



(51) International Patent Classification:

C12Q 1/68 (2018.01) C12Q 1/00 (2006.01)

(21) International Application Number:

PCT/US2019/065721

(22) International Filing Date:

11 December 2019 (11.12.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/778,863 12 December 2018 (12.12.2018) US

(71) Applicant: **UNIVERSITY OF WASHINGTON** [US/US]; 4545 Roosevelt Way, Suite 400, Seattle, Washington 98101-4721 (US).

(72) Inventors: **GUNDLACH, Jens H.**; c/o University of Washington, 4545 Roosevelt Way, Suite 400, Seattle, Washington 98105 (US). **NOAKES, Matthew**; c/o University of Washington, 4545 Roosevelt Way, Suite 400, Seattle, Washington 98105 (US). **LASZLO, Andrew**; c/o University of Washington, 4545 Roosevelt Way, Suite 400, Seattle, Washington 98105 (US).

(74) Agent: **SHELDON, David P.**; Christensen O'Connor Johnson Kindness PLLC, 1201 Third Avenue, Suite 3600, Seattle, Washington 98101 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: CURRENT-TO-CURRENT NANOPORE DATA BLAST ALIGNMENT

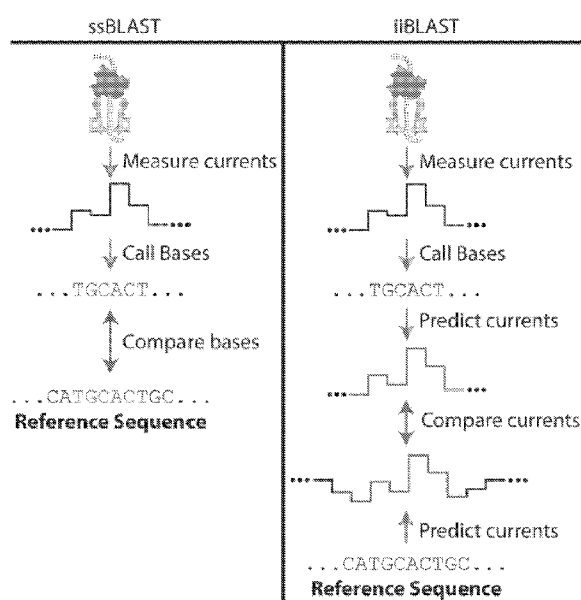


FIG. 2

(57) Abstract: In some embodiments, current-to-current BLAST techniques for aligning electric nanopore signals to signal predictions of a reference database are provided. Operating at the signal, rather than the sequence, level avoids introducing unnecessary errors from base calling and preserves a more faithful representation of the sequenced molecule. The current-to-current BLAST improves on the standard sequence-to-sequence BLAST over a wide range of sequencing accuracies. Similar techniques are disclosed for searching for other analog query signals in reference analog signal databases.



Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

CURRENT-TO-CURRENT NANOPORE DATA BLAST ALIGNMENT

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of Provisional Application No. 62/778863, filed December 12, 2018, the entire disclosure of which is hereby incorporated by
5 reference herein for all purposes.

STATEMENT OF GOVERNMENT LICENSE RIGHTS

[0002] This invention was made with government support under Grant No. R01 HG005115, awarded by the National Institutes of Health. The government has certain rights in the invention.

10 BRIEF SUMMARY

[0003] In some embodiments, a computer-implemented method for aligning genetic sequence reads to a reference genome is provided. A computing system receives a sequence read generated by a sequencing device. The computing system converts the sequence read to a series of predicted analog values that would be generated by the
15 sequencing device for the sequence read. The computing system aligns the series of predicted analog values to a reference genome. A non-transitory computer-readable medium having computer-executable instructions stored thereon that, in response to execution by one or more processors of a computing system, cause the computing system to perform such a method, as well as a computing system that performs such a
20 method, are also provided.

[0004] In some embodiments, a computer-implemented method of finding an analog query pattern in a set of analog reference data is provided. A computing system generates a set of binned query values based on the analog query pattern. The computing system determines a seed value based on the set of binned query values.
25 The computing system searches for the seed value in the set of analog reference data. For each location in the set of analog reference data in which the seed value is found, the computing system extends the seed value into a candidate result. The computing system evaluates the candidate results to determine confidence values that each candidate result represents a true result. A non-transitory computer-readable medium
30 having computer-executable instructions stored thereon that, in response to execution

by one or more processors of a computing system, cause the computing system to perform such a method, as well as a computing system that performs such a method, are also provided.

[0005] In some embodiments, a computer-implemented method for aligning genetic sequence reads to a reference genome is provided. A computing system receives a sequence read generated by a sequencing device, and aligns the sequence read to a reference genome. Aligning the sequence read to the reference genome includes scanning the reference genome for one or more seeds based on the sequence read; and extending the one or more seeds to generate one or more candidate alignments. Each candidate alignment includes an alignment score, and at least one of the scanning of the reference genome and the extending of the one or more seeds is based on a series of predicted analog values associated with the sequence read. A non-transitory computer-readable medium having computer-executable instructions stored thereon that, in response to execution by one or more processors of a computing system, cause the computing system to perform such a method, as well as a computing system that performs such a method, are also provided.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0006] To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced.

[0007] FIG. 1 illustrates a seed-extend approach of BLAST according to various aspects of the present disclosure.

[0008] FIG. 2 is a chart that compares a traditional sequence-to-sequence BLAST alignment technique (ssBLAST) to a non-limiting example embodiment of a current-to-current BLAST alignment technique (iiBLAST) according to various aspects of the present disclosure.

[0009] FIG. 3 includes four charts that illustrates the principle of performing a current-to-current comparison according to various aspects of the present disclosure.

[0010] FIG. 4 is a block diagram that illustrates a non-limiting example embodiment of a system for genomic sequencing and alignment according to various aspects of the present disclosure.

5 [0011] FIG. 5 is a flowchart that illustrates a non-limiting example embodiment of a method of aligning a sequence read to a reference genome according to various aspects of the present disclosure.

[0012] FIG. 6 illustrates an example of a Mealy FSM usable to find seeds according to various aspects of the present disclosure.

10 [0013] FIG. 7 illustrates a bin-free method of seed scanning for the case of continuously valued inputs between 0 and 1 according to various aspects of the present disclosure.

[0014] FIG. 8 is a schematic drawing that illustrates the simple gapped alignment seed extension method according to various aspects of the present disclosure.

15 [0015] FIG. 9 is a schematic drawing that illustrates the windowed extension technique according to various aspects of the present disclosure.

[0016] FIG. 10 and FIG. 11 are a pseudocode listing of a technique for seed extension according to various aspects of the present disclosure.

20 [0017] FIG. 12 is a block diagram that illustrates a non-limiting example embodiment of a computing device appropriate for use as a computing device with embodiments of the present disclosure.

[0018] FIG. 13 illustrates final parameters used in a validation experiment for a non-limiting example embodiment of the present disclosure.

[0019] FIG. 14 is a chart that illustrates the distribution of accuracies for full MinION reads of M13mp18 in the test dataset.

25 [0020] FIG. 15 is a chart that illustrates TPR and FPR variation with respect to an acceptance threshold in order to select a BLAST operational point.

[0021] FIG. 16 includes four charts that illustrate receiver operating characteristics for different read accuracies.

[0022] FIG. 17 is a chart that illustrates a comparison of true positive rate as a function of read accuracy for ssBLAST and iiBLAST techniques.

[0023] FIG. 18 includes two charts that show a length correction for alignment scores according to various aspects of the present disclosure.

5 [0024] FIG. 19 includes two charts that illustrate conversion of alignment scores to p-values according to various aspects of the present disclosure.

[0025] FIG. 20 illustrates a complete accounting of the average per-read time spent during the various phases of the BLAST algorithm in validation experiments of a non-limiting example embodiment of the present disclosure.

10 [0026] FIG. 21 illustrates alignment p-value results from validation experiments of a non-limiting example embodiment of the present disclosure.

DETAILED DESCRIPTION

INTRODUCTION

[0027] Many important clinical and in-field DNA sequencing questions can be
15 answered without conducting whole genome sequencing. Applications including pathogen detection, outbreak tracking, and metagenomic studies are more interested in coarse-grained information about the representation of different species or mutants in a sample, rather than in sequencing entire genomes. The common question asked in these studies is not, "what is the DNA sequence of this organism (or organisms)?" but
20 rather, "what organism (or organisms) are represented by these measured DNA sequences?"

