Targeted ExSeq -- Probe Generation V.2

2

Anubhav Sinha[1,2,3], Fei Chen[4,3], Shahar Alon[1,5,3], Ed Boyden[6,1,7,8,9,10,3]

[1]McGovern Institute, MIT; [2]Harvard-MIT Program in Health Sciences and Technology;
[3]Human Tumor Atlas Pilot Project; [4]Broad Institute of MIT and Harvard;
[5]Faculty of Engineering, Gonda Brain Research Center and Institute of Nanotechnology, Bar-Ilan University;
[6]Department of Media Arts and Sciences, MIT; [7]Department of Biological Engineering, MIT;
[8]Koch Institute for Integrative Cancer Research, MIT; [9]Howard Hughes Medical Institute;
[10]Department of Brain and Cognitive Sciences

2

Nov 29, 2021

1

dx.doi.org/10.17504/protocols.io.byrbpv2n

NCIHTAN

Anubhav Sinha

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

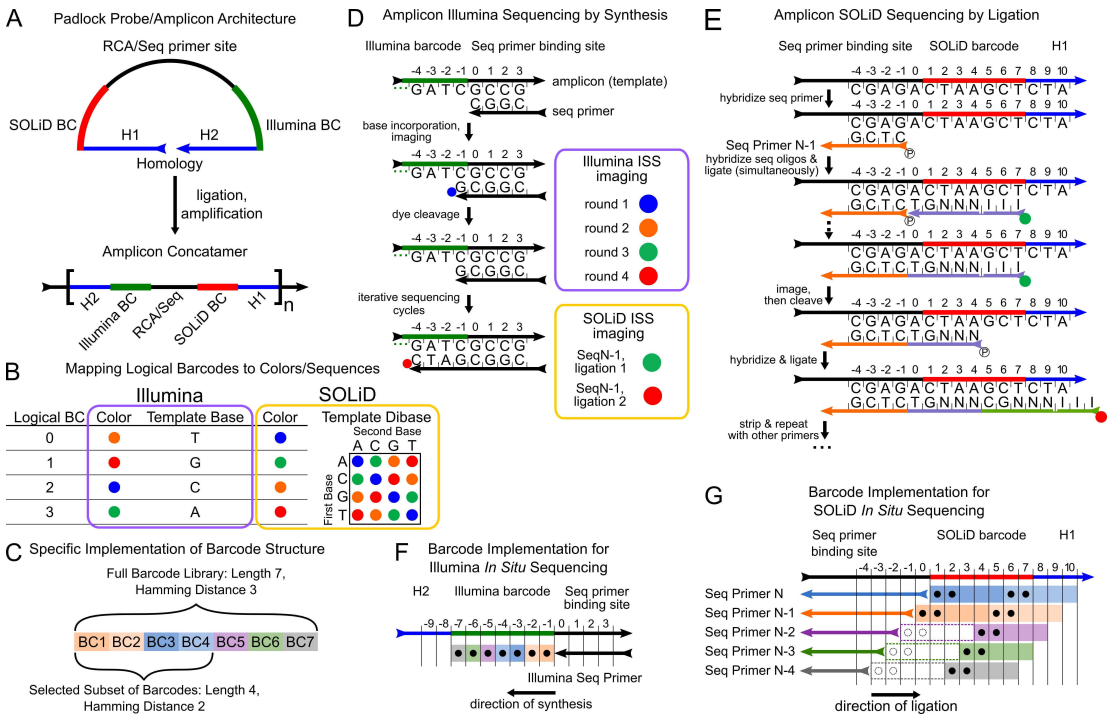

Figure 1. Summary of padlock probe structure, barcode design, and barcode readout processes. **(A)** Top: architecture of padlock probe. Bottom: amplicon sequence after RCA. Note that the amplicon sequence is the reverse complement of the probe sequence. **(B)** Barcodes are defined in a logical space, consisting of a sequence of the numbers 0-3, i.e. 2012310. These logical barcodes are implemented as specific nucleotide sequences for readout with different sequencing chemistries. The details of the mapping between logical barcodes and Illumina sequencing and SOLiD sequencing barcode nucleotides and fluorophores are shown here. **(C)** The specific implementation of the barcode structure that is discussed in this protocol. The barcode library consists of length 7 barcodes with Hamming distance 3. A subset of the total barcode library with length 4 and Hamming distance 2 is embedded within this library. **(D)** An illustration of the process of four rounds of Illumina sequencing by synthesis on an example sequence. **(E)** An illustration of two rounds of SOLiD sequencing by ligation on on an example sequence. **(F)** The specific implementation of the logical barcoding scheme in (C) for nucleotide sequences read out through the Illumina chemistry. **(G)** Like (F), but specifying the nucleotide sequences for the SOLiD chemistry implementation discussed here.

This protocol accompanies Expansion Sequencing (ExSeq), and describes the process of generating padlock probes for targeted ExSeq. The steps detailed here are a generalization of the protocol used to generate padlock probes for figures 4-6, and represent our recommendations for future users of the technology. This protocol has associated code (available at https://github.com/anusinha/TargetedExSeqProbeGen). We have added additional technical explanations here.

As a technical summary of the experimental steps of the method, targeted ExSeq begins with physical expansion of a sample of interest (using the chemistry of expansion microscopy), resulting in a tissue-hydrogel composite with the RNA molecules covalently anchored to the hydrogel. An *in situ* sequencing library is then prepared from a set of transcripts of interest. The library preparation begins with the hybridization of oligonucleotide padlock probes bearing barcodes to RNA transcripts of interest. Multiple padlock probes target one gene, and each of those padlocks have the same barcode identifying that gene. Each gene has a unique barcode.

The padlock probes are subsequently enzymatically ligated and amplified using rolling circle amplification (RCA), producing amplicons (often called RCA colonies, or

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

"rolonies" colloquially). The amplified barcode sequences on the amplicon are read out through the process of *in situ* sequencing, in which sequencing reagents are deployed within the gel. Through iterative rounds of chemistry and imaging, the barcode sequence is read out, one base at a time. The imaging data is processed through the ExSeq image processing pipeline (https://github.com/dgoodwin208/ExSeqProcessing/wiki), resulting in a map of reads in 3D space.

This protocol describes the process by which padlock probes are generated. Because the experimental design governs the design of the padlock probes, this protocol is necessarily less prescriptive than the other protocols.

The overall architecture of the padlock probes is shown in **Fig. 1A** (top). The probes are doubly barcoded, enabling the user to read out the barcode sequence either via SOLiD or Illumina sequencing chemistry. The probes consist of the following sequence regions: (1) a split homology region (H1, H2), which hybridizes to the transcript of interest; (2) a barcode for SOLiD sequence readout that is 5' to the Seq Primer; (3) a barcode for Illumina sequencing that is 3' to the Seq primer; and (4) a universal RCA/Seq primer binding site. After RCA, the amplicon concatamer **Fig. 1A** (bottom) is produced. Note that the concatamer is the reverse complement of the probe sequence. *In situ* sequencing is then performed on the concatamer.

The padlock probe generation process consists of five stages: (1) Target selection; (2) Barcode generation; (3) Barcode assignment; (4) Homology region generation; (5) Probe assembly and export.

The first stage of the design is target selection of genes of interest. As this is dependent on the scientific question, generalizable advice is difficult to provide. This protocol assumes that the user has already selected a set of transcripts of interest. Relevant considerations are provided in Step 2.

The second and third stages of the design process are closely linked. In these steps, the barcode space is defined (Steps 3-5), and those sequences are assigned to transcripts (Steps 6-7).

In the fourth stage, the homology regions of the padlock probes that hybridize to the RNA are identified (Steps 8-10). In the fifth stage, the homology regions, barcode sequences, and backbone sequences are assembled to form the final probe (Steps 11-14).

The probes are then ordered and pooled together to form the pooled padlock probe libraries (Steps 15-16), used in the Targeted ExSeq Library Preparation protocol.

To enable multiple sequencing chemistries to be used, barcodes are designed in a logical space, which is agnostic to the sequencing chemistry. At the time of assembly, these logical barcodes are converted to nucleotide sequences, which are read out during *in situ* sequencing with the SOLiD or Illumina sequencing chemistries. The

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

logical barcodes are a repeated sequence of the numbers {0, 1, 2, 3}. For example, 2012310 is a logical barcode of length 7, in which the first base of the barcode is 2, the second is 0, the third is 1, and so on.

The mapping between logical barcodes and nucleotide sequences is shown in **Fig. 1B**. The details of this mapping are discussed in further detail in Steps 11-14. Examples of these mappings being used in the *in situ* sequencing process are shown in **Fig. 1D** (Illumina sequencing) and **Fig. 1E** (SOLiD sequencing).

