OligoMiner provides a rapid, flexible environment for the design of genome-scale oligonucleotide in situ hybridization probes

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters

| Published Version | doi:10.1073/pnas.1714530115 |
| Citable link | http://nrs.harvard.edu/urn-3:HUL.InstRepos:37068165 |
| Terms of Use | This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA |
OligoMiner provides a rapid, flexible environment for the design of genome-scale oligonucleotide in situ hybridization probes

Brian J. Beliveauab,1 Jocelyn Y. Kishab, Guy Nirc, Hiroshi M. Sasakiab, Sinem K. Sakab, Son C. Nguyenc,2 Chao-ting Wu, and Peng Yinab,1

aWyss Institute for Biologically Inspired Engineering, Harvard University, Boston, MA 02115; bDepartment of Systems Biology, Harvard Medical School, Boston, MA 02115; and cDepartment of Genetics, Harvard Medical School, Boston, MA 02115

Edited by R. Scott Hawley, Stowers Institute for Medical Research, Kansas City, MO, and approved January 22, 2018 (received for review August 16, 2017)

Oligonucleotide (oligo)-based FISH has emerged as an important tool for the study of chromosome organization and gene expression and has been empowered by the commercial availability of highly complex pools of oligos. However, a dedicated bioinformatic design utility has yet to be created specifically for the purpose of identifying optimal oligo FISH probe sequences on the genome-wide scale. Here, we introduce OligoMiner, a rapid and robust computational pipeline for the genome-scale design of oligo FISH probes that affords the scientist exact control over the parameters of each probe. Our streamlined method uses standard bioinformatic file formats, allowing users to seamlessly integrate new and existing utilities into the pipeline as desired, and introduces a method for evaluating the specificity of each probe molecule that connects simulated hybridization energetics to rapidly generated sequence alignments using supervised machine learning. We demonstrate the scalability of our approach by performing genome-scale probe discovery in numerous model organism genomes and showcase the performance of the resulting probes with diffraction-limited and single-molecule superresolution imaging of chromosomal and RNA targets. We anticipate that this pipeline will make the FISH probe design process much more accessible and will more broadly facilitate the design of pools of hybridization probes for a variety of applications.

FISH is a powerful single-cell technique that harnesses the specificity afforded by Watson–Crick base pairing to reveal the abundance and positioning of cellular RNA and DNA molecules in fixed samples. Originally introduced as a radioactive in situ hybridization method in the late 1960s (1–3), FISH has undergone a series of optimizations that have improved its detection efficiency and sensitivity (4–7). Many of these refinements have centered on the preparation and labeling of the probe material, which traditionally has been derived from cellular DNA or RNA, and include the introduction of the nick translation method that increases the specific activity of labeling (8, 9) and the development of suppressive hybridization techniques that limit background originating from repetitive sequences contained in many probes (10).

More recently, advances in DNA synthesis technology have afforded researchers the opportunity to construct FISH probes entirely from synthetic oligonucleotides (oligos). Oligo probes offer many potential advantages, as they can be selected to have specific thermodynamic properties, engineered to avoid repetitive sequences, designed against any sequenced genome, and endowed with many different types and densities of labels. Whereas the use of oligo probes was initially restricted to the interrogation of multicopy targets such as repetitive DNA (11–13) and mRNA (14–16) with the use of one to a few dozen oligo probes, the recent development of oligo libraries produced by massively parallel array synthesis (17) has empowered a new generation of FISH technologies able to target single-copy chromosomal regions with highly complex libraries of hundreds to many thousands of oligo probes (18–20).

We have previously introduced Oligopaints, a method for the generation of highly efficient probes for RNA FISH and DNA FISH from libraries composed of dozens to many thousands of unique oligo species (20). In the Oligopaints approach, these libraries are encoded such that each molecule contains a short region of homology (~30–50 bases) to the RNA or DNA target flanked by PCR primers (Fig. L1). Following PCR amplification, ssDNA probes can be generated by a number of approaches, including nicking endonuclease treatment followed by gel extraction (20, 21), in vitro transcription followed by reverse transcription (22–24), and digestion by λ-exonuclease (25). Ultimately, these molecular biological approaches produce pools of ssDNA probes that can be labeled directly through the use of a fluorophore-conjugated primer during the PCR or reverse transcription steps (Fig. L4) or indirectly through the inclusion of

Significance

FISH enables researchers to visualize the subcellular distribution of RNA and DNA molecules in individual cells. The recent development of FISH methods employing probes composed of synthetic DNA oligonucleotides (oligos) allows researchers to tightly control aspects of probe design such as binding energy and genomic specificity. Although oligo FISH probes are central to many recently developed massively multiplexed and super-resolution imaging methods, no dedicated computational utility exists to facilitate the design of such probes on the genome-wide scale. Here, we introduce a streamlined pipeline for the rapid, genome-scale design of oligo FISH probes and validate our approach by using conventional and super-resolution imaging. Our method provides a framework with which to design oligo-based hybridization experiments.


