Best Practices in Designing, Sequencing and Identifying Random DNA Barcodes

Milo S. Johnson[1,*], Sandeep Venkataram[2,*], Sergey Kryazhimskiy[2,†]

[1]Department of Integrative Biology, University of California Berkeley, Berkeley, CA 94720

[2]Department of Ecology, Behavior and Evolution, University of California San Diego, La Jolla, CA 92093

[*]Equal contribution

[†]Corresponding author: skryazhi@ucsd.edu

Abstract

Random DNA barcodes are a versatile tool for tracking cell lineages, with applications ranging from development to cancer to evolution. Here we review and critically evaluate barcode designs as well as methods of barcode sequencing and initial processing of barcode data. We first demonstrate how various barcode design decisions affect data quality and propose a new optimal design that balances all considerations we are currently aware of. We then discuss options for the preparation of barcode sequencing libraries, including inline indices and Unique Molecular Identifiers (UMIs). Our main conclusion is that the utility of inline indices is high whereas that of UMIs is low. Finally, we test the performance of several established and new bioinformatic pipelines for the extraction of barcodes from raw sequencing reads and for error correction. We find that both alignment and regular expression-based approaches work well for barcode extraction, and that error correction pipelines designed specifically for barcode data are superior to generic ones. Overall, this review will help researchers approach their barcoding experiments in a deliberate and systematic way.

1 Introduction

Observing how clonal populations of cells change over time is key to many problems in evolution, development, cancer, and other fields. Until recently, tracking cell lineages was a slow and labor-intensive process (Conklin, 1905; Serbedzija *et al.*, 1989; Holland & Varmus, 1998; Kretzschmar & Watt, 2012; Hsu, 2015). Recent advances in genetic engineering and nucleic acid sequencing technologies spurred the development of a new generation of high-throughput lineage tracking methods based on DNA "barcodes" (Blundell & Levy, 2014; Woodworth *et al.*, 2017; Kebschull & Zador, 2018; Baron & van Oudenaarden, 2019; Masuyama *et al.*, 2019; Wagner & Klein, 2020; Dujardin *et al.*, 2021; VanHorn & Morris, 2021). In these approaches, individual cells are tagged with unique genetic markers called "barcodes". Many thousands of cell lineages carrying different barcodes can be tracked within a population over multiple generations using high-throughput sequencing. Although barcode lineage tracking (BLT) techniques are fairly nascent, they have already found many applications, e.g., for characterizing T-cell recruitment (Schumacher *et al.*, 2010), tracing cellular differentiation over the course of organismal development (McKenna *et al.*, 2016; Frieda *et al.*, 2017; Alemany *et al.*, 2018; Wagner *et al.*, 2018; Weinreb *et al.*, 2020), studying the clonal history of metastasis in cancer (Bhang *et al.*, 2015; Wagenblast *et al.*, 2015; Roh *et al.*, 2018; Gutierrez *et al.*, 2021; Umkehrer *et al.*, 2021; Fennell *et al.*, 2022), screening and characterizing mutant libraries (Giaever *et al.*, 2002; Bell *et al.*, 2014; Wetmore *et al.*, 2015; Johnson *et al.*, 2019; Li *et al.*, 2019; Schubert *et al.*, 2021), identifying the provenance of microbial strains (Qian *et al.*, 2020), and studying evolutionary dynamics (Levy *et al.*, 2015; Al'Khafaji *et al.*, 2018; Cira *et al.*, 2018; Nguyen Ba *et al.*, 2019; Fasanello *et al.*, 2020; Jasinska *et al.*, 2020). With such rapid growth, many methods have been developed for designing, sequencing and identifying barcodes in the raw sequence data. Multiple labs have independently developed their own BLT procedures without necessarily evaluating pros and cons of other methodologies. Here, we review various existing approaches to BLT experiments and identify some of the best practices for generating and reading barcodes.

BLT studies fall into two modalities (Woodworth *et al.*, 2017; Kebschull & Zador, 2018; Baron & van Oudenaarden, 2019). *Retrospective* studies, which are typically carried out in the context of development, infer the lineage history of a population of cells based on naturally occurring somatic genetic variation at highly mutable loci, such as microsatellites, that can be viewed as barcodes (e.g., (Reizel *et al.*, 2011, 2012). In *prospective* studies, random DNA barcodes are introduced into an organism by the experimentalist to observe future changes. Barcode diversity is usually generated *in vitro*, i.e., before the barcodes are integrated into the organism's genome (e.g., (Giaever *et al.*, 2002; van Heijst *et al.*, 2009; Bhang *et al.*, 2015; Levy *et al.*, 2015; Johnson *et al.*, 2019; Eyler *et al.*, 2020; Ge *et al.*, 2020). More recent methods have also been

developed that integrate a targeted-mutagenesis module into the organism which then generates barcode diversity at the barcode locus *in vivo* (e.g., (Peikon *et al.*, 2014; McKenna *et al.*, 2016; Frieda *et al.*, 2017; Kalhor *et al.*, 2018; Raj *et al.*, 2018; Spanjaard *et al.*, 2018; Chan *et al.*, 2019). In this review, we focus on DNA barcodes used for prospective lineage tracing, with a specific focus on *in vitro* barcoding approaches, although some of the discussion will be relevant to other cases as well.

Early prospective lineage tracking studies generated and engineered barcodes into individual strains (e.g., different deletion mutants) and then pooled them for the tracking experiment (Giaever *et al.*, 2002; Smith *et al.*, 2009). Today, pools of barcoded strains are typically generated by transforming populations of cells in bulk with libraries of constructs that contain a diversity of DNA barcodes. The number of distinct cell lineages in such pools can range from hundreds (Cira *et al.*, 2018; Fasanello *et al.*, 2020) to millions (Bhang *et al.*, 2015; Umkehrer *et al.*, 2021). A barcoded population is then sampled at one or more timepoints, and the PCR-amplified barcodes are sequenced, typically on the Illumina platform. The relative abundance of each barcode at each timepoint can be estimated from these data, which can then be used for downstream analysis e.g. estimating mutant enrichment over the course of the experiment.

Researchers who seek to use *in vitro*-generated barcodes for prospective lineage tracking face a number of choices with respect to barcode design, sequencing and barcode identification. These include questions regarding barcode length and base composition, strategies for barcode amplification, methods for extracting barcodes from raw sequencing data as well as methods for error correction. Previous studies have implemented a variety of solutions to each of these problems, but we are unaware of any systematic review or comparison of various approaches. Here we review current practices in barcode design, sequencing and identification, discuss the implications of various choices, and identify current best practices for designing and conducting lineage tracking experiments using DNA barcodes. In the Appendix, we also briefly discuss a related problem of high-throughput genotyping of clones at a barcode locus.

2 Barcode design, synthesis and integration

Designing DNA barcodes involves a number of decisions. How long should the barcode locus be? What should be its base composition? Where in the genome will it be integrated? etc. These choices can have various downstream implications, e.g. for the number of lineages that can be tracked, for the fidelity of barcode amplification and sequencing and for the accuracy with which lineage frequencies can be estimated. In this section, we discuss some design considerations for the barcode locus itself (Section 2.1) as well as some practical decisions involved in the construction of a barcoded strain library (Section 2.2).

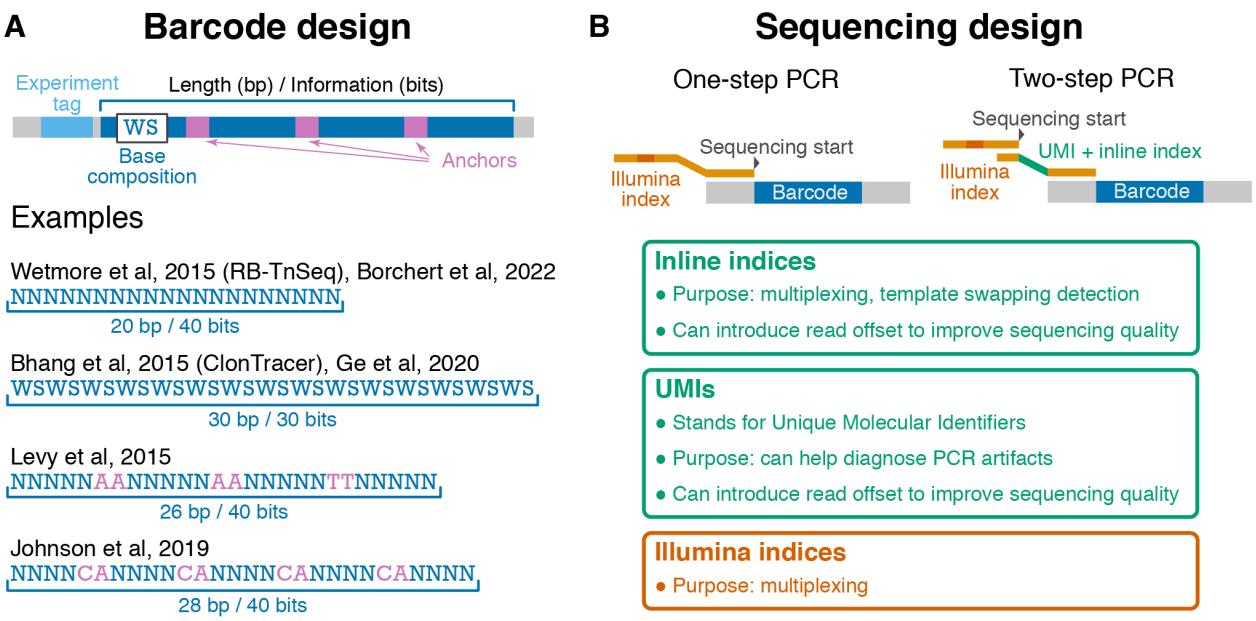

Figure 1. Barcode and sequencing design considerations. A. Structure of the barcode locus and examples of published barcode designs. **B.** Two commonly used barcode amplification strategies, one-step PCR (left) and two-step PCR (right). Key features on the primer sequences are indicated and explained in boxes. The optional experiment tag region on the template DNA is not shown for clarity. Note that in some one-step PCR strategies, inline indices with offsets are included, and sequencing starts at a similar location as in the two-step PCR strategy.

2.1 Structure of the barcode locus

In essence, barcodes are simply random sequences of nucleotides. Most DNA synthesis companies offer an option of including random nucleotide bases into oligonucleotide sequences. Such “barcode” oligos are chemically synthesized and then incorporated into plasmids and/or directly into the genome. In this section, we discuss only the structure of the barcode locus itself and leave out the discussion of other parts of the oligos that may be necessary for engineering and sequencing purposes, such as the presence of PCR priming sites.

