B**.

****

The horizontal lines show thresholds at the average of all the counts divided by five and ten (effectively five and ten times below the mean). We will qualitatively choose a threshold that appears to best remove the lowest count barcodes while keeping as many barcodes as possible.

**4.1.2 Normalization of Samples**

Once low-count barcodes are removed, samples can be normalized. First, the injection input biological replicates are re-normalized and averaged to account for changes due to the removal of low-count barcodes. Samples are then normalized in the same manner by dividing each sample’s barcode counts by the sum of each sample's barcode counts. These normalized counts are then normalized back to the injection input by dividing each sample by the injection input. Lastly, each sample is scaled such that the sum of the counts in each sample is 100.

#### 4.2 Normalization Script

Normalization is carried out in R using an R Markdown document. A template of this document is available from the Dahlman Lab GitHub [page](https://github.gatech.edu/Dahlman-Lab/Normalization-Template). The template walks through the steps of generating the normalization report and provides example code.

### 5. Post Normalization Quality Assessment

The second quality control step will identify sample biological replicates that show low correlation and should not be averaged together. In addition, this step helps to identify run-away barcode sequences.

#### 5.1 Bio-replicate Pairwise Correlation Studies

As with the inputs, we conduct pairwise Pearson correlation tests for every biological replicate group. Low correlation replicates are considered for removal from averaging. [figure needed].

#### 5.2 Run-away Identification

Run-away barcode sequences are those which show up with disproportionately high number of counts across a large number samples in a given sequencing run. There are several possible reasons as to why run-aways may occur... In order to help identify run-aways, we employ two statistical tests. First, we conduct single-linkage hierarchical clustering of the barcodes. Any outlying barcode(s) will appear as out groups clustering with a large distance between the rest of the barcodes. We then conduct a principle components analysis of the barcodes. Again, outlying barcode barcodes will appear as individual points not clustering with the rest of the barcodes.

It is important to note that identification of run-aways are subjective and the quantification of what constitutes a run-away can vary from experiment to experiment. Clustering and PCA are only meant to help the experimenter easily flag potential run-aways for further examination.