The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

This open access article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

Data deposition: Genome-scale probe mining sets are available at oligominer.net.

1To whom correspondence may be addressed. Email: py@hms.harvard.edu or biveau@wyss.harvard.edu.

2Present address: Department of Genetics, University of Pennsylvania, Philadelphia, PA 19104.

3Present address: Department of Genetics, University of Pennsylvania, Philadelphia, PA 19104.

This article contains supporting information online at www.pnas.orglookup/suppl/doi:10.1073/pnas.1714530115/-/DCSupplemental.

Published online February 20, 2018.

www.pnas.org/cgi/doi/10.1073/pnas.1714530115

PNAS | vol. 115 | no. 10 | E2183–E2192

Applied Biological Sciences
of a binding site for a fluorescently labeled “secondary” oligo that can be added during or after the FISH experiment (25).

A key feature of Oligopaints is their programmability, wherein the genomic and nongenomic sequences that compose each probe oligo can be specified precisely. This fine level of control has enabled several important technical advances in FISH imaging, including the single-molecule superresolution imaging of chromosome structure at nonrepetitive targets (25–27), the development of probes that can distinguish genomically unique regions of homologous chromosomes (25), and the introduction of a method able to label dozens of chromosomal loci (28). The general programmability of oligo FISH probes has also enabled the creation of related methods that use aspects of the Oligopaints approach to enable the highly multiplexed visualization of dozens to >1,000 distinct mRNA species in the same sample (22, 29) and >10,000 mRNA species in vitro (30).

Despite the rapid maturation of new FISH technologies reliant on oligo probes, comparatively little progress has been made in the development of computational tools to facilitate the design of these oligos. Such programs seek to identify optimal probe sequences within a block of input sequence based on thermodynamic properties such as melting temperature ($T_m$) while attempting to identify and exclude probes that are likely to hybridize at sites other than the intended target. Although computational utilities exist to create small numbers of oligo probes against targets such as bacterial rRNA (31, 32) and to design large pools of oligo pairs such as PCR primers (33–36) or padlock probes (37, 38), to our knowledge, no bioinformatic utility has been created for the explicit purpose of designing oligo

---

**Fig. 1.** Implementation of OligoMiner. (A) Schematic overview of the Oligopaints. (B) Schematic overview of the OligoMiner pipeline. (C and D) Schematic overviews of LDA model (C) creation and (D) implementation. (E) Receiver operating characteristic curves for each temperature-specific LDA model. (F) Heat map showing the support-weighted $F_1$ score for each temperature-specific LDA model when tested against validation data simulated at each of the six indicated temperatures. (G) Description of utility scripts also developed as part of OligoMiner.
hybridization probes at the genome-wide scale. Consequently, older utilities such as the microarray design program OligoArray (39) have been repurposed to facilitate probe design. Although OligoArray has produced effective oligo FISH probes (20, 25), it can only provide limited throughput, with large genomes such as those of human and mouse taking 1–2 mo of continuous cluster computing to mine with a single set of parameters (20) and smaller regions still requiring hours of cluster computing to complete. Additionally, OligoArray offers the user a limited amount of control over the probe discovery process, as users interact only with a compiled executable Java Archive file and cannot change the nature or order of steps taken or the values of many of parameters used for thermodynamic calculations and specificity checking.

Here, we introduce OligoMiner, a rapid and flexible genome-scale design environment for oligo hybridization probes. The modular, open-source OligoMiner pipeline is written in Python using Biopython (40) and uses standard bioinformatic file formats at each step in the probe mining process, greatly simplifying probe discovery. Additionally, OligoMiner introduces a method of assessing probe specificity that employs supervised machine learning to predict thermodynamic behavior from genome-scale sequence alignment information. The OligoMiner pipeline can readily be deployed on any sequenced genome and can mine the entirety of the human genome in minutes to hours and smaller <10-Mb regions in mere minutes on a standard desktop or laptop computer, greatly reducing the time and computational resource cost of probe discovery. We also demonstrate the effectiveness of probes produced by our approach with conventional and single-molecule superresolution microscopy.