[0028] These sorts of "what's in my pot?" experiments rely on quickly searching for matches of the sequencing reads against a large database of previously-sequenced reference genomes. This type of large-scale sequence-to-sequence comparison
25 problem is computationally difficult. The reference databases within which we want to look for matches can be dauntingly large. The entire database of all known genomes, available from the National Center for Biotechnology Information (NCBI), comprises over 2.9 Tb (nearly 3 trillion bases). Searches usually only use a subset of this vast database, but can still commonly run to several gigabases.

[0029] The computational issues posed by the large reference database sizes are compounded by the typical requirement that a search return inexact matches. Searching for exact sequence-to-sequence matches is computationally simple, even for long sequences. This sort of simple "word search" can run in $O(N)$ time, where N is the length of the sequence within which we are searching for matches (i.e. the reference database). However, given the possibility of mutations relative to the reference genome, and of errors in either the reference or the read, we are often interested in inexact matches.

[0030] The optimal inexact match ("alignment") of one smaller sequence to some subset of a larger sequence can be efficiently found by the Smith-Waterman alignment technique. The Smith-Waterman technique is a special case of the more commonly known Needleman-Wunsch technique. Needleman-Wunsch finds the optimal global alignment between two sequences, aligning all of sequence 1 against all of sequence 2. Smith-Waterman instead finds the optimal local alignment between subsets of the two sequences.

[0031] A full sequence-to-sequence Smith-Waterman alignment requires $O(N_1 * N_2)$ time, where N_1 is the length of sequence 1 and N_2 is the length of sequence 2. As the database of reference genomes becomes large, conducting a Smith-Waterman alignment of the sequencing reads against the entire reference becomes impractical.

[0032] The Basic Local Alignment Search Tool ("BLAST") is a fast alignment technique commonly used to solve these computational difficulties. Unlike the Smith-Waterman technique, BLAST is a heuristic algorithm and does not guarantee that it will find the optimal local alignment. However, it can reliably achieve near-optimal results while vastly reducing computation time relative to Smith-Waterman, making it a powerful tool for searching large sequence databases.

[0033] BLAST achieves its speed advantage through a seed-extend approach, as illustrated in FIG. 1. Given a "query" sequence (i.e. the sequencing read) and a reference database to search, BLAST first looks for seeds: short, exact matches of segments of the query within the database. Seed finding is a fast $O(N)$ word search problem, as described above. The found seeds are then extended using a Smith-

Waterman-type local alignment technique, terminating when the quality of the nascent alignment starts to deteriorate.

[0034] The seed-extend strategy saves time over full alignment by avoiding filling in the alignment matrix at locations unlikely to generate good matches. A Smith-Waterman alignment calculates a length(query) x length(reference) alignment matrix, representing all possible alignments of the two sequences. BLAST instead only fills in the alignment matrix around the exactly matching seeds. This selective calculation introduces some risk that the overall best alignment is missed, but this risk is low assuming the seeds are of a reasonable size. In return for this risk, we achieve a dramatic improvement in run time.

[0035] Several of the advantages of nanopore sequencing make it ideally suited for BLAST-based sequencing applications. Already, nanopore sequencing devices have demonstrated fast sample-to-answer timelines, low input and sample preparation requirements, and portability. These advantages indicate that nanopore sequencing technologies could be uniquely well-suited to these sorts of BLAST-based clinical and in-field sequencing applications where speed, ease-of-use, and portability are often of utmost importance. Indeed, researchers and clinicians have already used nanopore sequencing effectively in pathogen detection, outbreak tracking, species identification, and metagenomic sequencing.

[0036] However, the low single-read sequencing accuracies typical of existing nanopore sequencing devices can hinder BLAST performance. Lower read accuracy means that a read will match less strongly to the correct on-target location in the reference database. Weaker on-target alignments mean that the correct alignment is less distinguishable from the crowd of off-target alignments. Thus, low accuracy reads can increase the rate of both false positives (alignments to incorrect reference locations being called as matches) and false negatives (alignments to the correct reference location not being called as matches).

[0037] Aligning the measured ionic current signal from the nanopore against the predicted signal for the reference sequence using a Smith-Waterman-style alignment can give strong alignments even for low accuracy reads. Using the ionic current-to-sequence model to predict the signal that would be observed for the reference

sequence, then aligning the measured signal to that generated strong read-to-reference alignment, even for reads with low sequencing accuracy. Aligning current-to-current instead of sequence-to-sequence is effective due to nanopore sequencing errors which are caused by incorrect decoding of ambiguous ionic current signals, rather than by
5 fundamental errors in the signal itself.

[0038] These base calling errors arise as many different DNA sequences can generate only slightly different electronic signals. In the case of an ambiguous signal that could have been generated by multiple different underlying sequences, the base caller is forced to make a deterministic choice amongst the different sequence candidates.
10 These deterministic choices in response to ambiguous information can freeze in errors and destroy information that was present in the original signal. By aligning using the ionic current signal in lieu of the called bases, we preserve all the information available and avoid unnecessarily introducing errors into the reads.

SEQUENCE-TO-SEQUENCE BLAST AND CURRENT-TO-CURRENT BLAST

[0039] FIG. 2 is a chart that compares a traditional sequence-to-sequence BLAST alignment technique (ssBLAST) to a non-limiting example embodiment of a current-to-current BLAST alignment technique (iiBLAST) according to various aspects of the present disclosure. As shown, ssBLAST uses the base calls based on the nanopore ionic current signal, then compares the bases against the reference sequence. In the
15 illustrated embodiment, iiBLAST takes the same base calls, then uses them to reconstruct the ionic current signal based on the ionic current-to-sequence model. The ionic currents for the reference sequence are reconstructed in the same way. Finally, the reconstructed ionic currents of the measurement and reference are compared. As will be shown below, the current-to-current comparison of iiBLAST has numerous
20 advantages over the sequence-to-sequence comparison of ssBLAST, particularly for nanopore sequence reads.

[0040] FIG. 3 includes four charts that illustrates the principle of performing a current-to-current comparison according to various aspects of the present disclosure. In Chart A, a hypothetical series of ionic current measurements (the dotted line
25 connecting round ball current values) is shown. These measurements are similar to the predicted currents for both the true DNA sequence (solid line) and an incorrect

DNA sequence (dotted line). While the predicted currents for both the true and incorrect sequences are similar both to each other and to the measured signal, the sequences themselves differ in 6 out of their 10 nucleotides. A base calling algorithm attempting to decode the measured signal could easily read out the incorrect sequence in this instance, making it difficult to trace the read back to its correct location in a reference.

[0041] Chart B illustrates the residuals (measured ionic current versus predicted ionic current) of the data against the true sequence prediction and the incorrect sequence prediction. In both cases, the residuals are small, and the data matches slightly better to the incorrect sequence than the true sequence, thus causing the base caller to call the incorrect sequence. The measured data diverges from the true prediction primarily due to a somewhat low measurement at DNA position 4. This single low measurement in the raw data causes 6 incorrect base calls.

[0042] Chart C illustrates the result of providing the base-called sequence for the example read for use in a BLAST search. If this data was used in a BLAST search, it would yield a poor match to the true sequence, with only 4 out of 10 bases matching (scores are assigned as +1 for match, -1 for mismatch, for a total of -2).

[0043] Chart D illustrates the result of providing the predicted ionic currents for use in a BLAST search. The predicted ionic currents of the called sequence preserve the ambiguity of the measured data. Comparing the currents of the called sequence to those of the true sequence, we see a stronger match. Scores are assigned as 2 minus the z-score between the currents (for a total of +14.0). Though the score is assigned as 2 minus the z-score in the illustrated embodiment, this is only one possible scoring system. In other embodiments other signal-to-signal scoring systems may be used.

[0044] Accordingly, in some embodiments the present disclosure, the principle of current-to-current comparison is extended by adapting the BLAST algorithm to use the measured nanopore ionic currents instead of the called bases. One goal is to produce a fast heuristic sequence alignment method that optimally uses the data present in the nanopore signal. In some embodiments, an ionic current-based BLAST technique is used.

[0045] FIG. 4 is a block diagram that illustrates a non-limiting example embodiment of a system 400 for genomic sequencing and alignment according to various aspects of the present disclosure. As shown, the system 400 includes a measurement system 402 and an alignment determination system 412.

5 **[0046]** The measurement system 402 and the alignment determination system 412 may exchange information using any suitable technique, including but not limited to a wired networking technique (including but not limited to Ethernet, USB, and FireWire), a wireless networking technique (including but not limited to Wi-Fi, WiMAX, Bluetooth, or mobile communication techniques including but not limited to
10 4G, 5G, and LTE), medium transfer techniques (including but not limited to storing information on one device using a removable computer-readable medium of any suitable type including but not limited to an optical disk, a floppy disk, a removable hard disk drive, or a flash drive, and then moving the computer-readable medium to the other device), or any combination of such techniques.