To constrain this design space for this protocol, we focus on a specific implementation of the padlock probe, shown in **Fig. 1C**.

First, the barcode design: we focus on barcodes of length 7, with Hamming distance 3, which enables 1-bit error correction, and 2-bit error detection. This barcode library enables the interrogation of ~330 targets with 7 rounds of sequencing. We also discuss a subset of this barcode library, which enables the interrogation of up to ~40-50 targets with four rounds of sequencing. This subset is of length 4, with Hamming distance 2, enabling 1-bit error detection. The embedded subset is useful for smaller experiments, with 10-50 targets, where the readout can be performed with four rounds of sequencing. Notably, this subset of barcodes is extensible, i.e. additional genes can be added to the library if 7 rounds of sequencing are performed.

Second: we design doubly barcoded probes. These padlock probes carry two barcode sequences for readout using either the Illumina or SOLiD sequencing chemistries. The implementation details on the mapping between logical barcodes and the specific nucleotide sequences are shown in **Fig. 1F** (Illumina) and **Fig. 1G** (SOLiD).

Third: the total probe length is 74 nucleotides, corresponding to the design criterion of a padlock probe being at least 2*n+10 nucleotides long, where n is the total length of the homology region (in this case, n=32 nt).

Fourth: we generate up to 16 probes per transcript, which we find strikes a good balance between yield and cost.

This protocol was used (with an earlier version of the codebase) to design probes to profile human metastatic breast cancer biopsies as a part of the Human Tumor Atlas Pilot Project (HTAPP). For the design of the SOLiD sequencing barcodes, the alternative sequential barcode design was utilized (see Step 12.2).

Worked Example

Step 17 is a worked example of generating padlock probes for AKT1, CDK7, and KRT17, using pre-prepared input files. The output results of the script can be compared to reference results.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

DOI

dx.doi.org/10.17504/protocols.io.byrbpv2n

https://pubmed.ncbi.nlm.nih.gov/33509999/

Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden 2021. Targeted ExSeq -- Probe Generation. **protocols.io**
https://dx.doi.org/10.17504/protocols.io.byrbpv2n
Anubhav Sinha

protocol

Alon S*, Goodwin DR*, Sinha A*, Wassie AT*, Chen F*, Daugharthy ER**, Bando Y, Kajita A, Xue AG, Marrett K, Prior R, Cui Y, Payne AC, Yao CC, Suk HJ, Wang R, Yu CJ, Tillberg P, Reginato P, Pak N, Liu S, Punthambaker S, Iyer EPR, Kohman RE, Miller JA, Lein ES, Lako A, Cullen N, Rodig S, Helvie K, Abravanel DL, Wagle N, Johnson BE, Klughammer J, Slyper M, Waldman J, Jané-Valbuena J, Rozenblatt-Rosen O, Regev A; IMAXT Consortium, Church GM***+, Marblestone AH***, Boyden ES***+ (2021) Expansion Sequencing: Spatially Precise In Situ Transcriptomics in Intact Biological Systems, Science 371(6528):eaax2656. (* equal contribution, ** key contributions to early stages of project, *** equal contribution, +co-corresponding authors)

Targeted Expansion Sequencing Protocols

Targeted Expansion Sequencing Protocols

in situ sequencing, expansion sequencing, expansion microscopy, targeted ExSeq, ExSeq, spatial transcriptomics, spatially resolved transcriptomics, spatial omics, padlock probes

protocol ,

Oct 04, 2021

Nov 29, 2021

53763

Part of collection

Targeted Expansion Sequencing Protocols

Targeted Expansion Sequencing Protocols

This protocol assumes familiarity with MATLAB.

We have encountered issues with the probe generation if the working directory is synced with Dropbox.

Software requirements/dependencies:

- MATLAB with Bioinformatics Toolbox
- R with DNABarcodes library
- Legacy BLAST (2.2.26) and BLAST+ (2.2.26) from NCBI

This script has been tested primarily with MATLAB R2019b on Windows 10. Future versions of MATLAB may not support blastlocal (within the Bioinformatics Toolbox), but previous releases will retain this functionality. MATLAB does not support blastlocal in MacOS 10.15 (Catalina) and later (see note in Step 1.1).

Experimental:

- Basic lab wetware: microcentrifuge tubes, 50 mL tubes, pipette tips

Please carefully read all safety datasheets for all reagents used in the protocol, and perform all steps in accordance with relevant guidelines.

:

Preparation 1h

1 Setting up BLAST with MATLAB and Building BLAST Database

45m

The selection of appropriate homology regions for probe generation screens candidate transcript regions against the transcriptome to eliminate sequences with off-target homology. In this section, BLAST is installed locally, the files needed to generate a species-specific BLAST database are downloaded, and the database is generated.

In addition, the scripts for the rest of the protocol are downloaded from GitHub.

1.1 Legacy BLAST Installation

20m

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation https://dx.doi.org/10.17504/protocols.io.byrbpv2n

Legacy versions of BLAST are used to generate the BLAST database (using BLAST+ 2.2.26) and run the BLAST query with the blastlocal command in MATLAB (using BLAST 2.2.26). The reference page for blastlocal states that only version 2.2.17 is supported; in practice, BLAST 2.2.26 works.

Download, extract, and install version 2.2.26 of BLAST here (https://ftp.ncbi.nlm.nih.gov/blast/executables/legacy.NOTSUPPORTED/) and version 2.2.26 of BLAST+ here (https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/), taking care to select the appropriate installer for your system.

Save the extracted files to an appropriate location, and add this location to the MATLAB path and the system path variables.

To add the location to the MATLAB path: navigate to the Home tab, then Set Path (in the Environment section of the menu). In the window that pops up, click on Add Folder and navigate to the directory where the executable files have been installed. Then click Save to add the location to the MATLAB path.

Modifying the system path varies by operating system. For Windows, this support page may be useful. For OS X, this webpage may be useful.

Warning: blastlocal is NOT compatible with Mac OS X 10.15 or higher. Additionally, blastlocal is slated to be removed in a future MATLAB release. When this occurs, an older version of MATLAB could be used to run the code. Alternatively, the dependence could be replaced with a system call, with appropriate parsing of the output. The underlying system call from blastlocal is "blastall -i "temp.fa" -p blastn -d <refseqdatabase>.fna -S 1 -e E_VALUE" (see Step 9), which is parsed by the blastreadlocal.m built in MATLAB function. If blastreadlocal.m is removed, the function could be copied from an older version of MATLAB's Bioinformatics toolkit to parse the system call output.

1.2 Downloading the RefSeq mRNA Library for Species of Interest

20m

Download the appropriate compressed sequence files for the organism of interest from the RefSeq FTP site (**https://ftp.ncbi.nlm.nih.gov/refseq/**). Navigate to the correct organism (e.g. H_sapiens for human), then to the mRNA_Prot directory.

Download all of the *.rna.fna.gz files, extract them, then concatenate the *.fna files together together.

E.g. in bash/Mac OS Terminal:

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation https://dx.doi.org/10.17504/protocols.io.byrbpv2n

```
cat *.fna > humanrefseq.fna
```

In Windows Command Prompt:

```
type *.rna.fna > humanrefseq.txt

ren humanrefseq.txt humanrefseq.fna
```

Note that the concatenated file must first be output to a file with a different file extension; if the output of the type command were output to a .fna file, Command Prompt would have included that output .fna file as an additional input file. The concatenated file is saved to a txt file, then renamed to the fna extension.

1.3 Generating BLAST Database for Species of Interest

5m

We now generate the BLAST database using the BLAST tools.

In the terminal, navigate to the directory where the concatenated sequences file is saved, and run the following command to generate the database (where the input file is named as determined by the previous step).

```
makeblastdb -in humanrefseq.fna -dbtype nucl
```

Save the resulting .nhr, .nin, .nsq files in the working directory for probe generation.

1.4 Get Scripts from GitHub

Download scripts from GitHub repository (https://github.com/anusinha/TargetedExSeqProbeGen) and save to working directory.

Gene Selection 5m

2 Identification of Genes of Interest

5m

This protocol assumes that the user has selected a set of genes of interest with their RefSeq

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

Nucleotide accession numbers. Some relevant considerations for gene selection are described here.

Transcript length: it is easier to generate more probes for longer transcripts. A larger number of probes increases the detection efficiency. However, it is still possible to generate probes to detect shorter transcripts. We generally aim to generate 16 probes per transcript, but have had good results in tissue with 8 probes per transcript. As a rough number, there is a good chance of generating 16 probes for most transcripts that are >1.5 kb in length.

Splice variants: for transcripts with multiple splice variants, the RefSeq annotations can aid in selecting the specific splice variant to use. A common strategy is to look for the transcript that has the most number of shared exons across all transcript variants (roughly the intersection of exons). This is often the shortest variant, but not necessarily always.