Results

Identification of Candidate Probes. OligoMiner is a collection of Python scripts that facilitates the design of oligo-based hybridization probes. These scripts are designed to be run in a command-line environment, where they can be integrated into existing workflows with other bioinformatic utilities. OligoMiner can be used on any modern Windows, Macintosh, or Linux system and requires no direct knowledge of programming. To execute the probe-design process, users simply need to prepare a suitable input file and call the relevant scripts in the pipeline. The modular design of the OligoMiner framework allows users to choose the components of the workflow and providing installation and utilization instructions, and example input and output files can be found at oligominer.net. The contents of the readme document are also provided in Dataset S1.

Importantly, OligoMiner allows the user to specify a broad range of parameters, including the alignment method used to check specificity, providing substantially more flexibility than OligoArray (Fig. S1).

The OligoMiner workflow begins with a FASTA-formatted input file (41) containing the genomic sequence to be searched for probes, which can be masked by a program such as RepeatMasker (42) to exclude regions containing repetitive elements. This input file is first passed to the blockParse script (Fig. 1B and Fig. S2), which screens for prohibited sequences such as homopolymeric runs and “N” bases and allows users to specify allowable ranges of probe length, percent G+C content (GC%), and adjusted Tm calculated by using nearest-neighbor thermodynamics (43). Candidate probe sequences passing all checks are outputted in FASTQ format (44) to facilitate input into next-generation sequencing (NGS) alignment programs such as Bowtie/Bowtie2 (45, 46) and BWA (47), which can be used to assess off-target potential. Importantly, these NGS alignment programs are optimized for the extremely rapid alignment of millions of short sequences to a reference genome in parallel, thereby allowing the specificity-check step of the pipeline to proceed much more quickly than approaches like OligoArray that use BLAST (48) in serial.

Predicting Probe Specificity. Ultrafast alignment programs can provide a wealth of information about the relatedness of a given input sequence to other sequences present in a genome assembly. OligoMiner allows users to evaluate probe specificity by using two distinct approaches, in either case using the script outputClean to process the Sequence Alignment/Map (SAM) file (40) querying bowtie2, the alignment program and outputting to bedGraph for visualization in genome browsers and are fully compatible with our previously described tools that facilitate the design and ordering of Oligopaint probe libraries (20) (genetics.med.harvard.edu/oligopaints) and utilities such as BEDTools (51) (Fig. 1B and Fig. S3). The first approach, termed Unique Mode (UM), uses the number of reported alignments to differentiate between candidate probes predicted to only have one genomic target from those with multiple potential binding sites; candidates with more than one reported alignment or that fail to align are filtered, whereas candidate probes that align uniquely are passed to the output. Unique Mode thereby enables users to experiment with different groups of alignment parameters to find an optimal combination for a given application.

Ideally, the thermodynamics of hybridization between a candidate probe and potential off-target sites would be modeled in silico and employed as a means of identifying probe oligos likely to only find their intended targets in a given set of reaction conditions. Although powerful utilities such as NUPACK (52–54) are capable of performing such simulations, the limited throughput of these programs renders a direct thermodynamic approach impractical for genome-scale probe design. However, we hypothesized that features in rapidly calculated data such as alignment scores may be predictive of thermodynamic behavior and could therefore serve as a proxy for the information that would be produced by thermodynamic simulations. Inspired by this idea, we first selected 800 “probe” sequences identified by blockParse in the human hg38 assembly that represented three commonly used probe length ranges (26–32, 35–41, 40–46 nt; Methods). To simulate the types of binding sites that these “probes” might encounter in situ during a FISH experiment in a complex genome, we next generated 406,014 variant versions of the “probe” sequences in silico that each contained one or more “mutations,” insertions, deletions, or large translocations. In combination with the 800 “probe” sequences, a pool of 406,814 “target sites” (Methods and Fig. 1C). We then aligned each “probe” to its corresponding “target sites” in pairwise alignments by using Bowtie2 with ultrasensitive settings (Methods), generating a set of 406,814 alignment scores (Fig. 1C). In parallel, we also computed the probability of a duplex forming between each “probe” and each of its corresponding “target sites” in FISH conditions (2x SSC, 50% formamide at 32, 37, 42, 47, 52, or 57 °C) in pairwise test tube simulations by using NUPACK (Methods and Fig. 1C).

To connect our alignment scores and duplexing probabilities, we next performed supervised machine learning by using linear discriminant analysis (LDA) on 60% of the combined datasets with scikit-learn (55). Specifically, we built six temperature-specific LDA models that predict whether the duplexing probability of a “probe”−“target site” pair will be above a threshold level of 0.2 (i.e., less than fivefold weaker than a fully paired duplex) given the length and GC% of the “probe” sequence and the score of the alignment of the two sequences (Methods and Fig. 1D). We tested these LDA models on the remaining 40% of the data and found that all six performed exceptionally well, with each producing areas under receiver operating characteristic curves of ≥0.97 (Fig. 1E) and support-weighted F1 scores ≥0.92 (Fig. 1F and Fig. S4). Notably, all six models also performed strongly when tested against data simulated at hybridization temperatures 5 °C higher or lower than the training temperature (support-weighted F1 score range, 0.79–0.92; mean, 0.86; Fig. 1F).