15 **[0047]** The measurement system 402 may be any type of system for generating analog signals that represent nucleotides in a DNA molecule. One non-limiting example embodiment of a measurement system 402 is a nanopore measurement system, of which the MinION system from Oxford Nanopore Technologies is a non-limiting example. Other examples of types of measurement systems include, but are not limited to
20 to, single molecule real time sequencing (SMRT) systems and optical mapping systems. In some embodiments, systems that generate continuously valued analog information other than DNA information, including but not limited to voice/speech recognition systems and soil/rock strata analysis systems, may be used, and the iiBLAST techniques disclosed herein may be adapted to cover such technologies.

25 **[0048]** As shown, the measurement system 402 includes a signal generation device 404 and an analog to digital signal decoding device 410. In some embodiments, the signal generation device 404 includes structure that physically analyzes the sample, such as a DNA molecule, and generates the analog signal based on the physical analysis. A non-limiting example of such structure is a nanopore, for which the analog
30 signal is a measurement of a current associated with the nanopore for a base of the DNA molecule. In some embodiments, the analog to digital signal decoding device 410 includes logic that analyzes the analog signals generated by the signal generation

device 404 to determine a sequence of bases represented by the analog signals. The analog to digital signal decoding device 410 may then provide the sequence of bases to other systems, such as the alignment determination system 412. The measurement system 402 may also provide the raw analog signals to other devices (such as the alignment determination system 412).

[0049] In some embodiments, the alignment determination system 412 includes one or more computing devices configured to provide the functionality described below. Any suitable type of computing devices may be used, including but not limited to desktop computing devices, laptop computing devices, mobile computing devices, tablet computing devices, computing devices that are part of a cloud computing system, or any combinations thereof.

[0050] As shown, the alignment determination system 412 includes at least one processor 416, a computer-readable medium 418, and a reference data store 414. As shown, the computer-readable medium 418 has logic stored thereon that, in response to execution by the at least one processor 416, causes the alignment determination system 412 to provide an analog signal generation engine 406 and a BLAST alignment engine 408. In some embodiments, the analog signal generation engine 406 is configured with logic similar to the logic executed by the analog to digital signal decoding device 410 for converting analog signals into bases, but that operates in reverse (that is, to predict analog signals that would be expected to generate a given series of bases). In some embodiments, the BLAST alignment engine 408 is configured to use the information provided by the other components of the system to perform a BLAST alignment as described in detail below.

[0051] In some embodiments, the reference data store 414 is configured to store reference genome information to which the sample being analyzed is to be aligned. In some embodiments, the reference data store 414 may include information for multiple different genomes. In some embodiments, the reference data store 414 may also be configured to store predicted analog values associated with the reference genome information, so that such values only need to be calculated once.

[0052] As used herein, the term "engine" refers to logic embodied in hardware or software instructions, which can be written in a programming language, such as C,

C++, COBOL, JAVA™, PHP, Perl, HTML, CSS, JavaScript, VBScript, ASPX, Microsoft .NET™, Go, Python, and/or the like. An engine may be compiled into executable programs or written in interpreted programming languages. Software engines may be callable from other engines or from themselves. Generally, the engines
5 described herein refer to logical modules that can be merged with other engines, or can be divided into sub-engines. The engines can be implemented by logic stored in any type of computer-readable medium or computer storage device and be stored on and executed by one or more general purpose computers, thus creating a special purpose computer configured to provide the engine or the functionality thereof. The engines
10 can be implemented by logic programmed into an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), or another hardware device.

[0053] As used herein, the term “data store” refers to any suitable device configured to store data for access by a computing device. One example of a data store is a highly reliable, high-speed relational database management system (DBMS) executing on one
15 or more computing devices and accessible over a high-speed network. Another example of a data store is a key-value store. However, any other suitable storage technique and/or device capable of quickly and reliably providing the stored data in response to queries may be used, and the computing device may be accessible locally instead of over a network, or may be provided as a cloud-based service. A data store
20 may also include data stored in an organized manner on a computer-readable storage medium, such as a hard disk drive, a flash memory, RAM, ROM, or any other type of computer-readable storage medium. One of ordinary skill in the art will recognize that separate data stores described herein may be combined into a single data store, and/or a single data store described herein may be separated into multiple data stores, without
25 departing from the scope of the present disclosure.

[0054] FIG. 5 is a flowchart that illustrates a non-limiting example embodiment of a method of aligning a sequence read to a reference genome according to various aspects of the present disclosure. The method 500 implements a non-limiting example embodiment of the novel iiBLAST technique to align sequence reads. In some
30 embodiments, various changes from the illustrated flowchart, some of which are expressly described below, may be made without departing from the scope of the present disclosure.

[0055] From a start block, the method 500 proceeds to block 502, where an analog signal generation engine 406 of an alignment determination system 412 generates analog values corresponding to a reference genome. At block 504, the analog signal generation engine 406 stores the analog values in a reference data store 414 of the alignment determination system 412. The analog values for the reference genome may represent the current that would be expected to be observed if the sequence of nucleotides in the reference genome was sequenced by the signal generation device 404, and may be based on an inverse of a model used by the analog to digital signal decoding device 410 of the measurement system 402 to determine the bases corresponding with the signals generated by the signal generation device 404.

[0056] At block 506, a signal generation device 404 of a measurement system 402 generates an analog signal for a sample. In some embodiments, the analog signal includes a varying current detected by a nanopore as a molecule of DNA passes through the nanopore. At block 508, an analog to digital signal decoding device 410 of the measurement system 402 determines a sequence read based on the analog signal. In addition to correlating particular currents and/or sequences of currents to the presence of particular nucleotides, in some embodiments, the analog to digital signal decoding device 410 may also perform additional error correction actions that improve the quality of the sequence read in view of common processing errors. At block 510, the analog to digital signal decoding device 410 transmits the sequence read to the alignment determination system 412.

[0057] At block 512, the analog signal generation engine 406 converts the sequence read to a series of predicted analog values. There are several possible choices of which ionic currents to use in the current-to-current comparison. One option is to use the raw ionic current states measured by the nanopore device (in which case block 506, block 508, and block 512 would be unnecessary, and block 510 would instead transmit the analog signal for the sample). Another option is to take the base-called sequence and reconstruct it back into ionic currents using the ionic current-to-sequence model used in base calling (as illustrated in FIG. 5). A variant on this second approach is to directly compare the bases of the base-called sequence and reference, but under the statistical framework of a sequence-to-current generation model. In this implementation, current-generating blocks of the measured sequence (for example, n-base segments or

n-mers) are compared against the same blocks of the reference sequence, and scored according to how similar or different the current generation model expects their respective signals to be. This avoids storing a fully translated “signal” version of the reference database. While the first choice (measured ionic currents) seems the more natural choice, we found results can be improved by using the second choice (reconstructed ionic currents).

[0058] The reasons for this are two-fold. The first and primary reason is that of calibration. The ionic currents observed during various nanopore reads of the same DNA sequence can vary in their overall magnitude (offset from zero) and the relative magnitudes of the different ionic current states (scale). Variation can occur day-to-day, experiment-to-experiment, and even pore-to-pore due to variation in electrode offsets. Prior to base calling, each individual read must be calibrated to the ionic current-to-sequence model used by the base caller. Further complicating matters, the correct calibration for a given read is not necessarily constant over the duration of the read. Particularly for long reads, the calibration required to match the measured ionic currents to the model can drift as the electrode offsets change. The sophisticated base calling software provided by ONT can account for the confounding effects of calibration, and the base-called results of each read represent the optimal decoding of the calibrated ionic current measurements. Thus, by reconstructing the base calls to ionic currents, rather than using the raw measured states, we circumvent the difficulties of calibration and are guaranteed a well-calibrated ionic current signal.

[0059] The second, less significant advantage to using the reconstructed ionic currents is also predicated on corrections made by ONT's base caller. The measured nanopore ionic current signal can be marred by complex error modes. For example, the motor protein used to control DNA progression through the nanopore can take random, non-single-nucleotide steps. Additionally, the change point detection algorithm used to partition the time-series ionic current data into discrete states can make errors, introducing erroneous partitions (extra states) or failing to call enough partitions (missed states). The combination of enzyme missteps and partitioning errors means that the reported ionic current states do not always faithfully represent the true sequence of the DNA strand being sequenced. The ONT base caller has some ability to handle and ameliorate these error modes. By using the ONT base calls and

reconstructing the ionic current, we avoid the difficulties presented by these complex error modes while still harnessing the power of current-to-current alignment.