For sequences that are not in RefSeq (i.e. custom sequences), they can be saved in GenBank format and added to the generated GenBank file, with a custom string identifier in the locus name and accession number fields (i.e. "Plasmid001"). This can be used in place of the accession number in subsequent steps.

2.1 Downloading Transcript Sequences From RefSeq

5m

Transcript sequences can be downloaded from RefSeq in the GenBank format.

Prepare a comma separated list of all accession numbers (e.g. for the genes AKT1, CDK7, and KRT17 in human: NM_001014432.1, NM_001799.4, NM_000422.3) and copy that list into the search field of the RefSeq Nucleotide search.

From search results, click on: Send to -> Complete Record -> File -> GenBank (full) -> Create File.

Download GenBank file and save to working directory.

Barcoding Library Design

3 Overview of Barcoding Scheme

In a subsequent step, barcodes will be assigned to the genes of interest. Here, the space of potential barcodes is defined.

Barcodes are designed in a logical space. The logical barcode is a sequence of numbers from 0 to 3, where each number represents a color that will be read out during the *in situ* sequencing process, i.e. 2012310 is a potential barcode of length 7, where the order of colors detected is: 2, 0, 1, 2, 3, 1, 0. The specifics of which physical color (dyes, filters) depends on the sequencing chemistry.

This level of abstraction is useful because the mapping between a logical barcode and the

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

physical nucleotide sequence can be non-obvious. Designing the barcode in the logical space helps to compartmentalize the implementation details to a subsequent step. It also enables extensibility -- if a new sequencing chemistry is used, the barcoding scheme doesn't need to be changed. Only the additional code for the mapping between logical barcodes and nucleotide sequences for the new chemistry needs to be added.

The barcode library consists of the set of individual barcodes that could potentially be assigned to a transcript. The library is an implementation of an overall barcoding scheme, a specification of properties that the library must satisfy. The typical properties considered are length and minimum Hamming distance.

The length determines the total number of rounds of sequencing needed to read out the barcode. The Hamming distance between two barcodes is the number of positions at which the two barcodes differ. As an example, the barcodes 0303101 and 1201301 are both length 7, and have a Hamming distance of 4, as they differ in position 1, 2, 4, and 5. The minimum Hamming distance for a set of barcodes is the minimum Hamming distance between any two different barcodes in the set, and determines the error-robustness and error-correction properties. For a barcode library with minimum Hamming distance of k, the number of bit errors that can be detected is k-1, and the number of bit errors that can be corrected is $\left\lfloor \frac{k-1}{2} \right\rfloor$.

Here, several different options for generating a barcode library from a barcode scheme are presented. Option 1 is to use a pre-designed library, which has several useful properties. Option 2 is to create a custom barcode library from a specification. The primary focus is on generating barcode libraries with length 7 and Hamming distance 3 as in the specification in **Fig. 1C**.

4 [Option 1] Using a Pre-Generated Barcode Library

A pre-generated barcode library is saved in the GitHub repository (prioritizedBarcodeLibrary.csv). The library was generated by following a variant of the process described in Option 2A.

The barcodes in the library are of length 7, and the library has a minimum Hamming distance of 3. There are 326 barcodes in the library. In addition, there is a subset of 58 barcodes, that, when considering only the first four positions of the barcode, have a minimum Hamming distance of 2. As described in the abstract, this enables smaller scale experiments to be performed that interrogate <= 58 targets, that can be read out in four rounds of sequencing. An overview of the barcode structure is shown in **Fig. 1C**.

The barcodes are prioritized as follows:

Priority 1: this set of 47 barcodes satisfies the following properties:

- contain at least three unique values within the first four digits of the barcode
- are approximately base-balanced (as a set)

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

- these barcodes, along with those in priority 2 contain the subset library of length 4, Hamming distance 2

Priority 2: this set of 11 barcodes is the remaining set of barcodes in the subset library with the embedded length 4, Hamming distance 2 code.

Priority 3: this set of 258 barcodes contains at least three unique values within the entire length 7 barcode.

Priority 4: this set of 10 barcodes is the remaining set of barcodes from the original set. They have low complexity, containing only two unique values within the barcode.

For experiments with <= 47 targets, priority 1 can be used. For experiments with <= 58 targets, priority 1 and 2 can be used. For larger experiments, priorities 1-3 can be used, with the inclusion of priority 4 only if necessary.

The file (prioritizedBarcodeLibrary.csv) contains 327 rows, including a header row. The file contains three columns: barcode sequence (in the logical barcode-space), priority, and diversity (number of distinct values in the barcode sequence). Note that the barcodes are saved with a preceding single-quote (i.e. '0132200) to avoid Excel interpreting the barcode as a number and dropping leading zeros.

4.1 Technical Note on Library Generation

To generate this library, the script in option 2A was run many times, generating sets of length 7, minimum Hamming distance 3 barcode libraries. Each of these barcode libraries were split into a priority A subset, containing all barcodes with at least three unique barcode values (from 0 to 3), and a priority B subset containing the remaining barcodes.

The script was also used to generate a length 4, minimum Hamming distance 2 barcode library, and split it similarly into priority 1 and priority 2 subsets, based on the same diversity criterion.

The full-length barcode libraries were queried to find the library that contained the largest number of elements from the length 4, hamming distance 2 barcode library in the priority A subset, and the best library was selected.

The priority 1 barcodes were identified by finding priority 1 length-4 barcodes within the priority A length-7 barcodes. As each individual length-4 barcode could map to multiple full length-7 barcodes in the library, an assignment step was needed to expand length-4 barcodes to unique length-7 barcodes. A base-balanced barcode set was also sought. To generate the final barcode set, sets of length-7 barcodes were generated from the length-4 barcodes by randomly selecting one of the matching length-7 barcodes for each length-4 barcode. These barcode sets were tested for base balance, and the best one was

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

selected, defining a library of 47 priority 1 barcodes.

The remaining length-4 barcodes (that did not map to priority A length-7 in the prior step) were mapped to full length-7 barcodes, forming priority 2.

Priority 3 was defined by the remaining priority A length-7 barcodes, and priority 4 by all remaining barcodes.

5 [Option 2] Generating a Custom Barcode Library

In Option 2A, the DNABarcodes R package is used to generate a barcode library. A user can also devise their own way to generate the barcode library, noted in Option 2B.

5.1 [Option 2A] Using the DNABarcode R Package

A barcode library with specified length and minimum Hamming distance can be generated using the R package DNABarcodes associated with this paper: https://academic.oup.com/bioinformatics/article/33/6/920/2804018.

Install the R package DNABarcodes using the instructions here: https://www.bioconductor.org/packages/release/bioc/html/DNABarcodes.html.

Then, run the following R script (in the GitHub as /utilities/generate_barcode_library.R), which writes the barcode library to the file barcode_set.csv. This example script generates a barcode library with ~330 barcodes of length 7 and minimum Hamming distance 3.

```
library("DNABarcodes")

bcLength <- 7
minHammingDist <- 3
popSize <- 1000
numIter <- 500

mySet <- create.dnabarcodes(bcLength, dist =
minHammingDist, heuristic = "ashlock", cores = 4,
filter.triplets = TRUE, filter.gc = FALSE, population =
popSize, iterations = numIter)
mySet <- paste0("'", gsub("T", "3", gsub("G", "2",
gsub("C", "1", gsub("A", "0", mySet)))))

write(mySet, file = "barcode_set.csv")
```

The bcLength parameter is the barcode length and minHammingDist is the minimum Hamming distance. The parameters popSize and numIter are parameters for the evolutionary algorithm used to generate the barcode

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation https://dx.doi.org/10.17504/protocols.io.byrbpv2n

library; for these values, the software will take ~40-80 minutes to run on a typical laptop. The number of cores can be set appropriately to your system. The population size and number of iterations can be decreased to speed up runtime, but the library size may be slightly smaller.

The output of the create.dnabarcodes function is in nucleotides. This is converted to numerical values by substituting 0 for A, 1 for C, etc. A single quote (') is prepended to every entry to avoid Excel interpreting the csv file as numerical data, which would strip leading zeros.

The function create.dnabarcodes is a stochastic evolutionary algorithm, meaning that the output will be different each time.

5.2 [Option 2B] User-Generated Approach

There are other approaches that a user could devise that may suit their particular application better.

Assignment of Barcodes to Transcripts

6 Overview of Barcode Assignment to Transcripts

In the previous steps, genes of interest were selected, and a barcode library was generated. In this step, approaches for assigning barcodes to specific transcripts are suggested.

To minimize optical crowding and for better base-calling performance in the analysis of the *in situ* sequencing data, it is advantageous to distribute the amplicons across all color channels and rounds roughly evenly.