The simplest barcodes can be formed by a sequence of random nucleotides, i.e., a sequence of “N”s in the oligo design (see Wetmore et al, 2015 design in Figure 1A). Other barcode designs feature short constant “anchor” sequences that break up “variable” regions (see Levy et al, 2015 and Johnson et al 2019 designs in Figure 1A) or consist of alternating random bases that are constrained to be strong (“S”, i.e. G or C) or weak (“W”, i.e. A or T; see Bhang et al, 2015 design in Figure 1A). We show below that some designs produce barcodes that are less likely to exhibit extreme GC-content or long repetitive regions (e.g. “AAAAAA”), two features that can lead to high frequency of errors or biases associated with PCR amplification and sequencing. We then discuss the considerations that determine the length of the barcode and describe our recommended barcode sequence. We conclude this section with a brief discussion of

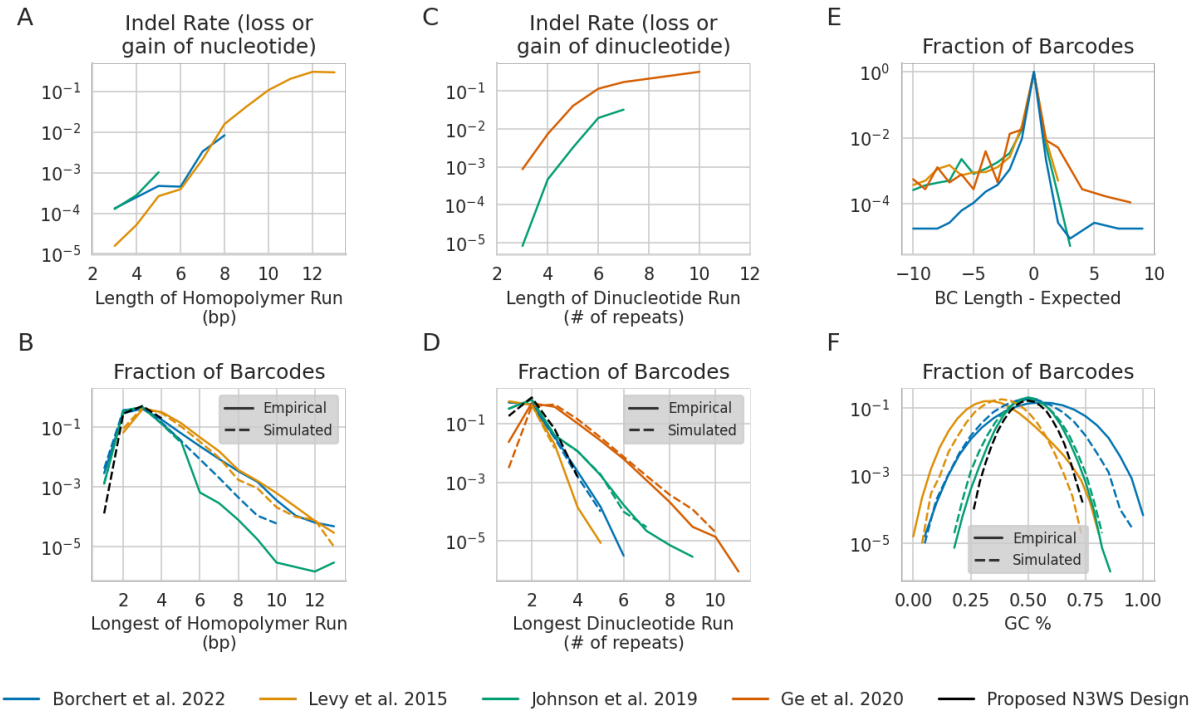

Figure 2. Barcode design features and error rates. (A) The total indel error rate in homopolymer runs, estimated from barcode data from four datasets. **(B)** The frequency of homopolymer runs of different lengths in the empirical and simulated datasets of barcodes with different designs (see Methods for details). **(C)** The total indel error rate in dinucleotide runs, estimated from barcode data from three datasets. **(D)** The frequency of dinucleotide runs of different lengths in the empirical and simulated datasets of barcodes with different designs. **(E)** The distribution of barcode lengths in each empirical dataset, using barcodes with at least 20 reads (see Methods for details). **(F)** The distribution of GC content in barcodes in the empirical and simulated dataset. The barcode designs are shown in Figure 1 and Table S1.

"pre-multiplexing", a way of leveraging barcode design to reduce labor and material costs at the library preparation stage.

Anchors and GC content control

The sequence of the barcode matters. To demonstrate this, we reanalyzed data from six barcode sequencing datasets (Table S1). We found that the empirical indel error rate increases exponentially with homopolymer run length (Figure 2A) and with dinucleotide run length (Figure 2C). For runs with more than 10 repeats of a single nucleotide or dinucleotide, up to 30% of reads associated with a barcode have an insertion or deletion in the repetitive sequence. Simulations predict that the prevalence of repetitive DNA sequences varies with the barcode design, and these predictions are quantitatively supported by the data (Figure 2B,D). Specifically, long homopolymer runs are most common in barcodes with homopolymer anchor sequences (e.g. "AA", Levy et al. 2015 design, Figure 1A), and long dinucleotide runs are most common in barcodes with repeating pairs of 2-fold degenerate bases ("WSWS…", Bhang et al. 2015 design

(Figure 1A), also used by (Eyler *et al.*, 2020; Ge *et al.*, 2020)) or repeated dinucleotide anchors (e.g. “CA”, Johnson et al. 2019 design, Figure 1A).

We have also observed that a barcode’s GC content can sometimes dramatically bias its representation in the sequencing data (Figure S1, unpublished data). This bias could be driven by GC-content dependent differences in the PCR amplification (Aird *et al.*, 2011; Benjamini & Speed, 2012; Laursen *et al.*, 2017). Furthermore, Figure S1 shows that the magnitude of this bias has a random component (i.e., the bias is stronger in some libraries than in others), which could stem from uncontrolled variation in the set-up of the PCR reaction, purity of the template, etc. These observations also suggest that GC-content driven biases can be reduced by constraining GC content of all barcodes to a narrow range. Anchors with balanced GC content (e.g., “CA” anchors as in the Johnson et al. 2019 design) can help achieve this goal (albeit at the expense of increasing the frequency of dinucleotide runs), while the “AA” and “TT” anchors used in (Levy *et al.*, 2015) lead to both low GC-content barcodes (Figure 2F) and a high occurrence of long homopolymer runs (Figure 2B). A new barcode design we propose and discuss below is an attempt to minimize each of these potential sources of bias and error (Figure 2B, D, F, black dashed lines).

Length and information

The choice of barcode length is dictated by a balance between several factors. On the one hand, barcodes cannot be too long because of current synthesis and sequencing limitations. Furthermore, longer barcodes, when read by sequencing, will contain statistically more errors than shorter barcodes. On the other hand, length of the barcode locus, together with its structure and base composition, determine the amount of information that the locus can encode, which in turn limits the number of distinct lineages that can be tracked. Specifically, the information content in bits of each barcode position is given by the logarithm with base 2 of the number of alternative nucleotides that can be present at the position. For example, each position where any one of the four nucleotides can be present encodes $\log_2 4 = 2$ bits of information, positions where only two different nucleotides are admissible encode 1 bit, whereas anchor positions encode 0 bits. The total information I of a barcode locus is given by the sum of information across all of its positions, such that there are at most 2^I distinct barcode sequences. In a lineage tracking study, each lineage must be tagged with a unique barcode, so that a barcode locus with information I enables tracking of at most 2^I distinct lineages. Thus, to track K lineages, the barcode locus must have information content that exceeds $I_{\min} = \log_2 K$ bits. A barcode locus that consists of L random nucleotides (the {N}×L design as in Ref. (Wetmore *et al.*, 2015), see Figure 1A) has the highest information content of $2L$ bits among all barcodes of length L. Thus, tracking K lineages requires the barcode of any design to be longer than $L_{\min} = ½ \log_2 K$ bp.

In practice, barcodes need to have information I that exceeds $I_{\min}$ by several bits (and, consequently, whose length exceeds $L_{\min}$ by several bp). Recent studies have successfully tracked $K = 10^5$ to $K = 10^6$ lineages ($I_{\min}$ between 16.6 and 19.9 bits and correspondingly $L_{\min}$ between 8.8 and 10 bp) with barcodes with length between 15 to 20 bp and information content between 30 and 40 bits (see Figure 1A; (Levy *et al.*, 2015; Johnson *et al.*, 2019; Eyler *et al.*, 2020; Ge *et al.*, 2020; Jasinska *et al.*, 2020; Borchert *et al.*, 2021)).

There are two reasons why I must exceed $I_{\min}$. First, since cells acquire barcoded DNA constructs at random, the barcode library must be diverse enough to ensure that the probability that two cells acquire the same barcode is small. If the frequency of the most common barcode sequence in the library is $f_{\max}$ and K cells are barcoded, then each barcode sequence is typically introduced into at most one cell whenever $Kf_{\max} \ll 1$. If all barcodes are represented in the library equally (so that their frequencies are 2^{-I}), this condition is always satisfied whenever $I > I_{\min}$. However, the distribution of barcode frequencies in the library is seldom uniform (Klein *et al.*, 2020), in which case $f_{\max} > 2^{-I}$, so that it is advisable to choose barcodes with information content exceeding $I_{\min}$ by at least a few bits to account for random sampling.

The second reason to increase I further is that barcode sequences cannot be synthesized or read with perfect accuracy. While errors are inevitable, good barcode designs account for error statistics and enable researchers to correct at least some of them. Sequencing errors can be accounted for most easily. On the Illumina platform, the error rate is estimated to be ≤ 0.4% per sequenced nucleotide (Stoler & Nekrutenko, 2021), such that up to 7.7% of reads of a 20 bp (40 bit) barcode are expected to contain at least one error and up to 0.3% are expected to contain two or more errors. Good barcode designs ensure that the true barcode sequence can be correctly inferred despite these errors. All error correction methods rely on the premise that true barcode sequences are sufficiently sparse in the sequence space, so that they all differ from each other at least at 2, or, better yet, at 4 positions (4-8 bits, see Section 4).

To evaluate the error-correction capacity of a given barcode design when tracking K lineages, it is useful to calculate the fraction of K random barcodes that have a nearest neighbor barcode within Hamming distance d. Our simulations of binary barcodes (see Methods) show that this fraction increases rapidly with K (Figure 3), such that if binary barcodes of length 30 are used to track $K = 10^5$ lineages, about 89.5% of them have another barcode at Hamming distance 4 or less, which can complicate or compromise our ability to correct many sequencing errors. However, increasing barcode length to 60 enables one to track $K \sim 10^7$ lineages while maintaining the capacity to correct sequencing errors since only about 0.04% of barcodes have a nearest neighbor within Hamming distance 6 (Figure 3).

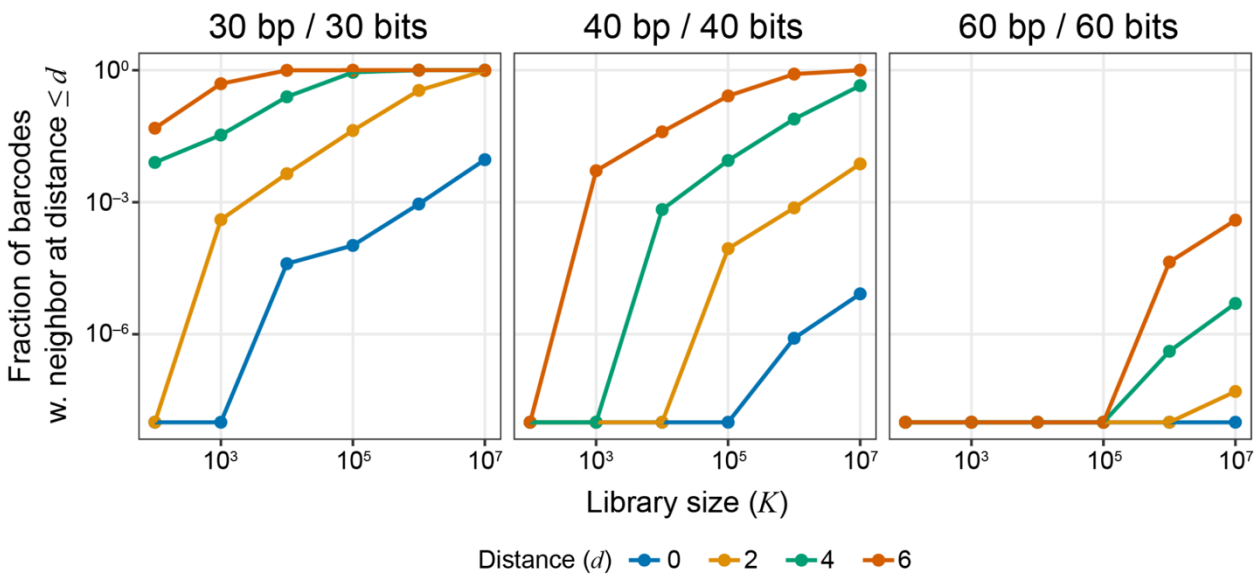

Figure 3. Fraction of binary barcodes with at least one other barcode within a certain Hamming-distance radius, as a function of library size. Lines correspond to different radii d, as shown in the legend. Panels show barcodes with different lengths and information content. For each size K, five replicate libraries of binary barcode sequences were simulated and the resulting fractions were averaged over the replicates.