[0060] At block 514, a BLAST alignment engine 408 of the alignment determination system 412 bins the analog values corresponding to the reference genome and the series of predicted analog values. In some embodiments, the binning may be constructed based on the composition of the ionic current-to-sequence model used in base calling. For example, if a k-mer model with $k = 5$ is used, the k-mers include $4^5 = 1024$ distinct sequence states, each with an associated ionic current. Given a number of bins N_{bins} , the partitions between bins may be chosen so that each bin contained the same number of sequence states. In other embodiments, different sized k-mers and/or different numbers of bins may be used.

[0061] At subroutine block 516, a subroutine is performed wherein the BLAST alignment engine 408 scans the binned reference genome for a seed of the binned series of predicted analog values. In some embodiments, the subroutine may include using a Mealy finite state machine (FSM) to search for seeds. FIG. 6 illustrates an example of a Mealy FSM usable to find seeds according to various aspects of the present disclosure. As shown, the Mealy FSM can be graphically represented as a collection of states linked by transitions triggered by inputs (individual bases or ionic currents from the reference database), or mathematically represented by a “transition matrix” and an “emission matrix.” The illustrated Mealy FSM shows the use of k-mers instead of binned values for the sake of simplicity, but similar techniques can be used to compare binned values to binned values in other embodiments of the present disclosure.

[0062] In the example, the genetic alphabet is made up of only As and Ts, and is using k-mers with $k=3$. All 3-mers present in the query sequence are added to the list of possible seeds. The state machine has a distinct state (each oval) for each possible substring of the possible seeds ($\{\text{---}, \text{A--}, \text{T--}, \text{AA-}, \text{AT-}, \text{TA-}\}$). Arrows show state-to-state transitions given that the next incoming base is an A or a T. The Mealy FSM emits on transitions, rather than from states. When a state gains an input that forms a completed seed, it emits a seed hit and returns to the initial blank state. The entire machine can be concisely expressed using a transition and emission matrix. The transition matrix summarizes which state an initial state will transition into given a

particular input. The emission matrix summarizes what seed hit is emitted from each state given different inputs (3-mer values) or if no seed is emitted (X values). Scanning each base in the reference genome amounts to a single lookup from both the transition and emission matrices.

5 **[0063]** Simply, the FSM-based seed scan works as follows. Starting in the null state (" - -" in FIG. 6), we read in the bases from the reference database one-at-a-time. As each base is read in, we move along the corresponding pathway from the initial state to a new state. Each time the input base takes us along an emitting pathway (pathways AAT, AAA, ATA, and TAA that return to the {"---"} state in FIG. 6), we report a seed
10 and continue. This procedure continues for the entire length of the reference genome and results in a complete list of all seed locations within the reference genome. Computationally, this procedure requires N lookups from the transition matrix and emission matrix, where N is the length of the reference genome.

[0064] In some embodiments, the Mealy FSM seed scanning technique is adapted to
15 work with continuously valued ionic current inputs. In the standard Mealy FSM seed scanning technique, each entry in a query sequence will have exactly one of the four possible values {A; C; G; T}. Consequently, what is meant by an "exact match" during the seed phase in the standard BLAST technique is clear: all letters in the query word must be identical to those at the reference location for a seed to be reported. The
20 discrete nature of the DNA sequence also makes the scoring of candidate matches during the extension phase of the standard BLAST technique straightforward. In the simplest case, a positive score (i.e. +1) can reward each pair of matched bases in the alignment and a negative score (i.e. -1) can penalize each pair of mismatched bases. More generally, the match score between any pair of bases can be expressed concisely
25 as a 4x4 scoring matrix.

[0065] To extend BLAST to handle current-to-current alignment, the technique is adapted to handle continuously valued inputs. In some embodiments, the seed-finding phase is adapted by discretizing the input ionic currents. In such embodiments, the query signal is binned into a finite number of discrete values, as is the predicted signal
30 of the reference database. Exact bin-to-bin matches are used for a seed to be reported in such embodiments. During seed extension, the technique reverts to compare the continuous (not binned) ionic currents in both the query and the reference. Match

scores are computed as a rescaling of the z-score between the query and reference currents, with smaller z-scores yielding more favorable alignment scores:

$$z = \frac{i_Q - i_R}{\sqrt{\sigma_Q^2 + \sigma_R^2}}$$

[0066] In some embodiments, the continuously valued inputs may be handled in another way. For example, in some embodiments, the seed-finding phase is adapted by comparing k-mers of the predicted bases of the query signal to k-mers of the reference database. The comparisons between k-mers are scored on a more complex scoring matrix than the 4x4 scoring matrix described above. The k-mer scoring matrix is computed based on how similar or different the ionic current of the query signal would be to the predicted ionic current of the corresponding k-mer of the reference database.

[0067] In some embodiments, a bin-free method of seed scanning may be used for the continuously valued inputs. This technique may also use a Mealy FSM, but with "fuzzy", rather than deterministic, logic. The query is still binned to give a finite number of possible seeds and states, but the input from the reference is kept as continuously-valued. As such, the instantaneous "state" of the FSM is expressed not as a single state value but rather as a vector of the fractional occupation of all possible states. Likewise, the "inputs" from the reference database are vectors expressing the relative input contribution along the different alphabet elements. Seeds are reported when the build-up occupation of an emission state exceeds some threshold value.

[0068] A simple example in FIG. 7 demonstrates this method for the case of continuously valued inputs between 0 and 1. In FIG. 7, inputs are now expressed as continuously valued numbers between 0 and 1. These inputs can be thought of as 2-vectors with the two entries representing the "fraction" of the input along the 0 or 1 direction. For example, an input = 0.75 is the 2-vector $(0,1) = (.25, .75)$. The instantaneous state is now expressed as an occupation vector of the fractional weight present in each of the possible states (upper table). A similar emission vector is tracked (lower table) and a seed is reported whenever a single component in the emission vector exceeds the emission threshold. In this example, the emission threshold is 0.75,

and the seed 000 is reported at the last step, in response to the input sequence 0.125 - 0 - 0.

[0069] Though either technique may be used, certain advantages may be provided with the binned method. First, the fuzzy method failed to demonstrate a meaningful improvement in performance over the binned method. It seems that for the purposes of finding short, exact seed matches, binning does not appreciably diminish the information in the signal. Indeed, in many cases seed finding was easier with the binned method, as in this case we did not require tuning of the emission threshold parameter, which is strongly sensitive to the seed size and the chosen discretization of the query signal.

[0070] Secondly, the fuzzy method is considerably more costly in terms of computation. Rather than a simple lookup at each step in the scan of the database, the fuzzy method requires a matrix multiplication, where an $N_{\text{states}} \times N_{\text{states}}$ size matrix multiplies the instantaneous state vector in order to progress the scan by a step. For typical iiBLAST run parameters, N_{states} is well over 500, so this sort of matrix multiplication is quite cumbersome.

[0071] Returning to FIG. 5, at subroutine block 518, a subroutine is performed wherein, for each seed location in the reference genome, the BLAST alignment engine 408 extends and evaluates the predicted analog values compared to the analog values corresponding to the reference genome. A seed extension technique, as described below, generates a set of candidate alignments, each of which is associated with an alignment score. The BLAST alignment engine 408 evaluates the set of candidate alignments based on their alignment scores to determine the confidence that each candidate represents a meaningful alignment. One or more of these candidate alignments may then be provided by the BLAST alignment engine 408 as the alignment for the sequence read.

[0072] In some embodiments, a non-standard Smith-Waterman-style seed extension method that allows for gapped alignments may be used in both ssBLAST and iiBLAST in order to cope with the high rate of insertions and deletions (indels) in nanopore sequencing data.

[0073] In the simplest seed extension method, each cell (i; j) in the growing alignment matrix can only be entered via a step from cell (i-1; j-1), corresponding to stepping one location forward in both the query and the reference. In our gapped alignment method, each cell can be entered from several initial cells, with each different pathway in weighted by an associated transition penalty. For each cell, all possible transition pathways in are scored, and the cell is populated with the best of the several scores. This gapped alignment strategy is able to handle the indels and poorly measured states common in nanopore reads.