Beliveau et al.
indicating that the models are all capable of predicting duplexing behavior over a relatively broad range of reaction conditions. Collectively, our data argue that the LDA model identifies potentially problematic “probe”–“target site” interactions (i.e., those with a probability of duplexing >0.2) effectively as well as the much slower thermodynamic simulations. We have integrated the six LDA models into outputClean to create the second specificity evaluating approach, “LDA Mode” (LDM); candidate probes are first aligned to the reference genome of interest by using the same Bowtie2 scoring settings used to construct the LDA models (Methods), and the resulting SAM file is processed by a selected temperature-specific LDA model such that candidate probes predicted to have more than one thermodynamically relevant target site (i.e., probability of duplexing >0.2) are filtered (Fig. 1B and Fig. S3).

Postprocessing Functionalities. We have written a series of utility scripts to augment the core OligoMiner pipeline (Fig. 1G). These utility scripts accept and return BED files, making them compatible with output files created by outputClean (Fig. 1B) and files created by the previous Oligopaint probe discovery method (20) and adding additional functionalities. For instance, kmerFilter enables the user to perform another layer of specificity checking by calling Jellyfish (56) to screen probe sequences for the presence of high-abundance k-mers (e.g., 16mers or 18mers) that may be missed by alignment programs because of their short lengths and could lead to off-target binding (57, 58). Users can also identify and filter probe sequences predicted to adopt secondary structures in a given set of experimental conditions by using structureCheck, which depends on NUPACK. Several additional tools facilitate the processing of probe files for specific applications, including the conversion of probe sequences to their reverse complements by probeRC for strand-specific DNA or RNA FISH and the collapsing of overlapping probes by bedChainer for the design of high-density probe sets. Finally, we have created additional modularity with pair of scripts, “fastqToBed” and “bedToFasta,” that allow users to convert between the BED and FASTQ format files.

Rapid Genome-Scale Probe Discovery. To assess the scalability of OligoMiner, we performed genome-wide probe discovery in the human hg38 genome assembly. We first developed three sets of input parameters spanning a range of commonly used probe lengths and experimental conditions: a “coverage” set designed to maximize the number of probes discovered (26–32 nt length, 37 °C hybridization), a “stringent” set designed to maximize probe-binding affinity and thereby permit stringent hybridization and washing conditions (40–46 nt, 47 °C hybridization), and a “balance” set that seeks to compromise between coverage and binding affinity (35–41 nt, 42 °C hybridization; Fig. 2A). We next deployed OligoMiner by using these parameter settings in UM and LDM, in both cases using Bowtie2 for the alignment step and LDM, in both cases using Bowtie2 for the alignment step. We also included the optional kmerFilter specificity check (Methods). Excitingly, both approaches were able to mine the entire hg38 assembly very rapidly by using all three parameter settings (Fig. S5). We next aligned probe sequences by bedChainer for the design of high-density probe sets. Finally, we have developed additional modularity with pair of scripts, “fastqToBed” and “bedToFasta,” that allow users to convert between the BED and FASTQ format files.

Fig. 2. Genome-scale probe discovery with OligoMiner. (A) Description of three parameter sets used for genome-scale mining runs. (B–E) Box plots displaying overall mining times and rates for UM (B and C) and LDM (D and E). Each chromosome was run separately and reported, resulting in 24 data points per parameter setting and a total of 72 data points per plot. The mean rate or time for all 72 data points is displayed beneath each box plot. (F and G) Swarm plots displaying changes in probe density (i.e., probes per kilobase) that occurred over the course of the pipeline in UM (F) and LDM (G). bP, blockParse; kf, kmerFilter; oC, outputClean. (H) Swarm plot displaying probe densities in the C. elegans (ce11), D. melanogaster (dm6), zebrafish (danRer10), human (hg38), mouse (mm10), and A. thaliana (tair10) genome assemblies after whole-genome mining using LDM and kmerFilter.