Optimal barcode sequence

The considerations discussed above place conflicting demands on barcode design. High information content is most easily achieved by using fully random nucleotides, but such barcodes have problems with GC content and homopolymer runs (Figure 2). At the same time, full control of the GC content is achieved at a great reduction of information or expansion of length (see Figure 1A) and can still have problems with dinucleotide runs (Figure 2) . Thus, we propose a new barcode design that is optimal in the sense that it achieves a reasonable balance between all these demands. We propose interspersing 2-fold degenerate “WS” nucleotides between every three 4-fold degenerate nucleotides to generate a 38 bp barcode: “NNNWSNNNWSNNNWSNNNWSNNNWSNNNWSNNN”. This sequence has 62 bits of information, a guaranteed GC content between 18% and 72%, and maximum homopolymer/dinucleotide run lengths of 4 (Figure 2B, D, F, black dashed lines).

Pre-multiplexing

It is often desirable to sequence barcodes from multiple BLT experiments on one Illumina lane. The standard solution to this problem is to use Illumina indices during library preparation (Figure 1B and Section 3). However, this approach requires that the sequencing library is prepared for every sample individually. It is possible to reduce this labor and material costs by “pre-multiplexing” different BLT experiments.

One pre-multiplexing strategy is to add a short sequence—referred to as the “experiment tag”—next to the barcode (Figure 1A) and to construct barcoded strain

libraries for different BLT experiments with different experiment tags (Boyer *et al.*, 2021). Another strategy is to create multiple plasmid libraries (see Section 2.2) with non-overlapping sets of barcode sequences (Johnson *et al.*, 2019). Of course, these plasmid libraries must be sequenced to determine which barcodes belong to each set. The second strategy can be implemented easily only if the number of tracked lineages is much smaller than the diversity of the library of chemically synthesized barcode oligos.

With either strategy, pre-multiplexed samples can be pooled together prior to DNA extraction and library preparation. The identity of the BLT experiment can then be inferred from the sequence of the "experiment tag" (first strategy) or the barcode itself (second strategy). In addition to or instead of increasing throughput, pre-multiplexing can be used redundantly with standard Illumina multiplexing to avoid potential misidentification of reads due to template switching, index hopping, or primer cross-contamination (see Section 3 and Johnson *et al.*, 2019).

2.2 Synthesis and integration

Once the barcode construct has been designed, the oligonucleotides carrying the barcodes must be synthesized and engineered into the organism. While an in-depth discussion of various engineering methods involved in the barcoding process is beyond the scope of this paper, we outline here the basic steps and then discuss some considerations related to barcode construct synthesis and to the choice of the locus into which barcodes are integrated.

Overview of the barcoding process

The barcoding process usually begins with the synthesis of oligonucleotides carrying the barcode sequences. Such an oligo library is then typically used to generate a library of larger DNA constructs that are ready to be transformed into the organism of interest. These constructs are typically integrated into a plasmid backbone and transformed into *Escherichia coli* for long-term storage. Before each application, plasmids are harvested and transformed into the target organism, either directly (Levy *et al.*, 2015) or after another manipulation step, such as backbone digestion (Jasinska *et al.*, 2020) or lentivirus generation (McKenna *et al.*, 2016). Sometimes, barcodes are integrated into the organism's genome using high-efficiency recombinase systems, such as transposon-based systems like Tn7(Jasinska *et al.*, 2020), Cre-Lox (Levy *et al.*, 2015), or CRISPR-Cas9 (Zhu *et al.*, 2019).

It is important to note that the construction of barcoded strain libraries involves multiple sampling steps, each of which inherently reduces barcode diversity. It is critical to ensure that sample sizes at each step are large enough that the diversity of the barcoded strains at the end is sufficient for the purposes of the BLT experiment. It may

also be useful to sequence the plasmid library before using it for the transformation of the target organism.

Synthesis

In vitro barcodes are typically generated using chemical oligonucleotide synthesis, which can result in errors in the length of the barcode as well as its sequence. Filges et al quantified the error rate of synthesized oligonucleotides from multiple manufacturers and various purification methods, and found that IDT Ultramer and Eurofins PAGE oligonucleotides had similarly high purity (~98.4% full-length molecules; (Filges *et al.*, 2021). Oligonucleotides without any purification ("de-salted") can result in as low as 86% full-length molecules, and should thus be avoided (Filges *et al.*, 2021). In our experience with IDT, ordering "custom/hand mixed" random nucleotides provided a more even frequency distribution than "machine mixed" nucleotides (see https://www.idtdna.com/pages/products/custom-dna-rna/mixed-bases).

Integration locus

In some BLT studies, barcodes are integrated into different, sometimes random, genomic locations in different lineages (Giaever *et al.*, 2002; Wetmore *et al.*, 2015; Johnson *et al.*, 2019). But in many others, researchers wish to integrate a barcode into one specific locus, in which case they need to decide what this locus would be. The first decision is whether the barcode will be maintained on the chromosome (Levy *et al.*, 2015; Jasinska *et al.*, 2020) or on an extrachromosomal plasmid (Cira *et al.*, 2018). While the latter strategy is easier to implement, barcodes maintained on plasmids are less stable (i.e., they can be lost), although stability depends on the organism, growth environment and the type of plasmid (Friehs, 2004; Shao *et al.*, 2021).

The second question is to identify the specific locus for barcode integration. Some considerations that will bear on this decision are study-specific, e.g., whether the barcode needs to be expressed (Wagner *et al.*, 2018). Others are more general, such as the aforementioned stability requirement, i.e., the requirement that lineages maintain their barcodes over the course of the experiment. For this purpose, one should avoid barcode integration into recombination hot-spots or into loci adjacent to mobile genetic elements. Barcode stability can be further enhanced by integrating the barcode in the immediate proximity of an essential gene, such as next to an antibiotic resistance marker (Giaever *et al.*, 2002) or in an intron of an essential gene (Levy *et al.*, 2015).

Another general consideration is that the presence of the barcode should minimally perturb cellular function. For example, in many evolutionary studies, barcodes should ideally have no effect on the organism's fitness, in which case pseudogenes or genes whose disruption is known to have no effect on fitness in the study environment are good candidates for integration.

3 Barcode sequencing

Once a lineage tracking experiment is complete and samples are collected, the next step is to characterize lineage diversity in these samples by sequencing them at the barcode locus. Since the number of barcodes per sample is often very large and their relative abundances can vary by multiple orders of magnitude, sequencing must be done to a substantial depth, often $\gtrsim 10^6$ reads per sample. Our discussion here focuses on the Illumina platform where such depths can currently be achieved at a relatively low cost.

Barcode amplification and sequencing begins with DNA extraction, usually with standard organism-specific methods. Then, PCR is used to simultaneously amplify the barcode locus and attach Illumina adapters necessary to create sequencing-ready DNA fragments. Both the sequencing-library preparation and the sequencing process itself introduce errors into the barcode sequence, which creates difficulties in identifying barcodes in the data and increases noise in the estimates of their frequencies. However, clever PCR designs can help reduce and correct some of these errors, as well as reduce labor and sequencing costs. In particular, we discuss the benefits and pitfalls of using one- versus two-step PCR setups, Unique Molecular Identifiers (UMIs), inline indices and a few other factors (see Figure 1B).

One- and two-step PCR setups

The simplest way to generate sequencing-ready barcode amplicons from a sample's genomic DNA is to PCR-amplify the barcode locus using primers that contain standard Illumina adapter components, including Illumina multiplexing indices, the sequencing priming site, etc. We refer to this simplest approach as the "one-step" PCR setup (Figure 1B). A slightly more complex alternative is the "two-step" PCR setup (Figure 1B). Here, the first PCR is typically carried out for a small number of cycles (2–10). Its purpose is to attach "overhangs" to template molecules. These overhangs contain useful components, such as inline indices, UMIs and read offsets, which we discuss in detail below, as well as a "universal" priming site for the standard Illumina primers used in the second PCR. The second PCR is typically carried out for a larger number of cycles (12–25) and results in sequencing-ready fragments.

Both setups have some advantages and disadvantages. A major advantage of the two-step PCR setup is that inline indices can greatly expand multiplexing capacity, which not only increases throughput but can also improve data quality (see below). This advantage is traded off against an additional bottleneck in the two-step PCR setup because a fraction of the original template molecules do not receive overhangs (which are necessary for in the second PCR) and a fraction of molecules with overhangs are lost during the cleanup after the first PCR. The advantage of the one-step setup is that it avoids this bottleneck, potentially reducing noise, and in general involves a bit less

hands-on work. On the other hand, one-step setup requires (somewhat expensive) long non-standard primers and, most importantly, lacks the multiplexing capacity endowed by inline indices.

Inline indices

A major advantage of a two-step PCR setup is that the inline indices added during the first PCR step greatly expand the multiplexing capacity enabled by standard Illumina indices (Figure 1B). Like the Illumina indices, inline indices are predefined sequences that encode sample information. For example, each replicate of a BLT experiment can be tagged with its own inline index during the first PCR step. In this setup, sample information can be encoded by a combination of four indices (two Illumina and two inline). In principle, samples tagged with different inline indices during the first PCR can be pooled together for the second PCR, although we do not recommend this practice due to the possibility of template switching events (Kinsler *et al.*, 2022) .

Expanded multiplexing capacity allows for redundant sample encoding whereby all samples are distinguished from each other by at least two indices, e.g., one inline index and one Illumina index. One redundant design that we found particularly useful is where each 5’ inline index is associated with a unique 3’ Illumina index and each 3’ inline index is associated with a unique 5’ Illumina index. Such redundancy can be used to effectively detect primer cross-contamination, “index hopping”, and template switching events that can occur during library preparation or on the Illumina flow cell (Illumina, 2017; Guenay-Greunke *et al.*, 2021; Kinsler *et al.*, 2022). These processes generate chimeric sequences, which introduce demultiplexing errors that in turn translate into errors in lineage frequency estimates. In the aforementioned design, most such events (those that occur in the bulk of the fragment, between the inline indices) generate “inadmissible” index combinations that can be easily identified and discarded. Using this approach, we found that ~5% of reads had inadmissible index combinations (Venkataram *et al.*, 2021), but others have reported rates of up to 43% (Kinsler *et al.*, 2022). Note that, while it is possible to include inline indices in the one-step PCR setup, their utility would be limited. They cannot expand the multiplexing capacity, but can help detect some index hopping events (those that occur between the Illumina index and the inline index that are on the same primer). The rate of index hopping is much higher on “patterned flow cell” Illumina machines, so we also recommend using a non-patterned flow cell machine for barcode sequencing whenever possible (Illumina, 2017; Guenay-Greunke *et al.*, 2021; Kinsler *et al.*, 2022).