[0074] FIG. 8 is a schematic drawing that illustrates the simple gapped alignment seed extension method according to various aspects of the present disclosure. The gapped alignment technique evaluates multiple possible transitions into a given cell. Here, M(1, 4) is populated with the maximum score of 6 different potential transitions in. The A transition is a "hold," where the query position progresses, but the reference position stays constant. The hold transition pays a penalty P_H in addition to the match score of query entry 1 and reference entry 4 ($S(1, 4)$). The B transition is a "step," where both the query and reference positions are incremented by 1. There is an associated penalty P_S along with $S(1, 4)$. The C transition is a "bad," where we decide to discard a query entry at the cost P_B . This transition avoids paying the match score $S(1, 4)$ at the expense of the bad penalty P_B . The D, E, and F transitions are different size "skips," where the reference position is incremented by a larger amount than the query position. These transitions pay a base penalty P_K to initiate a skip, and an additional penalty P_{K+} to extend the skip, as well as the match score $S(1, 4)$. Ultimately, M(1, 4) is populated with the best scoring option.

[0075] One potential downside of the gapped alignment method is that it may dramatically increase the computational cost of the extension phase. Rather than a single calculation to fill each cell as in the ungapped alignment method, this technique may use a calculation for each potential path in. Furthermore, a larger swath of the alignment matrix is explored, as the gapped alignment makes more cells conceivably reachable.

[0076] To cope with the computational cost and prevent it becoming intractable, embodiments of the present disclosure may use a windowed extension method with a finite lookout distance. As the extension proceeds, at each extension step a new row

and column are added to the alignment matrix. Rather than filling all of the newly added cells, the windowed method fills only those cells within a fixed lookout distance of the best scoring cell in the previously added row and column. This lookout cutoff prevents the computation time from blowing up as the extension gets long and the corresponding alignment matrix gets large. Instead of each subsequent extension adding linearly more cells to fill, the number of cells to fill now plateaus at a reasonable value, allowing extension to proceed in a tolerable time.

[0077] FIG. 9 is a schematic drawing that illustrates the windowed extension technique according to various aspects of the present disclosure. The existing 5x5 alignment matrix (shaded, empty cells) is extended by adding a new row and column. Of the newly added cells, only those within the lookout distance of the best scoring cell in the outer rank of the 5x5 matrix (shaded cell marked with a star) are to be filled. Red arrows show what constitute steps of distance 1. In the case of a lookout distance of 4, the shaded cells in the newly added cells would be filled while the white cells would be left unfilled.

[0078] In light of the windowed extension technique, FIG. 10 and FIG. 11 are a pseudocode listing of a technique for seed extension according to various aspects of the present disclosure. The illustrated technique may be used for both iiBLAST and ssBLAST, and extends seeds both to the right and to the left.

[0079] The method 500 then proceeds to an end block and terminates.

[0080] FIG. 12 is a block diagram that illustrates aspects of an exemplary computing device 1200 appropriate for use as a computing device of the present disclosure. While multiple different types of computing devices were discussed above, the exemplary computing device 1200 describes various elements that are common to many different types of computing devices. While FIG. 12 is described with reference to a computing device that is implemented as a device on a network, the description below is applicable to servers, personal computers, mobile phones, smart phones, tablet computers, embedded computing devices, and other devices that may be used to implement portions of embodiments of the present disclosure. Some embodiments of a computing device may be implemented in or may include an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), or other

customized device. Moreover, those of ordinary skill in the art and others will recognize that the computing device 1200 may be any one of any number of currently available or yet to be developed devices.

5 **[0081]** In its most basic configuration, the computing device 1200 includes at least one processor 1202 and a system memory 1204 connected by a communication bus 1206. Depending on the exact configuration and type of device, the system memory 1204 may be volatile or nonvolatile memory, such as read only memory (“ROM”), random access memory (“RAM”), EEPROM, flash memory, or similar memory technology. Those of ordinary skill in the art and others will recognize that system
10 memory 1204 typically stores data and/or program modules that are immediately accessible to and/or currently being operated on by the processor 1202. In this regard, the processor 1202 may serve as a computational center of the computing device 1200 by supporting the execution of instructions.

15 **[0082]** As further illustrated in FIG. 12, the computing device 1200 may include a network interface 1210 comprising one or more components for communicating with other devices over a network. Embodiments of the present disclosure may access basic services that utilize the network interface 1210 to perform communications using common network protocols. The network interface 1210 may also include a wireless network interface configured to communicate via one or more wireless communication
20 protocols, such as Wi-Fi, 2G, 3G, LTE, WiMAX, Bluetooth, Bluetooth low energy, and/or the like. As will be appreciated by one of ordinary skill in the art, the network interface 1210 illustrated in FIG. 12 may represent one or more wireless interfaces or physical communication interfaces described and illustrated above with respect to particular components of the computing device 1200.

25 **[0083]** In the exemplary embodiment depicted in FIG. 12, the computing device 1200 also includes a storage medium 1208. However, services may be accessed using a computing device that does not include means for persisting data to a local storage medium. Therefore, the storage medium 1208 depicted in FIG. 12 is represented with a dashed line to indicate that the storage medium 1208 is optional. In any event, the
30 storage medium 1208 may be volatile or nonvolatile, removable or nonremovable, implemented using any technology capable of storing information such as, but not

limited to, a hard drive, solid state drive, CD ROM, DVD, or other disk storage, magnetic cassettes, magnetic tape, magnetic disk storage, and/or the like.

[0084] Suitable implementations of computing devices that include a processor 1202, system memory 1204, communication bus 1206, storage medium 1208, and network interface 1210 are known and commercially available. For ease of illustration and because it is not important for an understanding of the claimed subject matter, FIG. 12 does not show some of the typical components of many computing devices. In this regard, the computing device 1200 may include input devices, such as a keyboard, keypad, mouse, microphone, touch input device, touch screen, tablet, and/or the like. Such input devices may be coupled to the computing device 1200 by wired or wireless connections including RF, infrared, serial, parallel, Bluetooth, Bluetooth low energy, USB, or other suitable connections protocols using wireless or physical connections. Similarly, the computing device 1200 may also include output devices such as a display, speakers, printer, etc. Since these devices are well known in the art, they are not illustrated or described further herein.

PERFORMANCE EXPERIMENTS

[0085] Performance of the iiBLAST technique was compared against the standard sequence-based BLAST algorithm. The two techniques were tested on data produced by the MinION device from Oxford Nanopore Technologies (ONT) (that is, on data produced using an existing commercial nanopore sequencing device).

[0086] To evaluate the relative performance of a heuristic alignment using ionic current in place of sequence, we implemented two versions of the BLAST technique in MATLAB: one (ssBLAST) comparing sequence-to-sequence, the other (iiBLAST) comparing current-to-current. Both techniques operate as described above, using Mealy FSMs and windowed Smith-Waterman-style alignment techniques allowing gaps. Gapped alignment helps to get strong alignments given the high frequency of insertions and deletions (indels) in typical nanopore reads. Ungapped alignments of nanopore data quickly encounter an indel and the query and reference get desynchronized, leading to a bad alignment.

[0087] With the two versions of BLAST implemented, we evaluated their relative performance by using both ssBLAST and iiBLAST to align the same set of nanopore

reads against a database of reference genomes. For this study, we used nanopore reads of the M13mp18 bacteriophage genome from a previously published study using ONT's MinION device. Data were downloaded from the European Nucleotide Archive, accession number ERR739515. The MinION reads were aligned using both
5 ssBLAST and iiBLAST against a 30.2 Mb subset of the NCB viral genome database containing the M13mp18 genome along with the sense and antisense sequences of 565 other viral genomes. To standardize run time, all alignments were run on a desktop computer with a 12 core Intel® Core™ i7-5820K CPU @ 3.30 GHz with 32 GB RAM. We ran a total of 1977 reads against the reference database using both algorithms, each
10 with fixed run parameters. Reads were truncated to a constant 100 nt length to give a wider variety of read accuracies and to ensure parameter stability.

[0088] For the validation experiment, we ran both iiBLAST and ssBLAST with fixed parameters. These parameters gave satisfactory performance during pre-run trials, but were not extensively optimized. A subset of 19 reads of variable read accuracy were
15 aligned against the reference viral database using a range of values for each of the run parameters. The parameters producing the best performance for this trial set (in terms of number of reads aligned and strength of alignment) were used for the full performance evaluation experiment. Although it is likely that further fine-tuning of the parameters of both algorithms could moderately improve performance, the alignment
20 results were qualitatively similar within reasonable choices of the run parameters. The final parameters we used are shown in FIG. 13. The parameters work as follows.

