Barcode Caddy

The problem of selecting appropriate barcode sets for pooling samples is oft ignored. This could have an impact on experiments and conclusions reached, because it determines the deconvolution of sequencing results. There are two problems that can arise from bad choice of barcodes,

· An unfortunate choice of barcodes can lead to reads being assigned to the wrong sample, leading to *contamination*. This happens when the barcodes are not sufficiently different from each other, so that sequencing errors in barcodes makes the assignation of read to sample ambiguous.

· Lack of diversity of nucleotides at each position in the pool of barcodes can lead to loss of samples. In sequencing systems based on imaging, the lack of diversity at a position (e.g. mostly A's in position 1), can present a unique challenge to image segmentation routines, which determine the location of clusters, resulting in a loss of cluster identification. This can lead to the read being *orphaned*. So paradoxically, even though the read sequencing can be of high quality, poor barcode read quality makes the read unusable.

Barcode Caddy considers both issues and makes a reasonable choice of barcodes, from sets selected by the user, to maximize yield and minimize errors. Barcode Caddy enables selection of a good subset from sets of readymade barcodes, including sets from frequently used kits from various manufacturers, as well as custom sets that the user might have in the lab.

Sequencing instruments based on imaging use light emitted by a newly added base, upon excitation by a laser, to build the sequence one base at a time. This involves imaging the flow cell after addition of each nucleotide, and the imaging system needs to detect where the clusters are located, in each image. This is done using segmentation algorithms and information on the location of clusters gleaned from images taken in the preceding cycles. If there is a bias in the nucleotides at a position, then there is not enough distinction between neighbors, making it difficult to distinguish clusters from each other. Thus, the first few cycles are critical in determining the locations of clusters and starting out with a low diversity of sequences (such as say 70% A's and 10% each of G, C and T) in the library can lead to low yields. Thus, sequencing amplicons can be problematic and a random library from phiX is spiked in such libraries, which consumes the sequencing space, but overcomes the diversity issue and enables high quality reads to be generated.

Next generation sequencing nowadays mostly implies sequencing on the Illumina platform. Cost per base and large volumes of high-precision data per run are the main reasons Illumina is preferred over other platforms.

Efficient use of the large capacity of each run requires **multiplexing,** allowing multiple samples to be run simultaneously. This is achieved through the use of unique sequences, called **barcodes** (typically 6-12bp), specific to each sample, that are added to each DNA fragment during library preparation. Libraries from different samples now can be pooled and sequenced simultaneously during a single sequencing run. The barcode for each sequenced read is identified to bin the reads into samples (**demultiplexing)** before final data analysis. Typically, the barcode is read separately as part of the index read cycle.

While the use of barcodes has proven to be popular, as it has reduced costs and increased speed of sequencing multiple samples, there are a few flies in the ointment. Because barcodes are the only means of identifying the reads from different samples, a mis-identification of a barcode can have serious consequences, leading to contamination of samples, and a reduction in read depth per sample. There are several issues to be considered in picking barcodes;

DIVERSITY of barcodes: The process of segmentation is even more fraught in the case of barcodes since they are short (usually < 10 nucleotides), there is not enough room (in terms of sequencing length) to build a model and correct mistakes, so diversity of barcodes in the first few bases is critical to get good data and not lose reads to missing/low-quality barcodes.

INDEX DISTANCE: Even though Illumina platform produces high precision sequencing reads, one has to be prepared for insertions, deletions, and substitutions events. . If the differences between the barcode sequences are less than 2, then a single error can lead to confusion about the correct assignment of reads to samples. If the differences are 3 or greater, then a single sequencing error still allows correct assignation of reads to samples.

Therefore, a minimum distance between barcodes is essential for accurate demultiplexing

INDEX HOPPING: Broadly, this refers to the potential for the wrong barcode to get inserted into samples. The potential for index hopping is present regardless of the library prep method or sequencing system used. The worst cases are when the sequencing instrument allows this to happen while clusters are being generated, there is no way to correct this post-sequencing.

We can broadly classify the sources of errors with barcodes into

1) Physical Errors.
    a. Cluster overlaps leading to the wrong barcode being assigned to a read[1].
    b. PCR artifacts leading to incorrect barcodes being sequenced in a cluster, a process termed index-hopping[2]
2) Information Errors.
    a. Sequencing errors in barcodes leading to incorrect sample assignments
    b. Low diversity at certain positions across barcodes, leading to low quality barcode sequences and low yields

Physical errors, once made, are difficult to correct, but there are strategies to reduce them, using either dual-indexing (indexing both adaptors), which can reduce errors and using dual-indexed phiX libraries which can reduce errors from index-hopping on the sequencing machine (this is from a comment in BioRxiv).

Information errors, as the name implies, can often be corrected in software, provided the barcodes are selected with care. Proper selection of barcodes using ideas from coding/information theory, is the best way to further ensure accuracy and sensitivity.