Unique Molecular Identifiers (UMIs)

The process of preparing a sequencing library introduces a number of potential errors that may influence the quality of BLT data. In particular, if the number of template molecules that are being amplified by PCR is small, data will be noisy despite high read

depth. In addition, sequence-specific biases may arise during PCR (i.e., some barcodes may be amplified more efficiently than others) which can lead to systematically inaccurate frequency estimates (Thielecke *et al.*, 2017). The two-step PCR setup allows researchers to employ Unique Molecular Identifiers, or UMIs, that can help diagnose these issues. UMIs are random sequences, typically 6 to 10 bp long, present on the first-step PCR primers (Figure 1B), such that each molecule that serves as a template in the second-step PCR is tagged with one UMI. Once the final DNA fragment is sequenced, the UMI appears at the start of each read and can be used to determine whether multiple reads with the same barcode sequence derive from the same template molecule (Kivioja *et al.*, 2011).

Although many BLT studies have used UMIs, few have clearly articulated what kinds of insight can and cannot be gained from them. UMI-tagged barcode data allow us to calculate two numbers for each barcode: the total number of reads containing the barcode and the number of unique barcode-UMI combinations among these reads. By dividing the latter by the former and subtracting this ratio from 1, we can obtain the fraction of “UMI duplicates”, i.e., the fraction of redundant reads derived from the same template molecule. To understand how the fraction of UMI duplicates can help diagnose potential PCR problems, consider two extreme cases of the distribution of UMI duplicates across barcodes.

At one extreme, the fraction UMI duplicates is close to 1 for most barcodes, which means that the same barcode is associated with the same UMI on many reads. In other words, the number of sequenced fragments greatly exceeds the number of original template molecules, so that most reads derive from a small number of templates. We refer to this regime as “template-limited”. At the other extreme, the fraction UMI duplicates is close to zero for most barcodes, which indicates that UMI duplicates are rare, i.e., almost every read contains a unique barcode-UMI combination. In other words, the number of original template molecules greatly exceeds the number of sequenced fragments, so that most templates are sequenced on at most one fragment. We refer to this regime as “read-limited”.

These regimes differ in two respects. First, given the same total sequencing depth, estimates of lineage frequencies will be noisier in the template-limited regime than in the read-limited regime simply because fewer molecules are being counted. In this sense, the read-limited regime is more cost-effective. Second, in the read-limited regime, UMIs provide little information about sequence-specific amplification biases because all templates that are represented in the sequencing data are represented equally (once) and it is unknown which templates are not represented. In contrast, sequence-specific amplification biases (if they exist) can be in principle detected in the template-limited regime because different template molecules may be represented by different numbers of reads. Such biases can also be to some extent corrected by removing UMI

duplicates, i.e., by counting unique barcode-UMI combinations rather than counting all reads carrying each barcode. However, the extent to which such biases can be corrected strongly depends on the fraction of UMI duplicates in the data. In fact, our simulations show that the power to correct biases grows slowly with the fraction of UMI duplicates (Figure S2). For example, if each template molecule is sequenced on average twice, UMI duplicates comprise 50% of reads, but discarding all them corrects only 40-70% of the underlying PCR biases.

Even if the biases cannot be corrected fully, removing UMI duplicates will in principle improve the estimation of lineage frequencies, in any sequencing regime. However, before removing UMI duplicates, researchers must ensure that the same UMI sequence is unlikely to associate with two distinct template molecules carrying the same barcode just by chance. This undesired event can happen if the UMI diversity is low. For example, if the UMI is only 6 bp long, there are only $4^6 \approx 10^3$ distinct UMIs available during the first PCR. If 10^4 distinct template molecules with a certain barcode are eventually sequenced, each UMI will on average associate with 10 different templates. Removing UMI duplicates in this case would erroneously reduce the abundance of this barcode by a factor of 10. Thus, we recommend removing UMI duplicates only if the number of possible UMI sequences is several orders of magnitude larger than the highest barcode read count.

In summary, the distribution of UMI duplicates can help us determine the sequencing regime. Sequencing in the read-limited regime will produce data that may contain unobserved PCR biases which can distort barcode frequencies. Sequencing in the template-limited regime will produce noisy data that will still contain biases, unless most of the reads are discarded. Thus, the read-limited regime is preferable in practice because of its cost-effectiveness, and most BLT studies have been done in this regime (Levy *et al.*, 2015; Johnson *et al.*, 2019). It appears more prudent to reduce sequence-specific amplification biases with careful barcode design (see Section 2.1). Thus, in our opinion, if a two-step PCR is required for multiplexing or other practical reasons, it is easy and beneficial to have UMIs on the first-step primers, but we see no fundamental issues with single-step PCR setups without UMIs.

Read offsets

Every sequencing-ready fragment contains a priming site for an Illumina sequencing primer. Although it is possible to design the barcode locus so that sequencing begins directly at the barcode (Eyler *et al.*, 2020; Ge *et al.*, 2020; Jasinska *et al.*, 2020), the standard location of the sequencing primer site is downstream of the Illumina index and upstream of the inline index/UMI region (two-step PCR in Figure 1B). This location implies that sequencing commences in a region that could have low nucleotide diversity in the sequencing library. Low diversity, particularly at the beginning of a read, can substantially reduce base-call accuracy on the Illumina platform (Illumina, 2022). This

problem is usually remedied with standard methods, such as spike-in of PhiX or by sequencing a barcode library together with a genomic library on the same lane. A barcode PCR design feature referred to as “Read offsets” can be used in conjunction with these methods to further increase nucleotide diversity at the beginning of barcode reads. The idea is simply to design a set of first-step PCR primers with inline indices or UMIs of variable length, which create “read offsets” in the downstream regions of otherwise low diversity (e.g., between the inline index and the barcode). Then, fragments with different offsets are read by the sequencer asynchronously, which increases base diversity.

Other ways to minimize errors and bias

In our experience, the quality of barcode sequencing data can vary depending on several factors, such as the type of polymerase, the PCR purification and size-selection method. We found that high-fidelity polymerases, especially during the first PCR step, consistently produce better quality data. We also found that bead-based size selection coupled with standard gel extraction works reliably better than strict E-gel-based (Thermo Fisher) size selection. While these simple general practices improve data quality, some biases remain and require more sophisticated approaches, such as those discussed above (see Section 2.1).

4 Identifying barcodes in sequencing data

Once the sequencing data has been obtained and de-multiplexed, the final technical step is to extract barcodes from sequencing reads and estimate the relative abundances of the lineages.

Barcode extraction

Extracting barcodes from the sequencing reads may appear as a trivial problem at first glance, given that the structure of the read is known by design. However, the challenge is that not all reads may have identical structure due to different read offsets (see Section 3.4), variability in barcode length that arose during synthesis, and errors that arose during sequencing library preparation and sequencing itself. These challenges can be solved using either regular expressions (“regex”, e.g. (Johnson *et al.*, 2019; Chochinov & Nguyen Ba, 2022); or sequence alignment (e.g. (Venkataram *et al.*, 2021)). The former scans each read for certain user-specified patterns of characters, whereas the latter uses sequence alignments to find the locations of constant regions (sequence regions shared by all fragments) flanking the barcode before extracting the barcode sequence between those regions.

We applied both of these approaches to six barcode sequencing datasets (Table S1) to test their speed and relative accuracy. To compare the two methods, we looked at the

first 100,000 reads of each dataset and directly compared extracted barcodes. We found that both methods successfully extracted barcodes from 94-98% of reads, with the vast majority of the remaining reads excluded due to low quality scores (Table S2). Excluding reads in which both methods did not extract a barcode (again usually based on low quality scores), the two methods extracted the same barcode in 97.5-99.5% of reads (Table S2). The most common exceptions to this overarching concordance are cases where barcodes have abnormal length. Such barcodes were correctly extracted by the alignment method but were not extracted or extracted incorrectly by our regex method, which only allows barcodes to vary in length by at most 2 base pairs. However, more lenient regular expressions can be developed to allow for more barcode length variation. Indeed, we used regular expressions with no length constraints to examine the distributions of barcode length in our datasets, which show that abnormally short barcodes exist at appreciable frequencies (Figure 2E). Finally, in very rare cases, both methods extracted incorrect barcode sequences, which happened usually due to misidentification of the constant regions flanking the barcodes.

In our hands, the regex approach ran 5 to 10 times faster than alignment, processing ~140 million reads in ~2 hours using a basic cloud machine from Deepnote. Given the speed of the regex approach, we believe it will be the method of choice for most applications despite a minor loss of accuracy. When using any method, researchers should pay attention to the fraction of reads without an extracted barcode. This fraction exceeding a few percent indicates a potential problem with sequencing quality, misspecification of parameters of the extraction method, or data (e.g., high abundance of abnormal barcodes).

Error correction

Even with the best practices suggested above, there will be a fraction of cases when the extracted barcode sequence differs from the sequence of its template molecule. The naive approach is to simply ignore these errors. However, it would come with a substantial data waste (and hence, reduced accuracy of lineage frequency estimates). Assuming a per-base error rate of 0.4% (Stoler & Nekrutenko, 2021), 7.7% of sequenced barcodes of length 20-bp contain at least one sequencing error; this fraction is 11% for 30-bp barcodes and 15% for 40-bp barcodes. Moreover, some errors may be sequence-specific (see Section 2.1), such that the naive approach may produce biased lineage frequency estimates. Fortunately, a number of error-correction techniques are available (e.g., (Li & Godzik, 2006; Edgar, 2010, 2016; Ghodsi *et al.*, 2011; James *et al.*, 2018; Wei *et al.*, 2021; Dasari & Bhukya, 2022; Millán Arias *et al.*, 2022)), some of which were developed specifically for barcode data (e.g., (Zorita *et al.*, 2015; Zhao *et al.*, 2018; Tavakolian *et al.*, 2022)).

All these methods rely on a few assumptions. True barcodes must be sufficiently sparse in the sequence space, errors must be relatively infrequent, and an erroneous barcode

sequence must be more similar to its “parent” barcode than to any other true barcode. With good barcode design and careful sequencing library preparation, these assumptions are usually met. Then, error correction can be achieved by clustering sequenced barcodes according to a sensible similarity metric, such as Hamming or Levenshtein distance. The primary challenge is computational: BLT data often contains tens or hundreds of millions of reads, and calculating pairwise distances between all of them is not feasible. Clever algorithms that limit the number of comparisons are thus key to computational efficiency.