1. Seed size (w0): number of bases (ssBLAST) or currents (iiBLAST) making up the seeds for the first phase of the BLAST algorithm. Typically, BLAST operates with a larger seed size (12 or 14 is common). However, since seeds are only found
25 at exact matches, seeds must be smaller for the low accuracy nanopore data. Setting the seed size too small hurts run time as too many candidate seeds are found.

2. Number of bins (Nbins): the number of bins used to discretize the currents for the seed phase. Too many bins makes exact matches too rare, while too few can cause many falsely matched seeds. There is a balance between Nbins and w0 which
30 yields good performance. Bins are chosen to partition the states in the sequence-to-current model into evenly-populated bins.

3. Match penalty (p_{match}): how much a base-to-base or current-to-current match is rewarded (positive values) during the seed extension phase of the BLAST algorithm. In the case of iiBLAST and its continuously valued signal, matches between two currents i_1 and i_2 with variances $\sigma_{i_1}^2$ and $\sigma_{i_2}^2$ are scored as:

$$score = p_{\text{match}} - \frac{i_1 - i_2}{\sqrt{\sigma_{i_1}^2 + \sigma_{i_2}^2}}$$

4. Mismatch penalty (p_{mismatch}): how much base-to-base mismatch is penalized (negative values) during the seed extension phase. For iiBLAST, the scores are calculated as a rescaling of the distance between the currents (see previous), so p_{match} also sets p_{mismatch} and this second parameter is not needed.

5. Step penalty (p_s): how much a forward step transition is penalized (negative values) during seed extension. A value of zero for the step penalty means that there is no penalty for a step transition.

6. Skip penalty (p_K): penalty for initiating a skip during seed extension.

7. Skip extension penalty (p_{K+}): penalty for extending an already-initiated skip during seed extension.

8. Bad penalty (p_B): penalty to completely discard a measured state during seed extension. This is unimportant to the ssBLAST algorithm, as a measurement can only ever simply mismatch with the reference and take the penalty p_{mismatch} , which is capped. However for iiBLAST, a measurement can in principle be arbitrarily far from the reference. In this case, it is likely that the measurement was generated by some other, non-DNA-translocation-related phenomenon and is not indicative of the target DNA sequence. This p_B allows for a work around against these nonsense measurements.

9. Hold penalty (p_{hold}): penalty to match a measured state to the same reference state as the previous measurement was matched to during seed extension.

10. Longest skip (K_{max}): upper limit on the longest skip ahead allowed during seed extension. This parameter mainly works to improve run time, as calculating longer skips adds computational requirements while the longer skips are only rarely the optimal transition.

11. Lookout (L): sets the lookout window during the seed extension.

12. Max misses ($M_{\max}$): maximum number of consecutive seed extensions allowed while failing to improve on the best alignment score before seed extension is terminated (see FIG. 11 and FIG. 12). Improves run time by forcing the extension to end if it is not proving productive.

13. Termination threshold (T_{term}): how far the present alignment score during extension is allowed to fall below the best score observed previously in extension before extension is terminated. Improves run time by stopping extensions once they stop generating better scores.

10 [0089] The validation experiment was conducted using fixed length 100 base reads, rather than the full length, several thousand base reads typical of nanopore sequencing data. As discussed above, the use of smaller reads provided a wider range of sample sequencing accuracies over which to evaluate performance. Additionally, constant length reads were important for the purposes of this validation for a second reason; we
15 found that the run parameters leading to good performance for both ssBLAST and iiBLAST were dependent upon the query length.

[0090] Specifically, the seed size parameter must change for long or short read lengths. For short reads, typical BLAST seed sizes of around 14 bases do not yield good results. This is because error-prone sequencing is unlikely to yield 14 consecutive
20 correctly called bases within a short read. So, for shorter reads, a smaller seed size is necessary.

[0091] However, small seed sizes (i.e. 7) are not practical for longer read lengths. In this case, the number of seed words existing within the query read approaches the total number of possible seed words. The total number of unique seed words for a seed
25 size w_0 and an alphabet of size d is given by $N = d^{w_0}$. For ssBLAST, d is 4 (the alphabet is {A; C; G; T}) and for iiBLAST d is equal to the number of bins used to partition the currents (N_{bins}). So, as our read lengths grow into the thousands of bases (or currents), the list of seed words present in the read becomes a significant fraction of the total possible seed words. For example, a random 5000 base sequence can be
30 expected to contain 30% of all 7-letter seed words, meaning a seed will be found at 30% of all positions in the reference genome. This density of found seeds is clearly

too high to be useful, and results in the BLAST algorithm effectively calculating a complete Smith-Waterman gapped alignment against the reference genome.

5 [0092] Overall, the important parameter for variable-length reads providing good performance is the ratio of seed words present in the read to the total number of possible seed words. This ratio can be increased (reduced) by shortening (lengthening) the seed size w_0 or by reducing (increasing) the number of bins N_{bins} in the case of iiBLAST.

10 [0093] Ultimately, the basic BLAST algorithm is not the best approach to error-prone nanopore data. The best application of this work on current-to-current alignment will be to integrate it into a nanopore-specific BLAST implementation. In such an integration, the better identity from current-to-current comparisons will still improve alignments, while the surrounding architecture should allow smoother application to typical nanopore data.

15 [0094] We expect that the ssBLAST and iiBLAST methods will perform differently for reads with different accuracies. In the limiting case where read accuracy tends to 100%, the distinction between operating using bases or ionic currents becomes immaterial. The interesting case is for error-prone reads with less than 100% sequencing accuracy. Here, we expect iiBLAST to outperform ssBLAST as it avoids the artificial introduction of errors during base calling.

20 [0095] The full reads in the test dataset fell within a narrow range of accuracies (90% of reads were between 59.1% and 68.5% accurate). FIG. 14 is a chart that illustrates the distribution of accuracies for full MinION reads of M13mp18 in the test dataset. A distinct population of reads ($N = 264$) had poor accuracies (the first three columns on the left) clearly apart from the typical distribution of accuracies. Following the
25 original authors of the dataset, we attribute these reads to off-target DNA, possibly from the *Escherichia coli* in which the M13mp18 was grown up. Of the remaining on-target reads ($N = 893$), 90% fall between 59.1% and 68.5% accuracy, with a median accuracy of 64.5%.

30 [0096] To explore a wider range of accuracies, we partitioned the longer reads into 100 base subsets which were then aligned against the reference genome database. These short subset reads allowed us to test the two BLAST methods for read accuracies

ranging from 50% to 90%. In total, we ran $N = 1977$ short read subsets against the reference genome database.

[0097] We binned the results of these alignments by read accuracy and evaluated the BLAST performance within each bin. To quantify performance, we used the maximum
 5 true positive rate (TPR) provided zero false positive rate (FPR):

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

[0098] wherein TP, FP, TN, and FN are the total numbers of true positives, false positives, true negatives, and false negatives, respectively. This metric tells us, given
 10 reads with a certain accuracy, what fraction of the reads we can expect to align unambiguously to the correct reference genome. The requirement that the FPR is ideally zero helps ensure that the BLAST results are useful. The number of off-target genomes (N_{off}) can be large, so even a small FPR can result in the total number of false positives ($= FPR * N_{\text{off}}$) vastly surpassing the number of true positives ($= 1$).

15 [0099] The concept with regards to TPR/FPR is that we have the freedom to choose an acceptance threshold of alignment scores at which we will mark an alignment as meaningful. If we turn this threshold too low, we will simply accept all the candidates. At this operational point, we will keep all the true positives ($TPR \rightarrow 1$) but also keep all the false positives ($FPR \rightarrow 1$). Conversely, if we crank the threshold too high, we
 20 will reject all candidates. In this case, we will successfully discard all the false positives ($FPR \rightarrow 0$) but simultaneously discard all the true positives ($TPR \rightarrow 0$). If BLAST is generating useful results, the FPR should fall faster with an increasing acceptance threshold than does the TPR, allowing us to find a point where the $FPR = 0$ while the TPR is still nonzero.

25 [0100] FIG. 15 is a chart that illustrates TPR and FPR variation with respect to an acceptance threshold in order to select a BLAST operational point. As the acceptance threshold is tuned from low to high, both the TPR and the FPR fall from 1 to 0. The FPR falls of faster than the TPR. We choose the BLAST operational point to be the point where the FPR first reaches 0. The TPR at $FPR = 0$ (star) is the metric discussed
 30 in FIG. 17.