This can be done in absence of any expression data, but can be improved if single-cell (or bulk) expression data is available.

6.1 [Option 1] Random Assignment of Barcodes to Transcripts

If no expression data is available, transcripts can be randomly assigned to barcodes.

6.2 [Option 2] Barcode Assignment Using scRNA-Seq

If single-cell RNA sequencing data is available, clustering into cell types can be performed, and mean expression profiles of the genes of interest for each cell type can be computed. This results in a in a cell types by genes of interest matrix.

This matrix can potentially be used to guide the barcode assignment. One simple approach is to generate candidate barcode assignments, and score them for color balance across rounds using the cell type profiles matrix.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

For a particular barcode assignment, a 3D cost matrix can be computed, where each entry in the matrix represents the total expression in a particular color channel, in a sequencing round, for a particular cell type. The size of the cost matrix is the number of rounds x number of colors x cell types. One possible metric to summarize the cost matrix is the L2 (Euclidian) norm of the cost matrix, though other summary indices are possible. To generate the barcode assignment, a large number of potential barcode assignments can be randomly generated, and the one with the lowest total cost selected.

This can also be done with bulk RNA-Seq if that data is available.

Compilation of Gene/Barcode List

7 Compilation of Gene/Barcode List

Tabulate the results from the previous steps into the genelist Excel file.

The columns are gene symbol, RefSeq accession number, minimum Tm threshold (see note describing this below), and barcode.

It is **critical** that the gene symbol match the symbol in RefSeq (including case). For example, human gene symbols should be in all caps (i.e. GAPDH), while mouse gene symbols should have the first letter capitalized (i.e. Gapdh). Certain genes, such as KIT in humans, are commonly referred to by a modified name such as c-KIT. However, the symbol in the genelist must be KIT, since that is the entry in RefSeq.

An example spreadsheet for the worked example (see step 17, example_genelist.xlsx in GitHub) is shown in **Table 1**. No header line is included in this spreadsheet.

A	B	C	D
AKT1	NM_001014432.1	55	'3212010
CDK7	NM_001799.4	53	'2102033
KRT17	NM_000422.3	52	'0221001

Table 1. Example genelist spreadsheet (example_genelist.xlsx)

Note again the prepended quote in front of the barcode sequence; this prevents Excel from interpreting '0221001 as the number 221001.

The minimum Tm threshold is the minimum melting temperature allowed for each individual homology arm of the padlock probe. This is used to screen potential homology regions in a subsequent step.

Overview of Padlock Design

8 Padlock Probe Architecture

The overall design of the padlock probe is shown in **Fig. 1A**.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

The generation of the homology regions (corresponding to H1 and H2 in **Fig. 1A**) is described in steps 9-10.

The generation of the SOLiD and Illumina barcode sequences (in **Fig. 1A**) is described in steps 11-14.

Under the standard probe design process, the RCA/Seq primer site is the fixed sequence 'TCTCGGGAACGCTGAAGACGGC', and can be optionally modified in Step 13.

Generation of Homology Regions

9 Overall Approach to Generating Homology Regions

In this step, the homology regions used in the padlock probes are generated. These regions are H1 and H2 in **Fig. 1A**.
This is implemented in the MATLAB script Targeted_ExSeq_ProbeGeneration_Homology.m.

For each transcript, a sliding window search for acceptable homology regions is performed. Candidate regions of the transcript are screened for acceptability. If a region is acceptable, the search window advances past the end of the accepted region; if the region is unacceptable, the window is shifted by one nucleotide and the process is repeated. Acceptable regions are collected and written out to a file in the output directory, with one file (named by the accession number) per gene of interest.

Criterion for exclusion of a homology region are any of the following:

1. Containing a single nucleotide repeat that is five bases or longer (i.e. 'AAAAA', 'CCCCC', 'GGGGG', 'TTTTT').
2. Containing three or fewer different nucleotides (i.e. A, C, G bases only).
3. GC content of either half of the homology region (corresponding to the two homology arms of the padlock probe at the 5' and 3' ends of the probe) outside of the specified range (see below for parameter specification).
4. Tm of either half of the homology region below the Tm threshold (defined on a per-gene basis in the genelist)
5. Tm difference between the two halves of the homology region outside of the maximum tolerated difference (see below for parameter specification).
6. Presence of secondary structure in the homology region: hairpins, defined by a minimum hairpin base length of 7 and loop length of 6; and dimers, defined with a minimum length of 10.
7. A BLAST hit (with E value below given threshold) to a different transcript that has more matching bases than a given threshold, and that is well-centered with greater than or equal to a given number of bases of the hit on both sides of the ligation junction (see below for parameter specification).

To run the script, specify the parameters that require user input (and optionally those that can be varied), and run the MATLAB script. Output is saved in a directory (specified below), with

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

one file per gene, named by the accession number. The format of each output file is a FASTA file, with the headers containing the probe number, position in transcript, and minimum Tm threshold for the gene, with the sequences being the reverse complement of the segments of the transcript that pass all exclusion criteria. The MATLAB console displays search progress. The estimated time to generate homology regions for each gene is 1-10 minutes on a typical laptop computer, though some genes may require a significantly longer search time.

For further technical details, consult inline comments in script (and in the called functions screen_homology_complexity.m, screen_homology_physical.m, screen_homology_blast.m).

Note: when running the script, the file Temp.fa, which containing the sequence of the current homology region being BLASTed, is repeatedly saved and deleted from the working directory. The script deletes Temp.fa at the start of execution (in case a prior run was terminated early, without deleting Temp.fa). However, if the BLAST search did not stop when the script was terminated, the BLAST program needs to be terminated before manual deletion of Temp.fa. If BLAST errors are encountered, the first debugging step should be to verify that Temp.fa was deleted.

9.1 Parameters Requiring User Input

There are four parameters that require user input.

1. genelist_spreadsheet, the location/filename of the genelist.
2. genbank_sequences, the location/filename of the GenBank format file downloaded in Step 2.1.
3. refseq_database, the location/filename of the .fna file generated previously. This filename should include the .fna extension.
4. homology_regions_output, the name of the directory (to be created by the script) where homology regions will be saved.

9.2 Optional Parameters to Set

There are eight parameters that can optionally be changed. They are listed below, along with their default values. The parameters are all stored within the params struct.

Most users will not need to change these parameters, except as described in Step 9.4. These parameters are made available primarily for advanced users.

1. HOMOLOGY_LENGTH = 32, the total length of the homology region queried, corresponding to a padlock probe with two homology arms of length HOMOLOGY_LENGTH/2.
2. GC_RANGE = [40, 65], the minimum and maximum acceptable GC content for the individual homology arms.
3. MAX_TM_SEPARATION = 8, the maximum acceptable difference in the melting temperatures of the individual homology arms.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

4. E_VALUE = 0.05, the maximum acceptable E value for BLAST queries to be returned for potential query sequence exclusion evaluation.
5. SPACER_LENGTH = 0, the minimum separation between adjacent homology regions.
6. BLAST_HOMOLOGY_MIN_LENGTH_CUTOFF = 13, the minimum number of matching bases of a BLAST hit for potential query sequence exclusion.
7. BLAST_HOMOLOGY_LIGATION_JUNCTION_MIN_OVERLAP_CUTOFF = 4, the minimum overlap of the BLAST hit on both sides of the ligation junction for potential query sequence exclusion.
8. REPORT_IF_BELOW_NUMBER_OF_PROBES = 9, the threshold for reporting additional information if too few homology regions are identified.

9.3 Example Output

The first several lines of output are shown for the first gene (AKT1) in the worked example (see Step 17).

NM_001014432.fa

```
>probe_1_index_264_MinTm_55
tttgacccaggctggctcggccttccctaagc

>probe_2_index_317_MinTm_55
cctctgatgcaccagctgacaggctgcctcct

>probe_3_index_425_MinTm_55
caaccctccttcacaatagccacgtcgctcat

...
```

9.4 Typical Use and Modifying Parameters to Increase Number of Homology Regions

In typical usage, the script will be run once, with the same high Tm threshold for all genes to generate draft candidate regions for all genes. For specific genes of interest where too few homology regions were generated, a new genelist will be constructed containing solely those genes requiring further optimization, and the script will be rerun with modified parameters. This process is iterated until enough candidate regions have been generated for all genes in the original genelist.

Several parameters can be modified to increase the number of binding regions. These include:

1. Adjusting the minimum Tm for individual genes: a decrease from 55 to 53 or 51 can potentially identify more homology regions.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

2. Increasing the maximum acceptable Tm range (MAX_TM_SEPARATION) can also potentially identify more homology regions. Together with (1), these two parameters are the most commonly modified parameters.
3. Increasing the range of acceptable GC content (GC_RANGE).
4. Decreasing the E value threshold (E_VALUE), can allow for more non-specific (lower E value) hits to be tolerated.