Barcode Caddy enables selection of a good subset from a sets of readymade barcodes. Bear in mind, there exist several tools for the creation of new barcode sets[3,4]. Our goal is to solve the practical problem faced frequently by labs and sequencing cores that need to mix and match libraries from a variety of sources and kits.

The key in selecting barcodes is to maximize the separation (distance) between all possible pairs in the set. Obviously, as the number of barcodes increases, the problem of ensuring an appropriate set becomes progressively harder. There are two ways in which a distance can be defined between barcodes, Hamming and Levenshtein. Hamming is intuitively obvious; a difference at each position adds 1 to the distance and the greater the distance, the more unlikely it is that one barcode will be mistaken for another one. But take the case of two barcodes, CAGCT and AGCTC, the distance between the pair is 6 according to Hamming, since each position is different, but it is obvious that one is obtained from the other by a deletion at one end and an addition of one letter to the other end (CAGCT -> AGCT -> AGCTC) and this can happen on a sequencing instrument. This would pair would have a short Levenshtein distance (2), and is more useful to avoid overestimating the distance. But some of the changes in the Levenshtein sense maybe be physically extremely rare, based on the error model of the sequencer. For example, ACCGTA and ACGTTA are a distance of two apart in the Levenshtein sense, but in reality this kind of error, which involves one deletion at position 2, and and insertion in position 5, is very difficult to imagine happening, so we would penalize this extra (in

sequence alignment sense this would give a different weight to say an insertion than a deletion), and would automatically increase the distance in this case.
Ideally, we want a metric that accounts for the kinds of errors a sequencing machine can make. The greater the hamming distance, the lower the chance of random errors leading to a difficulty in assigning the read to the wrong sample. So we use a combination of the two measures to ensure appropriate separation of barcodes.

But with a barcode of length 6, which allows for 4096 distinct barcodes, it becomes difficult to ensure that a set have all pairwise distances be 3 or above. Especially as the number of barcodes goes up. Using only distances can results in picks of this sort,
ACGTGA,
AGCACT
ATTGCA
AAACAG

Which are all quite distant from each other, but have an A at the start. To avoid this, we can weight the distance metric, so that differences in the first position carry more weight than differences in the second position etc. This would automatically avoid repeating nucleotides at the first few positions as much as possible.

That is what we do, modify the levenshtein to penalize insertions a little extra and increase the value of differences at the start of the sequence, at the expense of the end of the sequence. These two tricks enable construction of barcode sets that are relatively immune to sequencing errors.

Ignoring these can lead to contaminated samples, as well as reduced output, resulting in inaccuracies, loss of data and/or increased costs. While many tools help the end user generate custom barcodes, very few allow the selection of subsets from pre-synthesized set, addressing the problems explained above[1,2]. The **Barcode Caddy** (http://girihlet.com/tools.html) is an easy-to-use, web-browser based, tool that enables selection of sets. The calculation of distance between pairs of barcodes is modified to account for things like rarity of certain changes (like insertions,, translocation being rare) and tries to balance the nucleotide diversity at start and keep them sufficiently far apart to avoid collisions.

KEYWORDS: Double indexing[1], Levenshtein barcode design [5], index-hopping[2], R tool for barcode design[3],python code for barcode design[4], dual-index phiX[6], Random barcodes[7]

1. https://bioconductor.org/packages/3.7/bioc/manuals/DNABarcodes/man/DNABarcodes.pdf
2. http://comailab.genomecenter.ucdavis.edu/index.php/Barcode_generator
3. https://www.biorxiv.org/content/early/2017/04/09/125724

References:

1. Kircher, M., Sawyer, S. & Meyer, M. Double indexing overcomes inaccuracies in multiplex sequencing on the Illumina platform. *Nucleic Acids Res.* **40,** e3 (2012).

2. Sinha, R. *et al.* Index Switching Causes “Spreading-Of-Signal” Among Multiplexed Samples In Illumina HiSeq 4000 DNA Sequencing. *bioRxiv* 125724 (2017). doi:10.1101/125724

3. Buschmann, T. DNABarcodes. *Bioconductor* Available at: http://bioconductor.org/packages/DNABarcodes/. (Accessed: 9th May 2018)
4. Comai, L. Barcode generator - Comaiwiki. Available at: http://comailab.genomecenter.ucdavis.edu/index.php/Barcode_generator. (Accessed: 9th May 2018)
5. Buschmann, T. & Bystrykh, L. V. Levenshtein error-correcting barcodes for multiplexed DNA sequencing. *BMC Bioinformatics* **14,** 272 (2013).
6. Seqmatic. TailorMix Dual-Indexed PhiX Control Library (Denatured). *SeqMatic*
7. Ogawa, T., Kryukov, K., Imanishi, T. & Shiroguchi, K. The efficacy and further functional advantages of random-base molecular barcodes for absolute and digital quantification of nucleic acid molecules. *Scientific Reports* **7,** 13576 (2017).