We selected six error-correction software, two developed for generic sequence data, DNAClust (Ghodsi *et al.*, 2011) and CD-Hit (Li & Godzik, 2006), and four developed specifically for barcode data, Bartender (Zhao *et al.*, 2018), Starcode (Zorita *et al.*, 2015), Shepherd (Tavakolian *et al.*, 2022) and “Deletion-Correct”, a modified version of the algorithm used in Johnson *et al.* (2019). We first tested their accuracy by performing error correction on a dataset of simulated barcode reads with realistic errors (Methods). We found that all four barcode-specific methods successfully identified the vast majority of barcode sequences and correctly inferred lineage abundances (Pearson R = 1.0, Figure 4A-D), while both generic methods performed poorly (Figure 4E,F). While all four bespoke methods perform very well, they each had some idiosyncrasies. Bartender has a substantially higher false positive rate than either Shepherd or Starocode, where error sequences are incorrectly classified as distinct barcodes from the true sequence. Furthermore, Bartender incorrectly assigned an error sequence as the true barcode sequence 2.7% of the time and Starcode exhibited incorrect assignment 0.33% of the time, in comparison to 0.06% for Shepherd and 0% for Deletion-Correct. However, more than 95% of erroneous sequences inferred by each barcode-specific method were different from the correct sequences by a single basepair. Meanwhile, while Deletion-Correct did not misidentify any sequences, it failed to detect many barcodes with < 5 reads.

We next applied the barcode-specific methods on three empirical datasets after having extracted barcodes using the alignment-based method (Levy *et al.*, 2015; Johnson *et al.*, 2019; Borchert *et al.*, 2021). We found that Shepherd failed to identify many putative barcodes in these empirical datasets (Table S3). Specifically, the Levy et al, Johnson et al, and Borchert et al datasets contain 21,000, 10,000 and 2,800 barcodes with at least 10 reads each, respectively, that are found by Bartender, Starcode and Deletion-Correct but not by Shepherd. All lineages missed by Shepherd but identified by other methods have abnormal length, suggesting that Shepherd’s filtering criteria are too strict (it filters out barcodes whose length deviates from the expected by more than 1 bp). While Starcode consistently ran faster than the other methods, we note that each method took < 4 minutes to run on a personal desktop computer (AMD Ryzen 5 1600, 16GB RAM), with the exception of Shepherd on the Levy et al dataset, which took about 30 minutes.

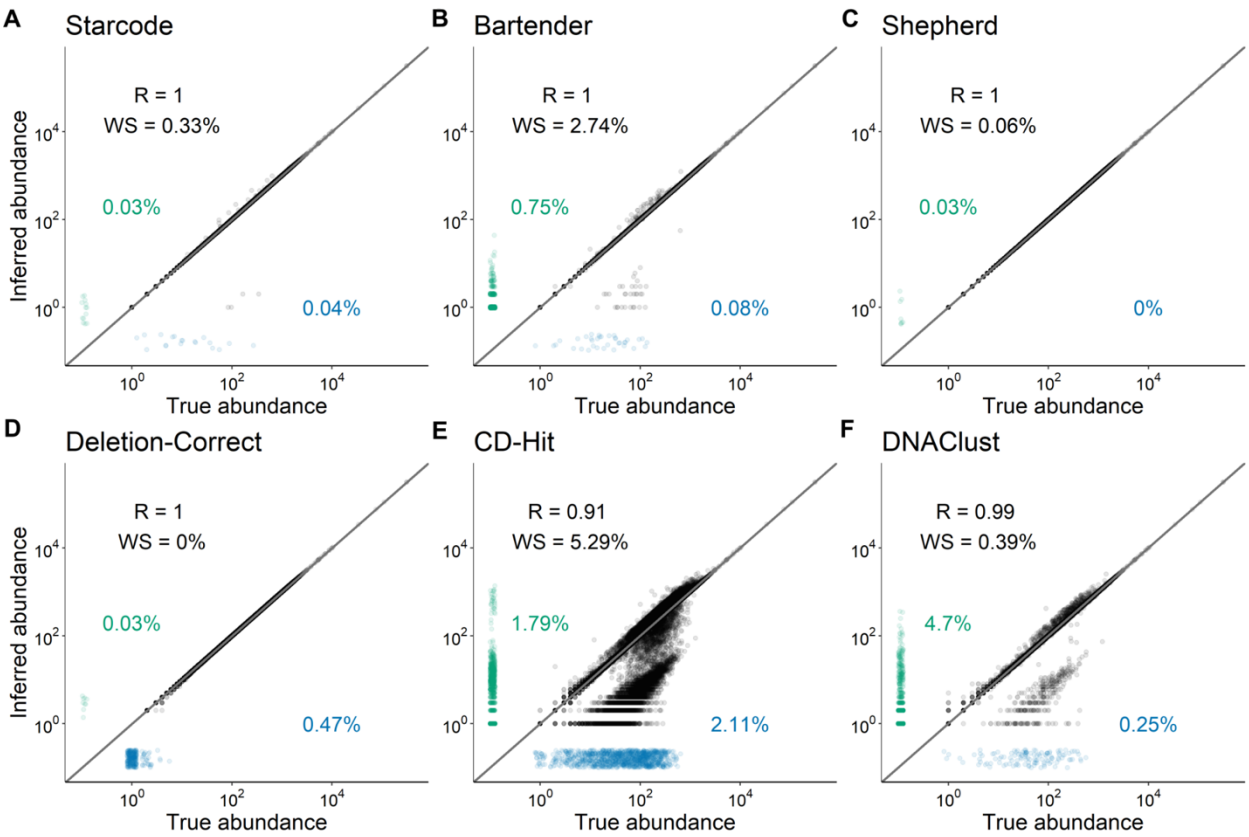

Figure 4. Comparison of error correction methods. We tested six published error correction methods on a simulated barcode dataset (see Methods for details). The true abundance of each barcode (x-axis) is shown against the inferred abundance of the barcode most closely associated with it after error correction (y-axis). "R" is the Pearson correlation coefficient of log-transformed data for the successfully inferred barcodes. "WS" is the fraction of barcodes where a wrong sequence was inferred by the error correction method. Blue points along the x-axis show true barcodes that were not identified (numbers show percentages). Green points along the y-axis show identified barcodes that are not associated with a true barcode (numbers show percentages). The gray line is the diagonal y = x.

For all practical purposes, these execution times are sufficiently short to not substantially influence the choice of method.

In summary, we strongly recommend using barcode-specific methods for error correction, including Shepherd, Starcode, Bartender and Deletion-Correct. It may be useful to use multiple methods in conjunction to better account for false positives, false negatives, incorrect barcode sequence assignment, and barcodes of abnormal length.

6 Summary

We have reviewed the choices faced by researchers during the design, sequencing and identification of random barcodes, as well as some of the implications of these choices for the quality of the data. Here we provide a succinct summary of our main points.

Design, synthesis and integration

- The base composition of the barcode sequence strongly affects the error rates during sequencing library preparation and/or sequencing process itself. In particular, long homopolymer or dinucleotide runs and extremely high or low GC content should be avoided.
- Barcode length and base composition limit the number of lineages that can be tracked. For barcodes with length 20 to 40 bp, the library size should be small enough that all but a small fraction of barcodes are at Hamming distance of at least four from each other.
- Barcode oligonucleotides synthesized with HPLC or PAGE purification and hand-mixed random bases result in barcode sequences with lower error rates.
- When choosing the integration locus, consider (i) its stability with respect to recombination events that can lead to barcode loss and (ii) the implications of genetic manipulations at the locus for the organism's physiology.

Sequencing

- Inline indices greatly expand multiplexing capacity and allow for detection of errors that arise due to template switching, index hopping and primer cross-contamination.
- UMIs help detect whether noise in the data comes from a low number of template molecules, but their power to correct PCR biases is low.
- Read offsets help improve sequencing quality.
- Use of high-fidelity polymerase during PCR reduces amplification errors.

Identification

- Regex and alignment approaches are both excellent at barcode extraction. Regex is faster, alignment is slightly better at identifying abnormal barcode sequences.
- Error correction methods designed specifically for barcode data work much better than generic methods. Among the former, Shepherd is most accurate on simulated data but fails to recover barcodes of abnormal length, which appear in real data at non-negligible frequencies.

7 Methods

7.1 Measuring variation in barcode length

To measure variation in barcode length in the empirical datasets, we extracted barcodes using regular expressions that strictly match the 10 base pairs before and/or after the barcode sequence, with no length criteria for the sequence in between. We

then measured the percentage of barcodes with each possible length, ranging from 10 bp less than expected to 10 bp more than expected. We only considered barcodes with at least 20 read counts for this analysis to minimize the impact of amplification and sequencing errors on the distributions. We show this data in Figure 2E.

7.2 Estimation of errors in barcodes with repetitive sequences

We estimated the frequency of errors in repetitive barcode sequences using the barcode sequences and associated counts extracted using the alignment method. For both single nucleotides and every combination of two nucleotides ("dinucleotide"), we looked for barcodes with *N* repeats of that nucleotide or dinucleotide, with *N* ranging from 3 to 13. For the top 50 most abundant barcodes with a particular length run (excluding barcodes with less than or equal to 100 reads), we searched for putative error barcodes in which the number of repeats was increased or decreased by 1 or 2, but the rest of the barcode was identical. In parallel, we searched for single nucleotide errors derived from each of these barcodes. We added the read counts from both the indel and single-nucleotide errors to each "true" barcode's read counts in order to ensure an accurate denominator when calculating error rates. We report the total indel error rate in Figure 2, which we calculate as the combined frequency of all four types of errors (insertions and deletions of 1 or two repeats).

7.3 Simulating barcode designs and measuring barcode statistics

In order to assess the features of various barcode designs, we simulated 100,000 random barcodes for 5 possible designs, 4 associated with existing designs in our empirical datasets, and one new design ("N3WS"). We then measured the statistics of these sets of barcodes, along with the sets of empirical barcodes. For each empirical dataset, we used the list of barcodes derived from alignment-based extraction, excluding any barcodes that are not the expected length. For each barcode, we measured the percentage of GC bases, the longest homopolymer run, and the longest dinucleotide run (Figure 2).

7.4 Distribution of Hamming distances between barcodes

We generated barcode libraries with 10^2, 10^3, 10^4, 10^5, 10^6 and 10^7 binary barcodes of length 30, 40 or 60. To reduce computation time, we utilized an approximate-nearest-neighbor algorithm as provided by the python Annoy library to find the nearest neighbor for every sequence in the dataset, which requires binary input. We report the fraction of sequences with a Hamming distance to their nearest neighbor less than or equal to 0, 2, 4 or 6, averaged over five replicate simulations for each parameter combination.

7.5 Identification of UMI duplicates and detection of chimeric reads

We previously used BarcodeCounter2 to extract barcodes from lineage tracking data (Venkataram *et al.*, 2021). This software uses inline and Illumina index information to identify chimeric reads during sample demultiplexing and provides a count of UMI duplicates found for each barcode within each sequenced sample. We report chimeric read and UMI duplicate rates for the lineage tracking data from (Venkataram *et al.*, 2021).

7.6 Simulations of bias detection using UMIs

We simulated the process of template sampling in order to assess the utility of removing UMI duplicates in correcting biases in barcode frequency that occur during library preparation. We simulated cases in which a focal barcode is at a true frequency of 0.05 or 0.25 in the template molecules, the total number of reads is 1 million, and the number of template molecules tagged with UMIs varies between 100,000 and 10 million. This variation in the number of template molecules represents a shift between the two regimes discussed in the main text: the lower the number of template molecules, the more commonly UMI sequences will repeat. For each frequency and number of template molecules, we simulate a range of biases. In each case, we randomly sample 1 million reads from a “post-library-preparation pool” in which the initial abundance of the focal barcode has been multiplied by the bias factor. We also draw UMIs for each of these reads from a pool of unique UMIs corresponding to the (unbiased) number of template molecules associated with the focal barcode and the remainder of the population, respectively. We assume that every template molecule has a unique UMI (note that this may not be the case in real datasets, depending on UMI length). Using the number of unique UMIs in the simulated reads associated with the focal barcode and the remainder of the population, we calculate the frequency of the focal barcode after UMI deduplication, shown in Figure S2.