On adjusting the E value threshold: the E value threshold is the maximum E value for hits returned by BLAST. Hits with higher E values are ignored for further consideration for elimination of the candidate binding region. Lowering the E value threshold allows for more non-specific (E values above the threshold) hits to be tolerated. For example, consider a candidate sequence with two hits, one self-hit in the original gene of interest with E value=1e-8, and an off-target partial homology match with E value=0.002. If the E value threshold is set to E_VALUE=0.05, the off-target hit will be considered for further exclusion criteria. However, if the E value is lowered to E_VALUE=0.001, the off-target hit will no longer be considered for further exclusion, and the sequence may be accepted as a homology sequence.

10 [Optional] Validating Homology Regions

This is an optional step to validate the homology regions identified in the previous steps. This validation step consists of re-BLASTing the homology regions primarily to confirm that there are no off-target hits resulting from the process by which self-hits and pseudogene hits are tolerated.

When searching for homology regions in a gene, hits in splice variants, pseudogenes, and non-coding variants of that transcript should be tolerated. This is implemented by searching the gene names of hits to ensure they start with the gene symbol of the gene of interest. For example, a hit in the KRT17 pseudogene KRT17P1 would be tolerated, as the gene symbol "KRT17P1" begins with "KRT17".

However, this may potentially create artifacts. As an example, when designing probes for CD4, there may potentially be a hit in CD44. Since "CD44" begins with "CD4", the hit would be tolerated, and the homology region could be accepted. The output of this script, with manual review, corrects for this potential issue.

To identify these events and exclude them if necessary, the previously found homology regions are BLASTed again under the same search criteria used previously, except without gene symbol/accession number filtering. Hits are collected, and the results are saved to a spreadsheet for manual verification. If problematic sequences are identified, they can be individually removed from the FASTA files containing the homology regions.

The script for this step is Targeted_ExSeq_ProbeGeneration_Validate_Homology_Regions.m.

10.1 Parameters Requiring User Input

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

There are four parameters that require user input.

1. genelist_spreadsheet, the location/filename of the genelist.
2. refseq_database, the location/filename of the .fna file generated previously. This filename should include the .fna extension.
3. homology_regions_input, the name of the directory (created by the previous script) where homology regions are saved.
4. output_filename, the filename of the spreadsheet where the summary of the homology region validation will be saved.

10.2 Optional Parameters to Set

There are four parameters that can optionally be changed. They are the same as the parameters defined previously in Step 9.2. They are listed below, along with their default values. The parameters are all contained within the params struct.

These values should be set to the same values used for the prior search.

1. HOMOLOGY_LENGTH = 32
2. E_VALUE = 0.05
3. BLAST_HOMOLOGY_MIN_LENGTH_CUTOFF = 13
4. BLAST_HOMOLOGY_LIGATION_JUNCTION_MIN_OVERLAP_CUTOFF = 4

10.3 Output and Interpretation

The output spreadsheet is saved to the file indicated by the parameter output_filename. This spreadsheet contains one row for every homology region generated by the previous script. The first two columns of the spreadsheet list the gene symbol and the homology region sequence. The third column onwards lists the gene symbols for the BLAST hits that pass filtering. Self-hits are included here, as no gene symbol filtering is performed.

An example excerpt of the output for Human KRT17 from the worked example (see Step 17) is shown in **Table 2**. There are hits in KRT17 pseudogenes for all of the homology sequences identified, which are tolerated for this application.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

A	B	C	D	E
KRT17	gcaggcacacaggagaagggctggagaggaga	KRT17	KRT17P1	KRT17P2
KRT17	gagccaaagctgtagcagctggagtagctgct	KRT17	KRT17P1	KRT17P2
KRT17	tctcacctccagccagcagcccatcaacaccc	KRT17	KRT17P1	KRT17P2
KRT17	ggctgtgaggatcttgttctgcagctcctcaa	KRT17	KRT17P1	
KRT17	ctgcagctctatctccaaggcctgcatggtgc	KRT17	KRT17P1	KRT17P2
KRT17	tgtctccgccaggttgccctccagggatgctt	KRT17	KRT17P1	KRT17P2
KRT17	caccggttctttcttgtactgagtcaggtggg	KRT17	KRT17P1	KRT17P2
KRT17	tggacctcttccacaatggtacgcacctgacg	KRT17	KRT17P1	KRT17P2
KRT17	caaggaagcatggggaagggactgaagcaggg	KRT17	KRT17P1	

Table 2. Excerpt of output for KRT17 example.

The user should look through the results and ensure that hits to other transcripts (if any) are tolerable (i.e. are hits in pseudogenes or very closely related homologs). If a hit is unacceptable, that homology region should be removed from the relevant FASTA file in the output of the previous script.

Generation of Barcode Sequences, Probe Assembly, and Export

11 Overview of Steps

In the previous script, sequences for the homology regions H1/H2 in the padlock probe (**Fig. 1A**) were identified.

In this section, three tasks are accomplished: (1) converting the logical barcode sequences to nucleotide sequences that can be read out with the SOLiD and Illumina chemistries (corresponding to SOLiD BC and Illumina BC in **Fig. 1A**); (2) assembly of the homology regions, barcode sequences, and fixed RCA/Seq primer site into the final padlock probe sequences (as in **Fig. 1A**); and (3) exporting the padlock probe sequences in spreadsheet form.

This is implemented in the MATLAB script Targeted_ExSeq_ProbeGeneration_Assembly.m.

This script assumes the existence of a genelist spreadsheet, and that the first script to generate homology regions has been run.

12 Mapping Logical Barcodes to Nucleotide Barcodes -- Overview

As described previously, barcodes are designed in a logical barcode space, to allow for the barcode design to be independent of the barcode implementation, which depends on the sequencing chemistry. The barcode readout occurs after rolling circle amplification on the RCA product, which is the reverse complement of the padlock probe sequence (**Fig. 1A**).

Here, the logical barcodes generated in Step 3 are converted to specific nucleotide sequences for readout with the Illumina and SOLiD sequencing chemistries. Technical details on this conversion are provided in the following substeps. An overview of the relationship between the logical barcodes and nucleotides is provided in **Fig. 1B**, with the specific structure of the

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

barcodes implemented here shown in **Fig. 1C**.

12.1 Technical Note: Mapping Logical Barcodes to Nucleotide Sequences -- Illumina Chemistry

In this step, the technical details of the mapping between logical barcodes and nucleotide sequences for readout with the Illumina chemistry are discussed. In short, each logical barcode position directly corresponds to a specific nucleotide position in the barcode sequence on the padlock probe, and the identity of that sequence is determined by the mapping, discussed below.

Barcode sequence amplification: the Illumina barcode sequence is located 3' to the RCA primer binding site on the padlock probe (**Fig. 1A**). After rolling circle amplification, the reverse complement of the original Illumina barcode sequence is located 5' to the sequencing primer binding site (**Fig. 1A, 1D**). Note that the sequencing primer binding site is the reverse complement of the RCA primer binding site.

Barcode sequencing with Illumina chemistry: after hybridizing the sequencing primer to the RCA amplicon, sequencing can proceed. Fluorophore-labeled nucleotides are sequentially added, imaged, and then cleaved using the Illumina chemistry. This allows the barcode sequence to be read out, base by base, using sequencing by synthesis. This is shown schematically in **Fig. 1F** for the 7-base barcode structure shown in **Fig. 1C**.

An example of this process is shown in **Fig. 1D**. The top strand is the RCA product, with the barcode template bases GATC (from 5' to 3'). The sequencing primer is shown as the bottom strand, and four consecutive rounds of *in situ* sequencing are shown, interrogating C, T, A, G in that order, with added bases of G, A, T, and C. The original sequence on the padlock probe was the reverse complement of GATC, which in this case is GATC. The decoded logical barcode is 2031. We note that the barcode sequence on the padlock probe is also the sequence of the nucleotides that are sequentially added during the sequencing by synthesis process.

Barcode construction: the translation of a logical barcode to a nucleotide sequence is a one-to-one mapping from logical values to nucleotides specified by the table in **Fig. 1B**. For the specific barcode architecture used in this protocol (see Step 4), this is shown in the colors shared between **Fig. 1C** and **Fig. 1F** for specific barcode positions.