~50–100-fold increase over 10–100-kb intervals and a ~800-fold increase over megabase-scale intervals (Fig. S5). The modularity of OligoMiner allows users to monitor how the parameters chosen at each step in the probe discovery process affect the final number of output probes. We have used this capability to examine changes in probe density (e.g., probes per kilobase) that occurred during the genome-wide probe discovery runs in hg38. As expected, blockParse discovered the highest density of candidate probes by using the “coverage” (c) settings, followed by “balance” (b) and “stringent” (s) (c: 8.5; b: 5.7; s: 3.4 probes per kilobase; Fig. 2F and G). However, we observed striking differences following outputClean depending on the mode used, with UM preserving the same order (c: 7.3; b: 4.7; s: 2.7 probes per kilobase) but the density of the “coverage” oligos plummeting in LDM (c: 2.6; b: 5.0; s: 3.0 probes per kilobase; Fig. 2F and G and Dataset S2). We also observed large relative decreases in the density of “coverage” oligos following the application of kmerFilter, but only a modest reduction with the other sets (UM, c: 3.1; b: 4.6; s: 2.6 probes per kilobase; LDM, c: 1.6; b: 4.8; s: 2.9 probes per kilobase; Fig. 2F and G); this effect is likely a result of the use of 16mer dictionary with “coverage” sets but an 18mer dictionary with the “balance” and “stringent” sets (Fig. 2A), a choice informed by differences in k-mer binding.
Fig. 3. OligoMiner enables highly efficient FISH. (A and B) Representative single-channel minimum-maximum (min-max) contrasted image (Left) and two-color image with manual contrast adjustment (Right) (A) and signal number quantification (B) of 3D FISH experiment performed with a probe set consisting of 4,776 UM oligos targeting 817 kb at Xq28 in human XX 2N WI-38 fibroblasts. (C and D) Representative single-channel min-max contrasted image (C, Left) and two-color contrast-adjusted (C, Right) and signal number quantification (D) of 3D FISH experiment performed with a probe set consisting of 3,678 LDM oligos targeting 1,035 kb at 19p13.2 in human XY 2N PGP-1 fibroblasts. (E) Quantification of background-subtracted SNR for the Xq28 and 19p13.2 probes. (F) Three-color 3D FISH experiment performed using ATTO 488-labeled “X.1” (green), ATTO 565-labeled “X.2” (magenta), and Alexa Fluor 647-labeled “X.3” UM probe sets targeting adjacent regions on Xq28 in WI-38 fibroblasts. (G and H) Two-color metaphase FISH experiment performed using ATTO 488-labeled “X.1” (green) and Alexa Fluor 647-labeled “X.2” (magenta) UM probe sets targeting adjacent regions on Xq28 on XX 46N (G) and XY 46N (H) chromosome spreads. (I and J) Two-color metaphase FISH experiment performed using Alexa Fluor 647-labeled “19.1” (green) and Cy3B-labeled “19.2” (magenta) LDM probe sets targeting adjacent regions on 19p13.2 on XX 46N (I) and XY 46N (J) chromosome spreads. All images in are maximum-intensity projections in Z. DNA is stained with DAPI (blue) in multichannel images. In G–J, the multicolor images of the full spread and single-channel images (Inset) are min-max contrasted and the multichannel images (Inset) have manual contrast adjustments. (Scale bars: 10 μm; G–J, Inset, 1 μm.) For each image, the minimum and maximum pixel intensity value used to set the display scale is indicated in the lower left.
affinities at the different simulated hybridization temperatures (Fig. S6 and Dataset S2).

Collectively, our genome-scale hg38 probe sets are similar in probe density to previous sets designed with OligoArray (20, 25), and a direct comparison using a set of 3-Mb intervals revealed a high degree of concordance between the probes discovered by the two methods (80–96% of probes shared, $R^2$ values of probes per kilobase of 0.86–0.97, $n = 3$; Fig. S5). Additionally, our results suggest that, when taking the thermodynamics of hybridization into account, longer oligo probes that can support higher hybridization temperatures can effectively provide higher probe densities, as observed with the UM and LDM “balance” sets (Fig. 2 F and G). Intriguingly, this phenomenon appears to depend on genome size and complexity; the same ordering of the three parameter sets was also observed in whole-genome probe discovery performed using LDM and kmerFilter in the mouse...
mm10 and zebrafish danRer10 assemblies, but the “coverage” set provided the highest densities in the smaller Drosophila melanogaster dm6, Caenorhabditis elegans ce11, and Arabidopsis thaliana tair10 assemblies (Fig. 2F and Dataset S2). The resulting probes discovered by these genome-scale probe discovery runs and additional LDM + kmerFilter whole-genome runs in the c66, dm3, hg19, and mm9 assemblies are available on the Oligopaints Web site (genetics.med.harvard.edu/oligopaints).