7.7 Comparison of barcode extraction methods

We implemented custom regular expression and alignment software to extract barcodes from each of six barcode datasets. To extract barcodes by regular expressions, a set of five custom regular expressions were composed for each dataset to extract barcode sequences based on the read sequences from each dataset. To extract barcodes by alignment, we used BLASTn+ v 2.6.0 (Altschul *et al.*, 1990; Camacho *et al.*, 2009) to identify the location of the constant sequences flanking each barcode within the read, and used these positions to extract the barcode sequence. BLASTn+ was run with the parameters ‘-word_size 6 -outfmt 6 -evalue 1E0 -maxhsps 1’. The abundance of each unique extracted sequence was tabulated for downstream analysis.

7.8 Comparison of error correction methods

Simulations of barcode data with errors

To simulate barcode data with a range of frequencies including high frequency outliers, we first drew 99,895 barcode abundances from an exponential distribution with mean 1, 100 barcode abundances from an exponential distribution with mean 10, and 5 barcode abundances from an exponential distribution with a mean of 1000. We assigned each abundance to a randomly generated 20 bp barcode ("N20"). We then drew a number of reads associated with each barcode from a poisson distribution with a mean of the frequency of the barcode multiplied by 25 million (such that we expect a total of approximately 25 million reads). For any barcode with a mononucleotide run of 5 or more base pairs, we first simulated indel errors, using our empirical data on the rates of these events (Figure 2) to draw a poisson-distributed number of reads with a single base insertion or deletion. This indel simulation process is carried out recursively such that multiple-base indels are possible. Next, we simulated single nucleotide errors for each individual read at a rate of 0.4% per base. The final simulated dataset consists of a single row for each unique barcode that was "read" in this process, associated with a number of reads and the "true" barcode from which it is derived.

Comparison of error correction methods

We tested six error correction methods (Bartender v1.1.0, DNAClust v3, Starcode v1.4, Shepherd downloaded Aug 15 2022, CD-Hit v4.8.1 and Deletion-Correct, provided in this manuscript) on each of four datasets (Levy et al, Borchert et al, Johnson et al and the simulated dataset). Each program was run with the following parameters, where *L* is the length of the barcode, including anchor sequences:

Bartender '-d 3'

DNAClust '-s {1-3.1/L} -k 6'

Starcode '-d 3 -s'

CD-Hit '-c {1-3.1/L} -n 6'

Shepherd '-l L -bft 4 -eps 3'

Deletion-Correct: min_counts_for_centroid=2, max_edits=3, poisson_error_rate=0.1

Programs were run on a personal desktop computer with an AMD Ryzen5 1600 3.2GHz processor and 16GB of ram. Software with multithreading support was run with 10 threads / allocated processing cores and 5000MB of allocated memory.

Data availability

All code used for simulations, analysis and generating figures have been deposited on Zenodo at https://doi.org/10.5281/zenodo.7052125.

Acknowledgements

We thank Alex Nguyen Ba and Morgan Price for helpful discussions. MSJ is supported by the NSF Postdoctoral Research Fellowships in Biology Program under Grant No. 2109800. SK acknowledges support by the NIH (Grant 1R01GM137112).

Appendix. Genotyping clones at a barcode locus

A common task when using barcoded strain libraries is to identify the barcodes for individual clones isolated from the library. The traditional approach, based on Sanger sequencing, is effective for a small number of clones, but it becomes prohibitively expensive and labor intensive at ~10^2 clones. At larger scales, approaches that leverage next-generation sequencing technologies are preferred.

The most straightforward cheaper alternative to Sanger sequencing is to individually amplify the barcode of each clone, tag it with a unique combination of indices and sequence it on the Illumina platform. Since this approach involves the same number of DNA extractions and PCR reactions as the Sanger approach, the cost of this approach scales linearly with the number of samples. The savings come from the reduction of sequencing costs per sample: sequencing of a sample with the Sanger technology currently costs about 2 USD, while the cost is less than 0.02 USD per sample on the Illumina MiSeq platform when sequencing 10,000 clones.

An even cheaper alternative for genotyping many clones is a pooled sequencing strategy sometimes referred to as “Cartesian pooling”, “Compressed sensing” or the “Sudoku method” (Barillot *et al.*, 1991; Erlich *et al.*, 2009; Shental *et al.*, 2010; Vandewalle *et al.*, 2015; Baym *et al.*, 2016). The idea is to pool clones into multiple groups, such that each clone is present in several groups, prepare one Illumina library per group, sequence them and then infer the genotypes of all clones based on the knowledge of their presence/absence in each group. For example, clones can be arrayed into a 3-dimensional grid of p plates, each with r rows and c columns, e.g., in a series of 96-well plates. This would result in $p + r + c$ groups, each containing all clones in a given plate, row or column across the entire collection. In this arrangement, each clone is present in only one specific combination of plate, row and column groups, and no two clones are present in the same combination of groups. In other words, group combination serves as a clone’s unique fingerprint. Further, if all clones have distinct barcodes, there will be only one barcode sequence present in any given combination of plate, row and column groups. In other words, each sequence will have a unique fingerprint, through which it can be assigned to the correct clone. While this strategy requires some additional work pooling clones into groups, the overall cost scales approximately as $K^{1/3}$, where K is the number of clones, since only about $K^{1/3}$ DNA extractions and PCR reactions are required. For example, a library of 960 clones can be

763 characterized using 30 pools (10 plate pools, 8 row pools and 12 column pools). The
764 efficiency can be further improved by using additional “dimensions” for pooling and
765 ensuring that all groups have similar numbers of clones (Barillot *et al.*, 1991).

766 A key limitation of the Cartesian pooling approach occurs when multiple clones have the
767 same barcode. In this case, some sequences are present in more than one group
768 combination (i.e., they have multiple fingerprints) which makes the association of
769 sequences with clones non-unique. For example, consider a collection of 96 clones,
770 pooled by row and column, where clones present in wells A5 and D7 have the same
771 barcode. In this scenario, row groups A and D as well as column groups 5 and 7 will
772 have this particular barcode sequence. Thus, the barcode could be assigned to any of
773 four wells: A5, A7, D5 and D7. Resolving these degeneracies may require additional
774 genotyping (Barillot *et al.*, 1991).

References

Aird, D., Ross, M.G., Chen, W.-S., Danielsson, M., Fennell, T., Russ, C., *et al.* 2011. Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol.* **12**: R18.

Alemany, A., Florescu, M., Baron, C.S., Peterson-Maduro, J. & van Oudenaarden, A. 2018. Whole-organism clone tracing using single-cell sequencing. *Nature* **556**: 108–112.

Al'Khafaji, A.M., Deatherage, D. & Brock, A. 2018. Control of Lineage-Specific Gene Expression by Functionalized gRNA Barcodes. *ACS Synth. Biol.* **7**: 2468–2474.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. 1990. Basic local alignment search tool. *J. Mol. Biol.* **215**: 403–410.

Barillot, E., Lacroix, B. & Cohen, D. 1991. Theoretical analysis of library screening using a N-dimensional pooling strategy. *Nucleic Acids Res.* **19**: 6241–6247.

Baron, C.S. & van Oudenaarden, A. 2019. Unravelling cellular relationships during development and regeneration using genetic lineage tracing. *Nat. Rev. Mol. Cell Biol.* **20**: 753–765.

Baym, M., Shaket, L., Anzai, I.A., Adesina, O. & Barstow, B. 2016. Rapid construction of a whole-genome transposon insertion collection for Shewanella oneidensis by Knockout Sudoku. *Nat. Commun.* **7**: 13270.

Bell, C.C., Magor, G.W., Gillinder, K.R. & Perkins, A.C. 2014. A high-throughput screening strategy for detecting CRISPR-Cas9 induced mutations using next-generation sequencing. *BMC Genomics* **15**: 1002.

Benjamini, Y. & Speed, T.P. 2012. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Res.* **40**: e72.

Bhang, H.-E.C., Ruddy, D.A., Krishnamurthy Radhakrishna, V., Caushi, J.X., Zhao, R., Hims, M.M., *et al.* 2015. Studying clonal dynamics in response to cancer therapy using high-complexity barcoding. *Nat. Med.* **21**: 440–448.

Blundell, J.R. & Levy, S.F. 2014. Beyond genome sequencing: lineage tracking with barcodes to study the dynamics of evolution, infection, and cancer. *Genomics* **104**: 417–430.

Borchert, E., Hammerschmidt, K., Hentschel, U. & Deines, P. 2021. Enhancing Microbial Pollutant Degradation by Integrating Eco-Evolutionary Principles with Environmental Biotechnology. *Trends Microbiol.*, doi: 10.1016/j.tim.2021.03.002.

Boyer, S., Hérissant, L. & Sherlock, G. 2021. Adaptation is influenced by the complexity of environmental change during evolution in a dynamic environment. *PLoS Genet.* **17**: e1009314.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., *et al.* 2009. BLAST+: architecture and applications. *BMC Bioinformatics* **10**: 421.

Chan, M.M., Smith, Z.D., Grosswendt, S., Kretzmer, H., Norman, T.M., Adamson, B., *et al.* 2019. Molecular recording of mammalian embryogenesis. *Nature* **570**: 77–82.

Chochinov, C.A. & Nguyen Ba, A.N. 2022. Bulk-Fitness Measurements Using Barcode

811 Sequencing Analysis in YeastYeasts. In: *Yeast Functional Genomics: Methods and Protocols*
812 (F. Devaux, ed), pp. 399–415. Springer US, New York, NY.

813 Cira, N.J., Pearce, M.T. & Quake, S.R. 2018. Neutral and selective dynamics in a synthetic
814 microbial community. *Proc. Natl. Acad. Sci. U. S. A.* **115**: E9842–E9848.

815 Conklin, E.G. 1905. *The Organization and Cell-lineage of the Ascidian Egg*. Academy of Natural
816 Sciences.

817 Dasari, C.M. & Bhukya, R. 2022. MapReduce paradigm: DNA sequence clustering based on
818 repeats as features. *Expert Syst.* **39**. Wiley.

819 Dujardin, P., Baginska, A.K., Urban, S. & Grüner, B.M. 2021. Unraveling Tumor Heterogeneity
820 by Using DNA Barcoding Technologies to Develop Personalized Treatment Strategies in
821 Advanced-Stage PDAC. *Cancers* **13**.

822 Edgar, R.C. 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*
823 **26**: 2460–2461.

824 Edgar, R.C. 2016. UNOISE2: improved error-correction for Illumina 16S and ITS amplicon
825 sequencing. *bioRxiv* 081257.

826 Erlich, Y., Chang, K., Gordon, A., Ronen, R., Navon, O., Rooks, M., *et al.* 2009. DNA Sudoku--
827 harnessing high-throughput sequencing for multiplexed specimen analysis. *Genome Res.* **19**:
828 1243–1253.