There are three columns in **Fig. 1B** that are relevant for the Illumina chemistry: the logical barcode, Illumina color, and Illumina template base. The first mapping is between the logical barcode value and the template base, which is the base in the RCA product. The complement of this base is the actual base in the barcode sequence in the padlock probe and is also the

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

added base (as described above). That is, a logical value of '2' corresponds to a template base of 'C', and a padlock and added base of 'G'. The second mapping is between the template base and the color. This is determined by the sequencing chemistry. This protocol uses the MiSeq 4-color chemistry from Illumina (described in more detail in the *in situ* sequencing protocol).

To implement the translation, the logical barcode is translated base by base to the nucleotide sequence using the specified mapping. The translated nucleotide sequence is placed 3' to the RCA primer binding site on the padlock probe. Using the standard parameters, this translation process is equivalent to a direct replacement of the logical bases {0, 1, 2, 3} with the nucleotide bases {'A', 'C', 'G', 'T'}.

As an example, the logical barcode 1020301 is translated into CAGATAC on the padlock probe, corresponding to GTATCTG on the RCA product.

12.2 Technical Note: Mapping Logical Barcodes to Nucleotide Sequences -- SOLiD Chemistry

In this step, the technical details of the mapping between logical barcodes and nucleotide sequences for readout with the SOLiD chemistry are discussed. The key difference between the Illumina sequencing chemistry and the SOLiD sequencing chemistry is that SOLiD interrogates dibase pairs of nucleotides. Because these dibases can also be interrogated in a partly flexible order, additional constraints on the barcode design are necessary to generate the SOLiD barcodes.

Barcode sequence amplification: the SOLiD barcode sequence is located 5' to the RCA primer binding site on the padlock probe (**Fig. 1A**). After rolling circle amplification, the reverse complement of the original SOLiD barcode sequence is located 3' to the sequencing primer binding site. Note that the sequencing primer binding site is the reverse complement of the RCA primer binding site.

Barcode sequencing with SOLiD chemistry: to interrogate different dibase positions, SOLiD sequencing uses multiple offset sequencing primers. The sequencing primers are called SeqN to SeqN-4, where the number indicates the number of bases that the primer is recessed from the start of the SOLiD barcode.

SOLiD sequencing proceeds as follows. (1) One of these sequencing primers is hybridized to the amplicons. (2) The first ligation of SOLiD sequencing oligos is performed. The ligation mix contains all SOLiD sequencing oligos. The oligo that is ligated to the growing sequencing product is the one that correctly interrogates the dibase on the template strand that is adjacent to the growing sequencing product. (3) As the the SOLiD oligos are fluorophore-labeled, imaging is performed. Each amplicon is detected in one (of four) colors. (4) A cleavage step is then performed, which cleaves the last three nucleotides off

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

the 5' end of the SOLiD oligo, including the linkage to the fluorophore. The cleavage also reveals a 5' phosphate on the sequencing product, which is used for the next ligation. (5) Further cycles of ligation-imaging-cleavage are performed. (6) At the end of sequencing on one primer, the sequencing product is stripped, and a sequencing primer with a different offset is hybridized to the amplicons. This process is repeated for each sequencing primer. The net result of this process is that information about the dibases in the sample is read out. The order of this readout is in a different order from the order of the dibases within the barcode sequence. Each round of imaging (i.e. detected dibase) corresponds to one round from the barcode.

An example of the sequencing process is shown schematically in **Fig. 1E**. This figure shows the hybridization of the SeqN-1 primer (as the end of the primer is recessed by one base relative to the start of the barcode sequence), followed by two ligations on the primer. In the first ligation, the template bases are 5' A, C 3', matched by 5' G, T 3' on the seq oligo. The oligo is then cleaved, and the second ligation is performed. On the second ligation, the template bases are 5' G, C 3', matched by 5' G, C 3' on the seq oligo. This process will then be repeated on the other sequencing primers: SeqN, and SeqN-2 to SeqN-4. Note that for SeqN-2 to SeqN-4, the first ligation does not need to be imaged, since this ligation occurs within the primer sequence and is detecting known dibases in the template.

The interrogated dibases of the sequencing process are summarized in **Table 3**. For targeted ExSeq under the constraints outlined in the abstract (for the scheme in **Fig. 1C** and **Fig. 1G**), we do not need to consider ligations beyond ligation 2.

A	B	C
	Bases interrogated by ligation 1	**Bases interrogated by ligation 2**
SeqN	1,2	6,7
SeqN-1	0,1	5,6
SeqN-2	not imaged	4, 5
SeqN-3	not imaged	3, 4
SeqN-4	not imaged	2, 3

Table 3. Summary of barcode bases interrogated using SOLiD sequencing on SeqN to SeqN-4 sequencing primers.

Note that the sequencing on different primers is an independent process, and that primers can be sequenced in any order.

Barcode construction: the translation of a logical barcode to a nucleotide sequence depends on the order of how interrogated dibases correspond to bases in the barcode. We refer to this order as the "barcode specification". Once the specification is set, the nucleotide sequences can be determined. **Fig. 1B** shows the mapping between logical barcodes, colors, and template

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

dibases. The specific barcode architecture used in this protocol (see Step 4) is shown in **Fig. 1C**, and is implemented as shown in **Fig. 1G** (note that the barcode colors are shared across these figures).

Setting the barcode specification: the barcode specification defines which dibases (and associated sequencing primers/ligation positions) correspond to which positions of the logical barcode. In this protocol, the barcode structure in **Fig. 1C** is implemented using the specification shown in **Fig. 1G**.

Explicitly, the barcode specification states: BC1 (SeqN-1, Ligation 1); BC2 (SeqN-1, Ligation 2); BC3 (SeqN, Ligation 1); BC4 (SeqN, Ligation 2); BC5 (SeqN-2, Ligation 2); BC6 (SeqN-3, Ligation 2); BC7 (SeqN-4, Ligation 2). BCx are the positions in the logical barcode, and the sequencing specifications in parentheses define the dibase corresponding to that logical barcode value. This is shown graphically in **Fig. 1C** and **Fig. 1G**. Corresponding logical barcode positions in **Fig. 1C** and sequencing rounds in **Fig. 1G** are shown in the same color. The dibases interrogated are colored with black circles. "Silent" ligations that are performed but not imaged are shown with dashed lines.

The formal barcode specification is defined as shown in **Table 4**. The rows and columns define the sequencing primer and ligation number, and the value is the logical barcode position, i.e. SeqN-2, Ligation 2 is logical barcode position 5.

A	B	C
	Ligation 1	**Ligation 2**
SeqN	3	4
SeqN-1	1	2
SeqN-2	not detected	5
SeqN-3	not detected	6
SeqN-4	not detected	7

Table 4. Formal specification of sequencing scheme shown in **Fig. 1G**.

Note that this barcode specification determines the total length of the oligonucleotide barcode, since the highest barcode position in **Table 3** for any of the ligations specified in **Table 4** is 7 (corresponding to the 6,7 dibase for SeqN, Ligation 2). Thus, the total oligonucleotide length of the SOLiD barcode is 7 bases.

The advantage of this scheme is that it allows the embedded four round, HD2 barcode to be read out in four rounds of SOLiD sequencing, without requiring the use of any silent ligations, and only requiring the use of two sequencing primers. In addition, the sequencing of the full barcode can be performed linearly; by using the primers SeqN-1, SeqN, SeqN-2, SeqN-3, and SeqN-4 in that order, the barcode can be read out from BC1 to BC7 in order, without requiring any data rounds to be shuffled.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

An alternative barcode specification would be to directly interrogate the barcode sequence from left to right, where each BCx corresponds to the x-1,x nucleotide dibase. Explicitly: BC1 is the 0,1 dibase (SeqN-1, Ligation 1); BC2 is the 1,2 dibase (SeqN, Ligation 1); BC3 is the 2,3 dibase (SeqN-4, Ligation 2); BC4 is the 3,4 dibase (SeqN-3, Ligation 2); BC5 is the 4,5 dibase (SeqN-2, Ligation 2); BC6 is the 5,6 dibase (SeqN-1, Ligation 2); and BC7 is the 6,7 dibase (SeqN, Ligation 2). This barcode structure is conceptually simpler, but requires additional silent ligations to read out the embedded length 4, HD2 embedded barcode library, and requires additional data shuffling (since, for example, the two ligations on SeqN will be performed sequentially, but correspond to logical barcode BC2 and BC7).

The formal specification for this scheme is shown in **Table 5**.

A	B	C
	Ligation 1	**Ligation 2**
SeqN	2	7
SeqN-1	1	6
SeqN-2	not detected	5
SeqN-3	not detected	4
SeqN-4	not detected	3

Table 5. Formal specification of alternative sequential sequencing scheme.

Translating the order to nucleotides: the barcode specification and the SOLiD logical barcode to nucleotide mapping (**Fig. 1B**) are used to construct the nucleotide sequence. There are three columns in **Fig. 1B** that are relevant for SOLiD sequencing: the logical barcode, the the color, and the table showing the dibase/color relationship. The first mapping is between the logical barcode values and the colors. This relates the logical barcode to the expected color during imaging, where blue through red correspond to the four SOLiD sequencing dyes from lowest to highest excitation wavelength.