OligoMiner Enables Conventional and Superresolution Imaging. To test the efficacy of oligo probes designed with OligoMiner in situ, we first performed 3D FISH (59, 60) in XX 2N WI-38 human fetal lung fibroblasts with a set of 4,776 40–45mer Oligopaint probes designed using UM without kmerFilter targeting 817 kb at Xq28 (Table S1). In line with previous Oligopaint experiments using probes designed by OligoArray (20, 25), we observed highly efficient staining, with 100% of nuclei displaying at least one FISH signal and 88.5% of nuclei displaying two signals (n = 130; Fig. 3 A and B). We observed similarly efficient staining after performing 3D FISH in XY 2N PGP-1 fibroblasts with a set of 3,678 35–41mer Oligopaint probes designed using LDM with kmerFilter targeting 1,035 kb at 19p13.12 (100% with >1 signal, 76.9% with two signals, n = 143; Fig. 3 C and D and Table S1), illustrating the high labeling efficiency of probes produced by the UM and LDM approaches. Additionally, automated image analysis (61) revealed excellent signal/noise ratios (SNRs) for both probes (mean SNR, 12.3, n = 261 signals for Xq28; mean SNR, 9.2, n = 331 signals for 19p13.2; Fig. 3E), demonstrating the robustness of probes discovered with both modes. We also validated our ability to design custom hybridization patterns by performing 3D FISH with two additional sets of 40–45mer Oligopaint probes designed using UM without kmerFilter targeting Xq28 in WI-38 cells, which led to the expected three-color colocalization pattern in situ (Fig. 3F).

Finally, we highlighted the specificity of our probes by performing two-color FISH on female and male metaphase spread chromosomes using two sets of Oligopaint probes targeting adjacent regions at Xq28 or 19p13.2 (Table S1), in all cases observing the expected number and distribution of signals (Fig. 3 G–J).

To further showcase the performance of oligos designed using OligoMiner in situ, we visualized 3D FISH by using stochastic optical reconstruction microscopy (STORM) (62) and DNA-based points accumulation in nanoscale topography (DNA-Paint) (63): these single-molecule superresolution imaging techniques spatiotemporally isolate the fluorescent emissions of individual molecules and are capable of achieving <20-nm lateral and <50-nm axial resolution, which represent an order of magnitude or more below the diffraction limit (64). Specifically, we performed STORM imaging of Oligopaints (OligoSTORM) (25) of human 19p13.2 with two sets of 35–41mer oligos designed by using LDM with kmerFilter targeting a 1,035-kb region with 3,768 oligos (Fig. 4 A and B and Table S1) or a 20-kb region with 104 oligos (Fig. 4 C and D and Table S1) and, in both cases, were readily able to resolve the nanoscale morphologies of these foci, including features <40 nm (Fig. 4D), values comparable to those obtained by using probes designed by OligoArray (25). We also performed DNA-Paint imaging of Oligopaints (OligoDNA-Paint) (25) to visualize our 817-kb Xq28 probe set (Fig. 4 E and F and Table S1) and a set of 167 35–41mer oligos designed by using LDM with kmerFilter targeting the Xist RNA (65) (Fig. 4 G and H and Table S1), which also enabled us to reveal <40-nm structural features in the superresolved images (Fig. 4 F and H). Taken together, these superresolution experiments demonstrate the OligoMiner oligos can readily enable the single-molecule superresolution imaging of a broad range of target types and sizes.

Discussion

OligoMiner provides a framework for the rapid design of oligo hybridization probes on the genome-wide scale. We have demonstrated the ease and scalability of our pipeline by mining the human hg38 genome assembly with three distinct parameter sets and in two specificity-checking modes, a feat that would have otherwise required many months of cluster computing, and further highlighted the effectiveness of our approach with conventional and single-molecule superresolution imaging. Created by using open-source Python and Biopython and freely available via GitHub (oligominer.net), OligoMiner can readily be run on any standard laptop or desktop computer and can seamlessly use standard bioinformatic file formats, providing users the opportunity to integrate OligoMiner scripts into existing pipelines and readily allowing additional and updated programs to be seamlessly integrated into the workflow. Critically, OligoMiner is capable of discovering the thousands to tens of thousands of oligo probes commonly ordered as pools from commercial suppliers in mere minutes, freeing the researcher to tailor the design of each probe set to the experimental question at hand instead of relying on preexisting collections of probe sequences obtained from previous probe mining runs or online databases (20). We expect the dramatic increase in speed and flexibility provided by OligoMiner will enable a much broader collection of research groups to use oligo FISH probes, including those working on developing new imaging technologies and in model organism systems not currently supported by existing probe collections. Moreover, we anticipate that OligoMiner could be employed more broadly to design hybridization probes for a wide range of experimental assays beyond in situ hybridization.

Methods

Genome Sequences. The hg19, hg38, mm9, mm10, c66, ce11, danRer10, dm3, and dm6 genome assemblies were downloaded with and without repeat masking from genome.ucsc.edu. The tair10 assembly was downloaded from www.arabidopsis.org. To generate a repeat-masked version of tair10, transposable element locations identified by TASSR (66) were converted to BED format and used as a guide for masking by pyfaidx (67).