829 Eyler, C.E., Matsunaga, H., Hovestadt, V., Vantine, S.J., van Galen, P. & Bernstein, B.E. 2020.
830 Single-cell lineage analysis reveals genetic and epigenetic interplay in glioblastoma drug
831 resistance. *Genome Biol.* **21**: 174.

832 Fasanello, V.J., Liu, P., Botero, C.A. & Fay, J.C. 2020. High-throughput analysis of adaptation
833 using barcoded strains of Saccharomyces cerevisiae. *PeerJ* **8**: e10118.

834 Fennell, K.A., Vassiliadis, D., Lam, E.Y.N., Martelotto, L.G., Balic, J.J., Hollizeck, S., *et al.* 2022.
835 Non-genetic determinants of malignant clonal fitness at single-cell resolution. *Nature* **601**: 125–
836 131.

837 Filges, S., Mouhanna, P. & Ståhlberg, A. 2021. Digital Quantification of Chemical
838 Oligonucleotide Synthesis Errors. *Clin. Chem.* **67**: 1384–1394.

839 Frieda, K.L., Linton, J.M., Hormoz, S., Choi, J., Chow, K.-H.K., Singer, Z.S., *et al.* 2017.
840 Synthetic recording and in situ readout of lineage information in single cells. *Nature* **541**: 107–
841 111.

842 Friehs, K. 2004. Plasmid Copy Number and Plasmid Stability. In: *New Trends and*
843 *Developments in Biochemical Engineering* (T. Scheper, ed), pp. 47–82. Springer Berlin
844 Heidelberg, Berlin, Heidelberg.

845 Ge, J.Y., Shu, S., Kwon, M., Jovanović, B., Murphy, K., Gulvady, A., *et al.* 2020. Acquired
846 resistance to combined BET and CDK4/6 inhibition in triple-negative breast cancer. *Nat.*
847 *Commun.* **11**: 2350.

848 Ghodsi, M., Liu, B. & Pop, M. 2011. DNACLUST: accurate and efficient clustering of

phylogenetic marker genes. *BMC Bioinformatics* **12**: 271.

Giaever, G., Chu, A.M., Ni, L., Connelly, C., Riles, L., Véronneau, S., *et al.* 2002. Functional profiling of the Saccharomyces cerevisiae genome. *Nature* **418**: 387–391.

Guenay-Greunke, Y., Bohan, D.A., Traugott, M. & Wallinger, C. 2021. Handling of targeted amplicon sequencing data focusing on index hopping and demultiplexing using a nested metabarcoding approach in ecology. *Sci. Rep.* **11**: 19510.

Gutierrez, C., Al'Khafaji, A.M., Brenner, E., Johnson, K.E., Gohil, S.H., Lin, Z., *et al.* 2021. Multifunctional barcoding with ClonMapper enables high-resolution study of clonal dynamics during tumor evolution and treatment. *Nat Cancer* **2**: 758–772.

Holland, E.C. & Varmus, H.E. 1998. Basic fibroblast growth factor induces cell migration and proliferation after glia-specific gene transfer in mice. *Proc. Natl. Acad. Sci. U. S. A.* **95**: 1218–1223.

Hsu, Y.-C. 2015. Theory and Practice of Lineage Tracing. *Stem Cells* **33**: 3197–3204.

Illumina. 2017. Effects of Index Misassignment on Multiplexing and Downstream Analysis.

Illumina. 2022. What is nucleotide diversity and why is it important?

James, B.T., Luczak, B.B. & Girgis, H.Z. 2018. MeShClust: an intelligent tool for clustering DNA sequences. *Nucleic Acids Res.* **46**: e83.

Jasinska, W., Manhart, M., Lerner, J., Gauthier, L., Serohijos, A.W.R. & Bershtein, S. 2020. Chromosomal barcoding of E. coli populations reveals lineage diversity dynamics at high resolution. *Nat Ecol Evol* **4**: 437–452.

Johnson, M.S., Martsul, A., Kryazhimskiy, S. & Desai, M.M. 2019. Higher-fitness yeast genotypes are less robust to deleterious mutations. *Science* **366**: 490–493.

Kalhor, R., Kalhor, K., Mejia, L., Leeper, K., Graveline, A., Mali, P., *et al.* 2018. Developmental barcoding of whole mouse via homing CRISPR. *Science* **361**.

Kebschull, J.M. & Zador, A.M. 2018. Cellular barcoding: lineage tracing, screening and beyond. *Nat. Methods* **15**: 871–879.

Kinsler, Schmidlin, Newell, Eder, Apodaca, Lam, *et al.* 2022. Extreme sensitivity of fitness to environmental conditions; lessons from #1BigBatch. *bioRxiv* 2022.08.25.505320.

Kivioja, T., Vähärautio, A., Karlsson, K., Bonke, M., Enge, M., Linnarsson, S., *et al.* 2011. Counting absolute numbers of molecules using unique molecular identifiers. *Nat. Methods* **9**: 72–74.

Klein, J.C., Agarwal, V., Inoue, F., Keith, A., Martin, B., Kircher, M., *et al.* 2020. A systematic evaluation of the design and context dependencies of massively parallel reporter assays. *Nat. Methods* **17**: 1083–1091.

Kretzschmar, K. & Watt, F.M. 2012. Lineage tracing. *Cell* **148**: 33–45.

Laursen, M.F., Dalgaard, M.D. & Bahl, M.I. 2017. Genomic GC-Content Affects the Accuracy of

16S rRNA Gene Sequencing Based Microbial Profiling due to PCR Bias. *Front. Microbiol.* **8**: 1934.

Levy, S.F., Blundell, J.R., Venkataram, S., Petrov, D.A., Fisher, D.S. & Sherlock, G. 2015. Quantitative evolutionary dynamics using high-resolution lineage tracking. *Nature* **519**: 181. Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved.

Li, W. & Godzik, A. 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**: 1658–1659.

Li, X., Patena, W., Fauser, F., Jinkerson, R.E., Saroussi, S., Meyer, M.T., *et al.* 2019. A genome-wide algal mutant library and functional screen identifies genes required for eukaryotic photosynthesis. *Nat. Genet.* **51**: 627–635.

Masuyama, N., Mori, H. & Yachie, N. 2019. DNA barcodes evolve for high-resolution cell lineage tracing. *Curr. Opin. Chem. Biol.* **52**: 63–71.

McKenna, A., Findlay, G.M., Gagnon, J.A., Horwitz, M.S., Schier, A.F. & Shendure, J. 2016. Whole-organism lineage tracing by combinatorial and cumulative genome editing. *Science* **353**: aaf7907.

Millán Arias, P., Alipour, F., Hill, K.A. & Kari, L. 2022. DeLUCS: Deep learning for unsupervised clustering of DNA sequences. *PLoS One* **17**: e0261531.

Nguyen Ba, A.N., Cvijović, I., Rojas Echenique, J.I., Lawrence, K.R., Rego-Costa, A., Liu, X., *et al.* 2019. High-resolution lineage tracking reveals travelling wave of adaptation in laboratory yeast. *Nature*, doi: 10.1038/s41586-019-1749-3.

Peikon, I.D., Gizatullina, D.I. & Zador, A.M. 2014. In vivo generation of DNA sequence diversity for cellular barcoding. *Nucleic Acids Res.* **42**: e127.

Qian, J., Lu, Z.-X., Mancuso, C.P., Jhuang, H.-Y., Del Carmen Barajas-Ornelas, R., Boswell, S.A., *et al.* 2020. Barcoded microbial system for high-resolution object provenance. *Science* **368**: 1135–1140.

Raj, B., Wagner, D.E., McKenna, A., Pandey, S., Klein, A.M., Shendure, J., *et al.* 2018. Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. *Nat. Biotechnol.* **36**: 442–450.

Reizel, Y., Chapal-Ilani, N., Adar, R., Itzkovitz, S., Elbaz, J., Maruvka, Y.E., *et al.* 2011. Colon stem cell and crypt dynamics exposed by cell lineage reconstruction. *PLoS Genet.* **7**: e1002192.

Reizel, Y., Itzkovitz, S., Adar, R., Elbaz, J., Jinich, A., Chapal-Ilani, N., *et al.* 2012. Cell lineage analysis of the mammalian female germline. *PLoS Genet.* **8**: e1002477.

Roh, V., Abramowski, P., Hiou-Feige, A., Cornils, K., Rivals, J.-P., Zougman, A., *et al.* 2018. Cellular Barcoding Identifies Clonal Substitution as a Hallmark of Local Recurrence in a Surgical Model of Head and Neck Squamous Cell Carcinoma. *Cell Rep.* **25**: 2208–2222.e7.

Schubert, M.G., Goodman, D.B., Wannier, T.M., Kaur, D., Farzadfard, F., Lu, T.K., *et al.* 2021. High-throughput functional variant screens via in vivo production of single-stranded DNA. *Proc. Natl. Acad. Sci. U. S. A.* **118**.

Schumacher, T.N.M., Gerlach, C. & van Heijst, J.W.J. 2010. Mapping the life histories of T cells. *Nat. Rev. Immunol.* **10**: 621–631.

Serbedzija, G.N., Bronner-Fraser, M. & Fraser, S.E. 1989. A vital dye analysis of the timing and pathways of avian trunk neural crest cell migration. *Development* **106**: 809–816.

Shao, B., Rammohan, J., Anderson, D.A., Alperovich, N., Ross, D. & Voigt, C.A. 2021. Single-cell measurement of plasmid copy number and promoter activity. *Nat. Commun.* **12**: 1475.

Shental, N., Amir, A. & Zuk, O. 2010. Identification of rare alleles and their carriers using compressed se(que)nsing. *Nucleic Acids Res.* **38**: e179.

Smith, A.M., Heisler, L.E., Mellor, J., Kaper, F., Thompson, M.J., Chee, M., *et al.* 2009. Quantitative phenotyping via deep barcode sequencing. *Genome Res.* **19**: 1836–1842.

Spanjaard, B., Hu, B., Mitic, N., Olivares-Chauvet, P., Janjuha, S., Ninov, N., *et al.* 2018. Simultaneous lineage tracing and cell-type identification using CRISPR-Cas9-induced genetic scars. *Nat. Biotechnol.* **36**: 469–473.

Stoler, N. & Nekrutenko, A. 2021. Sequencing error profiles of Illumina sequencing instruments. *NAR Genom Bioinform* **3**: lqab019.

Tavakolian, N., Frazão, J.G., Bendixsen, D., Stelkens, R. & Li, C.-B. 2022. Shepherd: Accurate clustering for correcting DNA barcode errors. *Bioinformatics*, doi: 10.1093/bioinformatics/btac395.

Thielecke, L., Aranyossy, T., Dahl, A., Tiwari, R., Roeder, I., Geiger, H., *et al.* 2017. Limitations and challenges of genetic barcode quantification. *Sci. Rep.* **7**: 43249.

Umkehrer, C., Holstein, F., Formenti, L., Jude, J., Froussios, K., Neumann, T., *et al.* 2021. Isolating live cell clones from barcoded populations using CRISPRa-inducible reporters. *Nat. Biotechnol.* **39**: 174–178.

Vandewalle, K., Festjens, N., Plets, E., Vuylsteke, M., Saeys, Y. & Callewaert, N. 2015. Characterization of genome-wide ordered sequence-tagged Mycobacterium mutant libraries by Cartesian Pooling-Coordinate Sequencing. *Nat. Commun.* **6**: 7106.

van Heijst, J.W.J., Gerlach, C., Swart, E., Sie, D., Nunes-Alves, C., Kerkhoven, R.M., *et al.* 2009. Recruitment of antigen-specific CD8+ T cells in response to infection is markedly efficient. *Science* **325**: 1265–1269.