The second mapping is between the template bases and colors, shown in the dibase nucleotide to color table. The first base of the dibase (from 5' to 3' on the template) determines the row, and the second base determines the column. For example, the template dibase 5' A, C 3' (from **Fig. 1E**) is row 1, column 2, which is green, corresponding to a logical barcode value of 1.

The challenge with SOLiD barcode construction is that each logical barcode value could correspond to multiple dibase nucleotides. For example, a logical barcode value of 2 (orange) could correspond to template dibases of: {AG, CT, GA, TC}. To disambiguate between these possibilities, information about the adjacent dibases (that share one nucleotide with the current dibase) is needed. If every nucleotide in the barcode is included in two dibases (with the exception of the terminal barcode nucleotide (i.e. position 7 here), then the

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

barcode is fully specified, and can be constructed by considering each dibase sequentially. The construction begins by considering the 0,1 dibase, which contains a base from the sequencing primer in position 0, which fixes the value of the base in position 1. Consideration of the 1,2 dibase now has a fixed value in position 1, which specifies the value of the base in position 2. This process is repeated to construct the complete barcode sequence. Again, the key coverage constraint is that every base in the barcode except the final base is included in two dibases.

Modifiability: this section focuses on the length 7, HD3 barcode library with embedded length 4, HD2 library (as shown in **Fig. 1C** and **Fig. 1G**), but the approach is flexible for different barcode specifications, as long as the barcode coverage constraint is satisfied.

To summarize, a logical barcode can be translated to a SOLiD nucleotide barcode if a SOLiD barcoding specification is defined, and if each nucleotide in the barcode (except the final base) is included in two dibases in the barcoding specification.

13 Running Probe Assembly Script

Targeted_ExSeq_ProbeGeneration_Assembly.m assembles the final probe sequences and formats them in several output formats.

Briefly, the assembly scripts loops through each gene, generates SOLiD and Illumina nucleotide sequences for that gene's logical barcode, then assembles the final probe sequences, including 5' phosphorylation. The data is output in three spreadsheets for export, further explained in Step 14.

Prerequisites: the script requires a genelist spreadsheet and a directory with homology regions (from the first script).

Parameters are specified in several sections in the script.

Section 1 covers the standard set of simple parameters -- the required parameters for reading/writing out files and basic metadata, along with optional parameters for changing aspects of the padlock probes and export formats.

Section 2 covers advanced parameters that define the mapping between logical barcodes and the Illumina and SOLiD sequencing chemistries. Most users will not need to change these parameters.

Section 3 covers parameters that determine the SOLiD barcoding specification. Most users will not need to change these parameters, as the scheme in **Fig. 1G** is implemented as shown. For alternative barcoding schemes, this section will need to be modified.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

13.1 Parameters Requiring User Input -- Section 1

There are nine parameters that require user input.

1. species, a string describing the species that the probes are designed for, i.e. 'Human' (part of the metadata for probe generation).
2. metadata, a string describing the nature of the probe set, i.e. 'Cancer' (part of the metadata).
3. date_of_generation, a string describing when the probes were generated (part of the metadata).
4. genelist_spreadsheet, the location/filename of the genelist spreadsheet.
5. homology_regions_directory, the directory where homology regions are saved (i.e. output from previous script).
6. output_directory, the directory where output sequences (in spreadsheet form) will be saved.
7. all_probes_output_filename, the filename where the spreadsheet containing a list of all probes will be saved.
8. probes_output_by_sheets_filename, the filename where the spreadsheet containing all probes formatted into sheets for upload to IDT will be saved.
9. plate_map_for_pooling_filename, the filename where the spreadsheet containing plate maps to help with pooling probes together will be saved.

13.2 Optional Parameters to Set -- Section 1

In section 1, there are five optional parameters to modify. Most users will not need to modify these parameters. All of these parameters are saved in the params struct.

1. RCA_PRIMER_BINDING_SITE= 'TCTCGGGAACGCTGAAGACGGC', the sequence on the padlock probe backbone that the RCA primer hybridizes to. The default sequence is a modified version of the universal primer binding sequence from FISSEQ to increase its Tm.
2. SPACER_SEQUENCE = 'AAA', the spacer between each homology region arm and the SOLiD/Illumina barcode sequences. The length of this spacer is set so that the total length of the padlock probe is equal to 2*n+10, where n is the length of the homology region. Under the standard design, n = 32 nt, so 2*n+10 = 74 nt.
3. MAX_NUMBER_OF_PROBES = 16, the maximum number of padlock probes to generate per gene. A constraint for exporting the probes is that PLATE_NUM_ROWS must be a divisor of MAX_NUMBER_OF_PROBES.
4. PLATE_NUM_ROWS = 8, standard geometry for 96-well plate, used for exporting probes.
5. PLATE_NUM_COLS = 12, standard geometry for 96-well plate, used for exporting probes.

13.3 Optional Advanced Sequencing Chemistry Parameters to Set --

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

Section 2

In section 2, there are four parameters that could optionally be modified to change the mapping between logical bases and nucleotide sequences. The default values match those shown in **Fig. 1B**, and can be changed if desired.

1. ILLUMINA_MAP_LOGICAL = {'0', '1', '2', '3'}, the acceptable logical barcode values to be translated to nucleotides for the Illumina chemistry.
2. ILLUMINA_MAP_TEMPLATE = {'T', 'G', 'C', 'A'}, the corresponding nucleotides for the logical barcodes on the sequencing template (amplicon).
3. SOLID_NUCLEOTIDE_KEYS = {'A'; 'C'; 'G'; 'T'}, the nucleotides defining the row/columns of the SOLID_DIBASE_MATRIX.
4. SOLID_DIBASE_MATRIX = [[0, 1, 2, 3]; [1, 0, 3, 2]; [2, 3, 0, 1]; [3, 2, 1, 0]], the standard SOLiD sequencing matrix mapping between nucleotides and dibases.

The key computed parameters stored in the params struct are ILLUMINA_MAP, SOLID_DIBASE_ENCODING_SPECIFICATION, and SOLID_LOGICAL2DIBASE.

13.4 SOLiD Barcoding Specification Parameters -- Section 3

In section 3, the SOLiD barcode specification is set. Optional parameters can be modified to change the specification.

The default specification is for the standard sequencing scheme (**Table 4**). Most users will not need to change any of these values.

There is one parameter (in the params struct) that needs to be defined before defining the SOLiD barcode specification.

1. MAX_NUM_LIGATIONS = 2, the maximum number of ligations on a single primer required for this barcoding scheme.

The barcode specification can then be defined. The specification is structured in the form of a MATLAB table, with row names 'SeqN', 'SeqN-1', ... , 'SeqN-4' and column names 'Ligation_1', ... 'Ligation_MAX_NUM_LIGATIONS'. The specification is defined by creating the table with all NaN entries, then individually defining the logical barcode position for each sequencing primer and ligation by setting the value of that entry in the table to the logical barcode position, as shown in **Table 4**. Silent ligations remain NaN values.

Two barcoding specifications are defined in this section: (1) the standard barcode specification (**Fig. 1G** and **Table 4**) as T_barcode_specification_embedded_len4_HD2; and (2) the alternative barcode specification (**Table 5**) as

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

T_barcode_specification_consecutive_bases.

The final two variables select the sequencing scheme used, and provide additional metadata.

1. T_selected_barcode_specification = T_barcode_specification_embedded_len4_HD2, this variable selects the barcode specification used in the probe assembly. The default is to select the standard barcode specification.
2. solid_barcode_descriptor = 'embedded_len4', this character array adds additional metadata describing the SOLiD barcode specification to the full probe name.

More complex changes to the SOLiD barcoding scheme will require significant edits to this section.

14 Script Output

This script outputs three spreadsheets that are saved in the directory specified by output_directory.

The first output spreadsheet (named as defined by all_probes_output_filename) lists all of the generated probes in a single spreadsheet. This spreadsheet has four columns: (1) gene symbol; (2) full probe name (containing a description of the species, metadata, date of generation, gene symbol, barcode, SOLiD barcoding scheme, and probe number); (3) an abbreviated probe name (containing the species, metadata, barcode, and probe number); and (4) padlock probe oligonucleotide sequence. Each gene has MAX_NUMBER_OF_PROBES rows in the spreadsheet; if the script was not able to generate MAX_NUMBER_OF_PROBES probes, empty rows are inserted at the appropriate positions.