Pipeline Construction and Implementation. OligoMiner is written for Python 2.7 and depends on Biopython (40) and scikit-learn 0.17+ (55). Additional optional dependencies include Jewelfish 2.0+ (56) for k-mer screening and NUPACK 3.0 (52–54) for secondary structure analysis. To generate data for this study, scripts were executed locally in an OS X Anaconda Python 2.7 environment (Continuum Analytics) created with the command “conda create --name probeMining biopython scikit-learn” or in a CentOS Linux environment on the Orchestra High Performance Compute Cluster at Harvard Medical School.

LDA Model Construction. Two sets of “probe” and “target site” sequences were used for the LDA model construction. For the first, all possible k-mers ≥8 were generated from 500 40–46mer sequences from hg38 chrX that were identified as candidate probes by blockParse, resulting in a total pool of 337,514 truncated and full-length sequences. In the second, 100 26–32mer, 100 35–41mer, and 100 40–46mer sequences from hg38 chrY identified as candidate probes by blockParse were used as a starting pool of sequences. A Python script was then used to generate variant sequences containing 1–10 point mutations, 1–3 insertions of 1–6 bases each, or 1–3 deletions of 1–6 bases each, resulting in a total pool of 68,300 parental and variant sequences. These two pools were then combined to create a final pool of 406,814 sequences. To generate Bowtie2 alignment scores for each “probe”–“target site” pairing, the “probe” sequence flanked by 3 “T” bases on both the 5′ and 3′ ends was used to create a Bowtie2 alignment index against which the “target-site” sequence was aligned by using the following settings: “--local -D 20 -R 3 -N 1 -L 10 -i 5 |0.05 --score-min G,1,1 -k 1.” To generate NUPACK doubling probabilities for each pairing at a given temperature, the “complexes” executable was first called and given an input of the reverse complement of the “probe” sequence flanked by 3 “T” bases on both the 5′ and 3′ ends and the “target site” sequence in a two-strand simulation with a maximum complex size of two strands. To account for FISH conditions, the Na+ concentration was set to 390 mM and the input temperature was increased by 31 °C (0.62 × 50) to
account for the presence of 50% formamide. The resulting partition function outputted by “complexes” was then passed to the “concentrations” executable, with each strand being assigned an initial concentration of 1 μM. The percentage of the “prob” oligo contained in the “prob-target” complex was then stored as the duplexing probability. If the probability of duplexing was < 0.2, the pairing was assigned to the “not likely to bind stably”(-1) class. If the probability of duplexing was ≥ 0.2, the pairing was assigned to the “likely to bind stably”(1) class. LDA model building, testing, and validation was performed by using scikit-learn 0.17 (55).

Whole-Genome Probe Discovery. Genome assemblies in FASTA format without repeat masking were used to build Bowtie2 alignment indices and Jellyfish files. Repeat-masked input files were used for probe discovery. The block-Parse script was run with the settings indicated in Fig. 2A and all other values set to their defaults (Fig. 52). Bowtie2 was run with “-s,” “-u,” “-k,” “-t,” “-D,” “-R,” “-N,” “-l,” “-L,” “-i,” “-c,” “-d,” “-s,” “-g,” “-x,” “-m,” “-p,” “-P,” “-s,” “-x,” “-g,” “-t,” “-G,” “-X,” “-M,” “-L,” “-g,” “-x,” “-t,” “-G,” “-X,” “-M,” “-L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,” “t,” “G,” “X,” “M,” “L,” “g,” “x,"---"n"","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","".---"c"","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","",""...
Automated Quantification of FISH Signals. Raw, multichannel .czi Z-stack images were imported into Fiji/ImageJ, in which a macro was used to create maximum-intensity projection in Z-plane images for the DAPI and FISH signal channels. These .png images were imported into CellProfiler 3.0 (61), in which an automated image-analysis pipeline was constructed to identify nuclei and signal pixels in the FISH image overlapping with the nucleus, but not part of the FISH foci (i.e., the nuclear background of the FISH signal), and the baseline background of pixels in the FISH image not overlapping with the FISH foci, nuclei, cell bodies, or other objects of increased intensity such as debris. A parent–child relationship was also established between nuclei and FISH signals. From these data, a background-subtracted SNR was calculated as follows: \( \frac{\text{mean FISH signal pixel intensity} \times \text{mean baseline background pixel intensity}}{\text{mean nuclear background pixel intensity}/(\text{mean nuclear background pixel intensity} - \text{mean baseline background pixel intensity})} \) for each nucleus. The parent–child relationship was used to determine the number of FISH signals in each nucleus. The complete CellProfiler pipeline used, as well as example images for Xq28 and 19p13.2, are available at https://github.com/brianbeliveau/OligoMiner/tree/master/ImageQuantification.