[0101] The performance metric of true positive rate provided zero false positive rate as a function of read accuracy was extracted from the receiver operating characteristics of the iiBLAST and ssBLAST algorithms over sets of reads binned by accuracy. The aligned reads were binned by their sequencing accuracy into 10 bins each spanning a 4% range in accuracy (i.e. 74 -78%). For each set of reads, we varied the acceptance threshold (how good an alignment score is required to report a match) and plotted the resulting true positive rate against the resulting false positive rate.

[0102] FIG. 16 includes four charts that illustrate receiver operating characteristics for different read accuracies. The true positive rate is plotted against the false positive rate for reads with different accuracies from 86-90% (a); 74-78% (b); 62-66% (c); and 50-54% (d). The square points show the performance of ssBLAST (generally lower on the y-axis) and the round points show the performance of iiBLAST (generally higher on the y-axis). The starred data points mark the true positive rate at zero false positive rate. The lines connecting the data points are to guide the eye.

[0103] We extracted the TPR value corresponding to $FPR = 0$ for each set of accuracies to quantify performance as a function of read accuracy. We looked specifically at this $FPR = 0$ point as this performance point best reflects the requirements of a typical BLAST-based experiment. In a typical experiment, the reference database will contain almost exclusively off-target genomes, meaning that we expect an overwhelming number of true negatives. Consequently, even a miniscule FPR will lead to most matches being false positives. Thus, a useful BLAST implementation should operate at FPR very close to 0 to generate useful results.

PERFORMANCE EXPERIMENT RESULTS

[0104] FIG. 17 is a chart that illustrates a comparison of true positive rate as a function of read accuracy for ssBLAST and iiBLAST techniques. The maximum true positive rate provided a zero false positive rate is plotted as function of read accuracy for iiBLAST (circles) and ssBLAST (squares). The dashed lines show $TPR = 0$ and $TPR = 1$, corresponding to no useful performance and perfect performance, respectively. Reads are binned by their accuracy into bins of width 0.04 (i.e. 62-66%). The two techniques' performance converges at low and high read accuracies (56% bin, 88% bin), but iiBLAST performs better for intermediate read accuracies (60% bin to

84% bin), unambiguously aligning a greater portion of the available reads to the correct reference genome.

[0105] In addition to measuring the rate of on-target read alignment, we also evaluated the significance of the alignments produced by both ssBLAST and iiBLAST.

5 We quantified alignment significance by calculating a p-value for each on-target alignment based on the alignment's score against the distribution of scores for all off-target alignments within the database.

[0106] Alignment scores provide a relative ranking between alignments (higher scores are better), but the actual value of the score is not inherently statistically
10 meaningful. The value of the alignment score is a function of the parameters used in the alignment algorithm and is not comparable to any alignment generated using different parameters. For example, the scores from iiBLAST and ssBLAST alignments cannot be directly compared, as alignment in the two algorithms uses different score parameters.

15 **[0107]** To compare alignment quality across algorithms, we need to assign a statistical significance to the alignment based on the score. We can do this by comparing the candidate alignment score against the distribution of scores observed for off-target alignments. We expect off-target scores to increase with increasing size of the off-target genome, as a longer genome provides more opportunities for a high
20 scoring random alignment. Indeed, we observe that off-target scores increase logarithmically with genome length.

[0108] FIG. 18 includes two charts that show a length correction for alignment scores according to various aspects of the present disclosure. For each off-target reference genome, the mean best alignment score over all reads is plotted against the length of
25 the genome in nucleotides (charted points). Both ssBLAST (chart (a)) and iiBLAST (chart (b)) scores show a logarithmic dependence on the reference genome length (diagonal lines). The vertical dashed lines show the M13mp18 genome length (7249 bp). An alignment score to a differently sized genome (the unfilled red star) is shifted along the logarithmic fit to a corrected score for the M13mp18 genome length (filled
30 red star).

[0109] For all of the reads, we took the best alignment score to each off-target reference genome and used the genome length correction function to bring all scores to the reference length of M13mp18. This provides us an empirical distribution of scores for random alignments. FIG. 19 includes two charts that illustrate conversion of alignment scores to p-values according to various aspects of the present disclosure. We fit an extreme value distribution, expressed by the equation:

$$f(s|\mu, \sigma) = \sigma^{-1} e^{\left(\frac{s-\mu}{\sigma}\right)} e^{-e^{\left(\frac{s-\mu}{\sigma}\right)}}$$

[0110] to the tail of the distribution of off-target alignment scores (gray columns) for both ssBLAST (a) and iiBLAST (b). An alignment score s^* is converted to a p-value by integrating the fit distribution from s^* to $+\infty$ (shaded region). The histograms are pdf-normalized and displayed on a log y-scale.

[0111] Using this fit, we can calculate the probability that a given alignment score s^* was generated by an alignment of a random sequence to the reference genome by:

$$p_{\text{random}}(s_*) = \int_{s_*}^{+\infty} f(s|\mu, \sigma) ds$$

[0112] where f is defined by the extreme value distribution equation above, and $[u]$ and $[d]$ are specific to the technique used to generate the alignment.

[0113] This p_{random} is the p-value for the alignment, and can be interpreted as a statement about how large of a reference database the read could be aligned against before a false positive alignment would be likely to generate a better score. For example, a p-value of 10^{-4} means that a random alignment to a 7249 bp reference genome will generate a better score once in 10,000 times. Therefore, this read could be aligned against a database of 72.49 Mb before we are likely to see a false positive outscoring the on-target alignment:

$$Size_{\text{database}} = \frac{Size_{M13mp18}}{p_{\text{random}}}$$

[0114] Alignments with p-values below 10^{-10} are good enough that they should align unambiguously against the 2.9 Tb database of all known genomes in the NCBI database.

[0115] FIG. 21 is a chart that illustrates the the p-value comparison results between ssBLAST and iiBLAST. Each on-target read in the validation experiment is plotted as a single point at the p-value of the read's alignment using ssBLAST (x-axis) and its p-value using iiBLAST (y-axis), color coded by the read's sequencing accuracy (color bar). The diagonal line shows $x = y$, representing where the two techniques yield equally strong alignments. Points below the line represent reads where iiBLAST produced a stronger alignment. Points above the line represent reads where ssBLAST gave a stronger alignment.

[0116] Comparing the p-values of on-target alignments generated by ssBLAST and iiBLAST, we find that the alignment p-values are consistently several orders of magnitude smaller (better) using iiBLAST than ssBLAST. Furthermore, the iiBLAST method generates stronger alignments than ssBLAST across all read accuracies tested. Consequently, although both methods had a 100% true positive rate for the high accuracy reads (86-90%) in our test set against our trial database, iiBLAST would continue to outperform ssBLAST if the search were expanded to include a larger reference database.

[0117] Over the 1977 reads aligned in this study, the mean run time for iiBLAST was 394 seconds and the mean run time for ssBLAST was 1807 seconds. The improvement in run time for iiBLAST over ssBLAST is attributable entirely to spending less time on seed extension. In general, the seeds found during the scan phase were higher quality for iiBLAST than ssBLAST, meaning that a lower percentage of the candidate seeds led to uninteresting final alignments. So, ssBLAST spent much more time extending seeds into ultimately discarded alignments. Overall, the extension phase makes up nearly all of the computational burden. For iiBLAST, extension made up 99.8% of the total run time, with the next most costly step being the scan phase at 0.1% of the total. For ssBLAST, extension was 98.9% of total run time, with scanning taking up most of the remainder at 1.1%. The scanning phase was somewhat slower for ssBLAST as more seeds were found, and the computational cost to record a found seed was consequently larger than for iiBLAST. A complete accounting of the average per-read time spent during the various phases of the BLAST algorithm can be found in FIG. 20.

CONCLUSION

[0118] We've shown that by implementing a version of the BLAST algorithm that makes current-to-current, rather than sequence-to-sequence, comparisons between nanopore reads and reference genomes, we are able to dramatically improve the performance of the BLAST search for error-prone nanopore sequencing reads. The next steps of scaling and integrating this technique bear discussion, as do the potential implications of this new method on the future of nanopore sequencing.

[0119] We conducted our validation experiment using a relatively small reference database. Using a small reference allowed us to test many reads using both ssBLAST and iiBLAST using limited computing power and without extensively optimizing our code for parallelization. However, the results from this experiment are extensible to applications using much larger references. The primary concerns with scaling to a larger reference are keeping the false positive rate low and managing the computation time.