The second output spreadsheet (named as defined by probes_output_by_sheets_filename) lists all of the generated probes, formatted into multiple sheets, where each sheet contains the probes in one plate. This spreadsheet can be directly uploaded to IDT through their plate upload interface for ordering. The columns of each sheet are Well Position, Name, and Sequence. The well position specifies the position in the plate for the oligo with the listed sequence.

The third output spreadsheet (named as defined by plate_map_for_pooling_filename) contains multiple sheets, one for each plate. The sheets are maps of the individual plates, where the gene symbol is listed in the corresponding wells containing probes for that gene. This serves as a visual map of the plates during probe pooling in step 16.

Ordering Probes

15 Ordering Probes

Padlock probes can be ordered from any major DNA oligonucleotide synthesis company (i.e.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

IDT, Eurofins Genomics, others). Note that the phosphorylation modification code inserted in the previous section may be different between companies (i.e. /5Phos/ for IDT; [PHO] for Eurofins).

Typical orders are in a 96-well plate format at 25 nmol scale, full-yield, shipped wet in RNAse-free water in V-bottom plates at 200 uM concentration.

The output plates by sheets spreadsheet from Step 14 can be directly used to upload the sequence information to the IDT website. Go to Products & Services -> DNA & RNA -> Custom DNA Oligos -> DNA oligos (Order now) -> Select Plates -> 25 nmol, 96 well plate. Click on upload plates, and upload the spreadsheet. Set the options listed in the previous paragraph, and apply properties to all plates in the order.

When ordering through IDT, the minimum number of occupied wells per 96-well plate is 24 wells. If a plate does not have 24 padlock probes, additional sequences need to be added to other wells to reach that number of utilized wells. These additional sequences do not need to be padlock probes -- they just have to be oligo sequences that meet the minimum sequence length.

We recommend discussing large orders for multiple plates with a representative from the company, which may significantly reduce the cost (especially for the 5' phosphorylation modification).

Pooling Probes

16 Preparing Pooled Probe Library

In this step, probes are pooled together to prepare the pooled probe library, used in the Targeted ExSeq Library Preparation protocol.

Before starting, clean the bench by wiping down surfaces and pipettes with 70% ethanol, followed by RNAseZap.

16.1 Preparation of Plates for Pooling

Thaw 96-well plates of probes at **Room temperature** until completely thawed (typically 15 to 30 minutes).

Briefly spin down plates in centrifuge to at least 100x g to pull fluid to the bottom of the wells.

Carefully remove plates from the centrifuge and proceed to preparing individual probe pools.

16.2 Preparation of Individual Probe Pools

In this step, individual probe pools for each transcript of interest in the library

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

are prepared.

Carefully open the rubber seal for one plate at a time. Avoid any motions that may cause splashes between wells.

Using the exported plate map spreadsheet as a guide (from Step 14), pool the entire volume of all padlock probes for a transcript into a single 2 mL microcentrifuge tube to form the individual probe pool for that transcript. Repeat for all transcripts in the library across all plates. The pooling can be performed with an ordinary pipette.

The per-probe concentration of each padlock probe in the individual gene pool is the probe concentration that the plates were ordered at, divided by the number of probes in the individual gene pool (i.e. for 200 µM plates and 16 probes for the current gene, the per-probe concentration in the individual probe pool is 200 µM/16 = 12.5 µM).

After preparing all the probe pools, vortex the tubes and spin them down on a benchtop centrifuge, then either proceed to preparing the pooled probe library, or freeze the individual probe pools down at **-20 °C** .

Note: there is some variation in probe yields between wells in the plate. These effects are relatively minor (~20% variation) and we assume that they do not affect the downstream results with targeted ExSeq. A more precise approach to pooling the probes would be to use a multichannel pipette to pull the same volume from each well, pool them together in a reservoir, and then transfer to a microcentrifuge tube, but this is significantly more labor intensive when pooling multiple plates of probes.

16.3 Preparation of Pooled Probe Library

When pooling probes together to form the pooled probe library, the concentration of each individual padlock probe is normalized to the same concentration.

To implement this, the volume pooled together from each individual gene pool should be proportional to the number of probes in that pool.

As an example, if there are three transcripts in the library, gene A with 16 probes, gene B with 10 probes, and gene C with 14 probes, 160 µL from the gene A pool, 100 µL from the gene B pool, and 140 µL from the gene C pool are mixed together. If the probes were ordered at 200 µM each probe, the per-probe concentration in the gene A pool is 200/16 µM, in the gene B pool is 200/10 µM, and in the gene C pool is 200/14 µM. By taking an amount proportional to the number of probes in that pool, the variable per-probe concentrations are compensated for, and the final per-probe concentrations

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

are all the same (in this case, 5 µM each; computed by 200/16 µM * 160 µL / (160 + 100 + 140 µL)).

Pool proportional volumes of probes together (scaling up or down in volume) into a 50 mL tube, mix, and aliquot the final pooled probe libraries into microcentrifuge tubes. Typically, (10 µL x # of probes) is a reasonable volume to take from for each individual probe pool.

Store individual probe pools and pooled probe libraries at **-20 °C** .

Example Probe Generation Worked Example

17 Worked Example of Probe Generation on Example Dataset

In this worked example, the user will generate probes for three human genes, AKT1, CDK7, and KRT17, using a pre-prepared genelist and Genbank format files.

The results can be compared to reference results to confirm that the probe generation code is working correctly.

17.1 Environment Setup

Follow Step 1 to set up BLAST+ 2.2.26 and MATLAB/System path variables.

Download Human RefSeq reference, and build BLAST database from humanrefseq.fna. Save BLAST database files in working directory (referred to as working_directory/ below).

Download code from GitHub (https://github.com/anusinha/TargetedExSeqProbeGen) and save to working directory.

Copy example_genbank.gb and example_genelist.xlsx from working_directory/example/example_input/ to working_directory/.

17.2 Generating Homology Regions

Open Targeted_ExSeq_ProbeGeneration_Homology.m.

Set the following variables to the following values in the user input section:
1. genelist_spreadsheet = 'example_genelist.xlsx'
2. genbank_sequences = 'example_genbank.gb'
3. refseq_database = 'humanrefseq.fna'
4. homology_regions_output = 'HomologyRegions'

Do not change any of the optional parameters.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation https://dx.doi.org/10.17504/protocols.io.byrbpv2n

Run the MATLAB script, section by section. Updates will be printed to the console, and the output homology regions will be saved in working_directory/HomologyRegions/.

The output files should match those in working_directory/example/example_output/HomologyRegionsExample/.

If there are slight discrepancies in the homology regions, this may result from slight differences in the current reference human transcriptome relative to the version used when generating the example data.

The expected run time is < 10 minutes.

17.3 Validating Homology Regions

Open Targeted_ExSeq_ProbeGeneration_ValidateHomologyRegions.m.

Set the following variables to the following values in the user input section:
1. genelist_spreadsheet = 'example_genelist.xlsx'
2. refseq_database = 'humanrefseq.fna'
3. homology_regions_input = 'HomologyRegions'
4. output_filename = 'example_probe_validation_summary.xlsx'

Do not change any of the optional parameters.

Run the MATLAB script, section by section. Updates will be printed to the console, and the validation results will be saved in working_directory/example_probe_validation_summary.xlsx.

The output file should match the example output in working_directory/example/example_output/reference_example_probe_validation_summary.xlsx.

The expected run time is < 2 minutes.

17.4 Probe Assembly

Open Targeted_ExSeq_ProbeGeneration_Assembly.m.

Set the following variables to the following values in the Section 1 Required Parameters section:

1. species = 'Human'
2. metadata = 'Example'
3. date_of_generation = '20210101'
4. genelist_spreadsheet = 'example_genelist.xlsx'

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n

5. homology_regions_directory = 'HomologyRegions'
6. output_directory = 'ProbeOutput'
7. all_probes_output_filename = 'OutputAllProbes.xlsx'
8. probes_output_by_sheets_filename = 'OutputProbesBySheets.xlsx'
9. plate_map_for_pooling_filename = 'PlateMapForPooling.xlsx'

Do not change any of the values in Section 1 Optional Parameters, Section 2, or Section 3.

Run the MATLAB script section by section. Updates will be printed to the console, and the output with the assembled padlock probes will be saved in working_directory/ProbeOutput/OutputAllProbes.xlsx, working_directory/ProbeOutput/OutputProbesBySheets.xlsx, and working_directory/ProbeOutput/PlateMapForPooling.xlsx.

These output files should match the example output in working_directory/example/example_output/ProbeOutputExample.

The expected run time is < 5 seconds.

Citation: Anubhav Sinha, Fei Chen, Shahar Alon, Ed Boyden Targeted ExSeq -- Probe Generation
https://dx.doi.org/10.17504/protocols.io.byrbpv2n