VanHorn, S. & Morris, S.A. 2021. Next-Generation Lineage Tracing and Fate Mapping to Interrogate Development. *Dev. Cell* **56**: 7–21.

Venkataram, S., Kuo, H.-Y., Hom, E.F.Y. & Kryazhimskiy, S. 2021. Early adaptation in a microbial community is dominated by mutualism-enhancing mutations. *bioRxiv* 2021.07.07.451547.

Wagenblast, E., Soto, M., Gutiérrez-Ángel, S., Hartl, C.A., Gable, A.L., Maceli, A.R., *et al.* 2015. A model of breast cancer heterogeneity reveals vascular mimicry as a driver of metastasis. *Nature* **520**: 358–362.

Wagner, D.E. & Klein, A.M. 2020. Lineage tracing meets single-cell omics: opportunities and

961 challenges. *Nat. Rev. Genet.* **21**: 410–427.

962 Wagner, D.E., Weinreb, C., Collins, Z.M., Briggs, J.A., Megason, S.G. & Klein, A.M. 2018.
963 Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo.
964 *Science* **360**: 981–987.

965 Weinreb, C., Rodriguez-Fraticelli, A., Camargo, F.D. & Klein, A.M. 2020. Lineage tracing on
966 transcriptional landscapes links state to fate during differentiation. *Science*, doi:
967 10.1126/science.aaw3381.

968 Wei, Z.-G., Zhang, X.-D., Cao, M., Liu, F., Qian, Y. & Zhang, S.-W. 2021. Comparison of
969 Methods for Picking the Operational Taxonomic Units From Amplicon Sequences. *Front.*
970 *Microbiol.* **12**: 644012.

971 Wetmore, K.M., Price, M.N., Waters, R.J., Lamson, J.S., He, J., Hoover, C.A., *et al.* 2015. Rapid
972 quantification of mutant fitness in diverse bacteria by sequencing randomly bar-coded
973 transposons. *MBio* **6**: e00306–15.

974 Woodworth, M.B., Girskis, K.M. & Walsh, C.A. 2017. Building a lineage from single cells:
975 genetic techniques for cell lineage tracking. *Nat. Rev. Genet.* **18**: 230–244.

976 Zhao, L., Liu, Z., Levy, S.F. & Wu, S. 2018. Bartender: a fast and accurate clustering algorithm
977 to count barcode reads. *Bioinformatics* **34**: 739–747.

978 Zhu, S., Cao, Z., Liu, Z., He, Y., Wang, Y., Yuan, P., *et al.* 2019. Guide RNAs with embedded
979 barcodes boost CRISPR-pooled screens. *Genome Biol.* **20**: 20.

980 Zorita, E., Cuscó, P. & Filion, G.J. 2015. Starcode: sequence clustering based on all-pairs
981 search. *Bioinformatics* **31**: 1913–1919.

Supplemental Figures

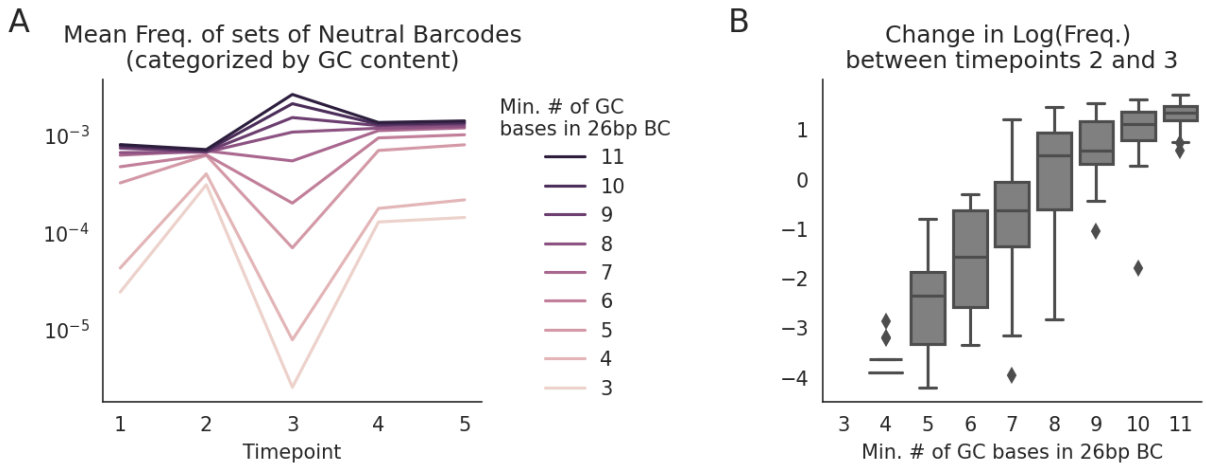

Figure S1. Evidence of GC content affecting barcode frequencies. (A) Dynamics of the mean frequency of putatively neutral lineages carrying barcodes with different GC content (unpublished data). This experiment featured two 26 bp barcodes; different lines show the minimum number of G or C bases in the two barcodes. In the absence of GC-content-dependent biases, all lines should be parallel. **(B)** Change in log-frequency between timepoints 2 and 3 in **(A)**. This change is expected to be independent of GC content. We note that GC-content bias was highly variable between samples in this experiment, suggesting that the specific library preparation conditions contribute to this effect. We also note that this is the strongest example of bias we have observed so far.

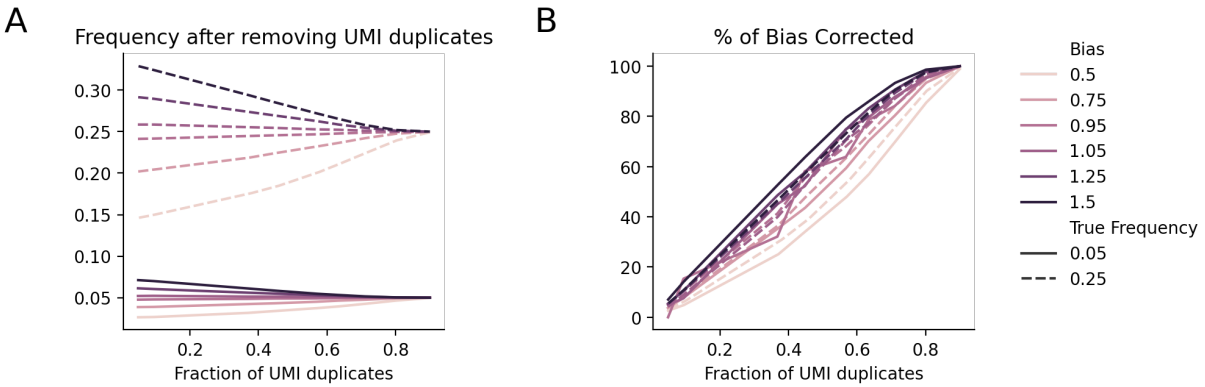

Figure S2. Removing UMI duplicates rarely corrects biases in amplification. (A) The frequency of a simulated focal barcode with a library-preparation bias (e.g. PCR amplification bias) after removing UMI duplicates as a function of the fraction of UMI duplicates. **(B)** The percent of the difference between the true frequency and read-based frequency of the focal barcode that is corrected as a function of the fraction of UMI duplicates. Data is the same as in **(A)**.

Supplemental Tables

Study	Description	Read file used (SRR accession)	Approximate Library Size (K)	Reads	Barcode Design (length / information)
Johnson et al. 2019	Timepoint 0 from a yeast RB-TnSeq experiment	SRR9850741	400,000	18,560,760	NNNNCANNNNCANNNNCANNNNCANNNN (28 bp / 40 bits)
Levy et al. 2015	Timepoint 0 from a yeast lineage tracking experiment	SRR5747458	500,000	142,918,126	NNNNNAANNNNNAANNNNNTTNNNNN (26 bp / 40 bits)
Jasinka et al. 2020	Initial barcode library for *E. coli* lineage tracking experiment	SRR10556795	50,000	6,131,498	NNNNNNNNNNNNNNN (15 bp / 30 bits)
Eyler et al. 2020	Timepoint 0 from stem-like glioblastoma cell lineage tracking experiment	SRR10704145	50,000	7,465,619	WSWSWSWSWSWSWSWSWSWSWSWSWSWSWS (30 bp / 30 bits)
Ge et al. 2020	Timepoint from breast cancer cell line lineage tracking experiment (JQ1 treatment, passage 11, rep. 3)	SRR9162708	80,000	11,809,554	WSWSWSWSWSWSWSWSWSWSWSWSWSWSWS (30 bp / 30 bits)
Borchert et al. 2022	Timepoint 0 from a *Pseudomonas putida* RB-Tnseq experiment (M9 + 20 mM D-glucose, Replicate A)	SRR18112661	200,000	5,618,453	NNNNNNNNNNNNNNNNNNNN (20 bp / 40 bits)

Table S1. Datasets reanalyzed in this paper. Approximate library sizes are based on preliminary error correction using Deletion-Correct.

Dataset	No BC extracted by either method	At least one BC extraction succeeded				
		BCs Match	Match with 1-3 edits	Mismatch	Regex Failed	Alignment Failed
Johnson et al. 2019	1.67%	98.058% (66,266 BCs)	0.589% (409 BCs)	0.018% (11 BCs)	1.300% (973 BCs)	0.036% (8 BCs)
Levy et al. 2015	3.82%	98.726% (80,386 BCs)	0.155% (141 BCs)	0.057% (49 BCs)	0.995% (833 BCs)	0.067% (63 BCs)
Jasinka et al. 2020	1.87%	99.302% (33,555 BCs)	0.269% (81 BCs)	0.015% (6 BCs)	0.413% (127 BCs)	0.001% (1 BCs)
Eyler et al. 2020	2.87%	97.588% (36,754 BCs)	0.310% (238 BCs)	0.045% (39 BCs)	2.056% (1618 BCs)	0.001% (1 BCs)
Ge et al. 2020	3.33%	98.837% (16,492 BCs)	0.095% (45 BCs)	0.080% (53 BCs)	0.981% (480 BCs)	0.007% (6 BCs)
Borchert et al. 2022	5.94%	98.468% (66,554 BCs)	0.165% (132 BCs)	0.416% (385 BCs)	0.314% (268 BCs)	0.638% (596 BCs)

Table S2. Comparison of two barcode extraction methods on 6 published datasets. Each row represents one barcode sequencing dataset used for testing. The first 100,000 reads were used to test a regex-based barcode extraction method and an alignment-based barcode extraction method. We report the percentages of reads and number of unique barcodes identified by both methods or only one method (e.g. “Regex Failed” indicates cases where the alignment method identified a barcode in the read but the regex method did not).

Dataset	Number of Extracted Sequences	Starcode	Bartender	Shepherd	Deletion-Correct
Borchert et al. (2022)	336,219	260,684	266,068	246,583	236,428
Johnson et al. (2019)	719,584	447,068	455,360	426,998	381,047
Levy et al. (2015)	2,086,173	500,565	539,250	480,067	500,806
Simulation	1,544,849	99,581	100,257	99,615	99,152

992 **Table S3.** Number of identified barcodes before and after error correction for three empirical datasets and
993 simulated data across four error correction methods.