STORM Imaging. STORM imaging was performed on a commercial Nikon N-STORM 3.0 microscope featuring a Perfect Focus System and a motorized total internal reflection fluorescence (TIRF) illuminator at the Nikon Imaging Center located at Harvard Medical School. STORM was performed by using highly inclined and laminated optical sheet illumination (HILO) (70) and with pulsed activation of the 405-nm laser, followed by 647 nm, and then 561 nm. Light was focused through a CFI Apo TIRF 100× oil (N.A. 1.49) objective. The 561-nm laser was used at 2% (out of 50 mW) to image 200-nm orange Fluosphere (F8809; Thermo Fisher), which were used as fiducial markers to facilitate drift correction. The 405-nm laser was used at 5% (out of 20 mW), and the 647-nm laser was used at 100% power (out of 125 mW measured at fiber optic). Emission light was spatially filtered (Chroma ET600/50m for 561 nm; Chroma ET700/75m for 647 nm) and imaged on an EMCCD camera (Andor iXon ×3 DU-897) with 16-μm pixels using a CCD readout bandwidth of 10 kHz at 16 bit, 1 bit preamp gain, and 1.48 e−/pixel.readout noise. Background or low-magnification images were obtained with the center 256 × 256 or 120 × 190 pixels, resulting in an effective pixel size of 160 nm. A total of 6,250 or 12,500 10-ms frames were acquired. Single-molecule localization events were identified by using in-house MATLAB software (71) that calls a 2D fitting algorithm (72). Individual localization events were blured with 2D Gaussian functions whose “sigma” parameter was set according to the global drift-independent localization precision as determined by nearest neighbor-based analysis (NeNA) (73). NeNA values were as follows: 19p13.2 1035 kb × 3 nM of Cy3B-labeled 10mer oligo in 1 PBS solution 125–500 nm NaCl. Gold nanoparticle (40 nm; no. 753637; Sigma-Aldrich) were used as fiducial markers to facilitate drift correction. Single-molecule localization events were identified by using in-house MATLAB software (71) that calls a 2D fitting algorithm (72). Individual localization events were blured with 2D Gaussian functions whose “sigma” parameter was set according to the global drift-independent localization precision as determined by NeNA (73). NeNA values were as follows: Xq28– 5.6 nm sigma, 13.2 nm supported resolution; Xist RNA– 5.1 nm sigma, 12.0 nm supported resolution. One- and two-component Gaussian fits of the line traces presented in Fig. 4A–D were calculated by using the “Gaussian Mixture Model” module in skit-learn (55).

DNA-PAINT Imaging. DNA-PAINT imaging was performed on a commercial Nikon N-STORM 3.0 microscope featuring a Perfect Focus System and a motorized TIRF illuminator. DNA-PAINT was performed using HILIC with 15–30% of a 200-nm, 561-nm laser (Coherent Sapphire) using a CFI Apo TIRF 100× oil (N.A. 1.49) objective at an effective power density of ~0.5–1 kW/cm². The 561-nm laser excitation light was passed through a clean-up filter (Chroma ZET561/10) and directed to the objective by using a multi-band beam splitter (Chroma ZT405/488/561/647/70pc). Emission light was spectrally filtered (Chroma ET600/50m) and imaged on an EMCCD camera (Andor iXon ×3 DU-897) with 16-μm pixels using a CCD readout bandwidth of 10 kHz at 16 bit, 1 bit preamp gain, and 1.49 e−/pixel.readout noise. Background or low-magnification images were obtained with the center 256 × 256 or 120 × 190 pixels, resulting in an effective pixel size of 160 nm. A total of 15,000 100-ms frames were acquired for each image by using 1–3 nM of Cy3B-labeled 10mer oligo in 1 PBS solution 125–500 nm NaCl. Gold nanoparticles (40 nm; no. 753637; Sigma-Aldrich) were used as fiducial markers to facilitate drift correction. Single-molecule localization events were identified by using in-house MATLAB software (71) that calls a 2D fitting algorithm (72). Individual localization events were blured with 2D Gaussian functions whose “sigma” parameter was set according to the global drift-independent localization precision as determined by NeNA (73). NeNA values were as follows: Xq28– 5.6 nm sigma, 13.2 nm supported resolution; Xist RNA– 5.1 nm sigma, 12.0 nm supported resolution. One- and two-component Gaussian fits of the line traces presented in Fig. 4A–D were calculated by using the “Gaussian Mixture Model” module in skit-learn (55).