[0120] For larger reference databases, more high-scoring false positives will arise as there are more chances for random matches to occur. However, iiBLAST's stronger alignment p-values indicate that the current-to-current method is better suited to large searches (FIG. 21). Based on the p-value results, we expect iiBLAST will outperform ssBLAST on larger databases, even at higher read accuracies where the two methods both generated 100% true positive rates during the validation experiment (FIG. 17). The orders-of-magnitude improvement in alignment significance indicates that iiBLAST queries will return useful, false-positive-free results against reference databases orders-of-magnitude larger than would be searchable using ssBLAST.

[0121] It is important that current-to-current comparison not dramatically add to the computational burden of the BLAST algorithm. In terms of computational complexity, both the seed and extension phases of the iiBLAST implementation are identical to those of ssBLAST. As discussed above, both approaches use a Mealy finite state machine to scan the reference database for seeds in $O(N)$ time (N is the size of the reference database). Likewise, both approaches use similar gapped alignment techniques requiring $O(M * L)$ time (M is the number of extensions prior to terminating the alignment; L is the window lookout distance). The iiBLAST method

does require the additional computational step of predicting the signal for the reference sequences. However, the prediction only takes $O(N)$ time (N again the size of the reference database), and only needs to be done once for a given reference genome. Once made, the prediction can be stored along with the sequence in the reference database for future use. Overall, iiBLAST aligned reads over 4 times faster than ssBLAST (FIG. 20). The improved run time is due to better efficiency in finding good seeds rather than any difference in the intrinsic computational load of the two algorithms. The iiBLAST method found fewer bad seeds that led to uninteresting alignments and thus saved time during seed extension.

[0122] Looking forward, the current-to-current method may also be useful by integrating it into a more sophisticated heuristic alignment technique better suited to long, error-prone sequencing data than the standard BLAST algorithm. Such an implementation will harness the improved read-to-reference identity offered by current-to-current alignment within the architecture of a technique specifically designed to handle the unique aspects of nanopore sequencing data. The dramatic improvement in performance achieved by using current-to-current instead of sequence-to-sequence alignment has myriad implications for both the present and future applications of nanopore sequencing.

[0123] Immediately, our ability to generate strong, unambiguous alignments with low accuracy reads will increase the effective throughput in BLAST-based nanopore sequencing experiments. Fewer reads will fail to align due to low accuracy, making a better fraction of the sequencing data useful to the researcher. It is particularly worth noting that iiBLAST consistently generates good results for reads in the 70-80% accuracy range{the typical accuracy range for 1D nanopore reads (in which only one strand of the DNA is read through the pore). In comparison, ssBLAST can only achieve similar results for reads well above 80%--above the accuracy of most 1D reads and typical only of 2D reads (in which both strands are read one after the other, then combined into a single, higher accuracy sequence). The iiBLAST method's better tolerance for low accuracy reads thus makes 1D reads substantially more useful and reduces the need for 2D reads which are necessarily only half as fast (as both strands must be read). By improving nanopore sequencing's effective throughput by increasing the fraction of usable reads, iiBLAST can reduce sample-to-answer timelines, input

sample requirements, and sequencing costs. Effectively, the iiBLAST method can increase the speed and decrease the cost of nanopore sequencing.

5 [0124] The power of the current-to-current comparison method is not limited to improving only BLAST-based nanopore sequencing applications. The same fundamental method can improve nanopore sequencing's ability to perform various other sequencing tasks, including variant detection and epigenetic mapping.

10 [0125] Looking forward, the ability to use low accuracy reads for BLAST-based sequencing experiments offers an alternative way forward for nanopore sequencing to make an impact on the larger sequencing community. To date, many efforts to improve nanopore sequencing have focused on improving raw single-read sequencing accuracy. This work shows that, while improved sequencing accuracy is certainly important, much more can be done with the currently-available low accuracy reads than was previously believed. A future nanopore sequencing device sacrificing sequencing accuracy for lower cost, faster turnaround time, improved portability, and better
15 throughput could prove useful for clinicians and researchers interested in BLAST-based applications.

[0126] The particulars shown herein are by way of example and for purposes of illustrative discussion of the preferred embodiments of the present invention only and are presented in the cause of providing what is believed to be the most useful and
20 readily understood description of the principles and conceptual aspects of various embodiments of the invention. In this regard, no attempt is made to show structural details of the invention in more detail than is necessary for the fundamental understanding of the invention, the description taken with the drawings and/or examples making apparent to those skilled in the art how the several forms of the
25 invention may be embodied in practice.

[0127] As used herein and unless otherwise indicated, the terms “a” and “an” are taken to mean “one”, “at least one” or “one or more”. Unless otherwise required by context, singular terms used herein shall include pluralities and plural terms shall include the singular.

30 [0128] Unless the context clearly requires otherwise, throughout the description and the claims, the words ‘comprise’, ‘comprising’, and the like are to be construed in an

inclusive sense as opposed to an exclusive or exhaustive sense; that is to say, in the sense of “including, but not limited to”. Words using the singular or plural number also include the plural and singular number, respectively. Additionally, the words “herein,” “above,” and “below” and words of similar import, when used in this application, shall refer to this application as a whole and not to any particular portions of the application.

[0129] The description of embodiments of the disclosure is not intended to be exhaustive or to limit the disclosure to the precise form disclosed. While the specific embodiments of, and examples for, the disclosure are described herein for illustrative purposes, various equivalent modifications are possible within the scope of the disclosure, as those skilled in the relevant art will recognize.

[0130] All of the references cited herein are incorporated by reference. Aspects of the disclosure can be modified, if necessary, to employ the systems, functions, and concepts of the above references and application to provide yet further embodiments of the disclosure. These and other changes can be made to the disclosure in light of the detailed description.

[0131] Specific elements of any foregoing embodiments can be combined or substituted for elements in other embodiments. Moreover, the inclusion of specific elements in at least some of these embodiments may be optional, wherein further embodiments may include one or more embodiments that specifically exclude one or more of these specific elements. Furthermore, while advantages associated with certain embodiments of the disclosure have been described in the context of these embodiments, other embodiments may also exhibit such advantages, and not all embodiments need necessarily exhibit such advantages to fall within the scope of the disclosure.

CLAIMS

What is claimed is:

1. A computer-implemented method for aligning genetic sequence reads to a reference genome, the method comprising:
 - 5 receiving, by a computing system, a sequence read generated by a sequencing device;
 - converting, by the computing system, the sequence read to a series of predicted analog values that would be generated by the sequencing device for the sequence read; and
 - 10 aligning, by the computing system, the series of predicted analog values to a reference genome.
2. The computer-implemented method of claim 1, wherein aligning the series of predicted analog values to the reference genome includes aligning the series of predicted analog values to analog values associated with the reference genome.
- 15 3. The computer-implemented method of claim 2, wherein aligning the series of predicted analog values to analog values associated with the reference genome includes performing an analog-to-analog Basic Local Alignment Search Tool (BLAST) alignment technique.
4. The computer-implemented method of claim 3, wherein the analog-to-analog
20 BLAST alignment technique includes binning the predicted analog values.
5. The computer-implemented method of claim 4, wherein a scan step of the analog-to-analog BLAST alignment technique includes comparing bins of the predicted analog values to bins of the analog values associated with the reference genome.
- 25 6. The computer-implemented method of claim 4, wherein binning the predicted analog values includes placing the predicted analog values into a plurality of bins determined based on a size of a k-mer to be used.

7. The computer-implemented method of claim 6, wherein the k-mer to be used has a k of 5, and wherein the plurality of bins determined based on the size of the k-mer to be used includes 1024 bins.
8. The computer-implemented method of claim 4, wherein performing the analog-to-analog BLAST alignment technique includes scanning for seeds of a size
5 determined based on the number of bins.
9. The computer-implemented method of claim 4, wherein performing an extend step of the BLAST alignment technique includes comparing un-binned predicted analog values to analog values associated with the reference genome.
- 10 10. The computer-implemented method of any of claims 2-9, further comprising generating the analog values associated with the reference genome based on sequence values for the reference genome.
11. The computer-implemented method of any of claims 1-10, wherein the sequencing device is a nanopore sequencing device, and wherein the series of predicted
15 analog values is a series of predicted ionic current values.
12. The computer-implemented method of any of claims 1-10, wherein the sequencing device is a single molecule real time (SMRT) sequencing device, and wherein the series of predicted analog values is a series of dye fluorescence values.
13. The computer-implemented method of any of claims 1-10, wherein the
20 sequencing device is an optical mapping sequencing device, and wherein the series of predicted analog values is a series of fluorochrome labeling values.
14. A non-transitory computer-readable medium having computer-executable instructions stored thereon that, in response to execution by one or more processors of a computing system, cause the computing system to perform a method as recited in
25 any of claim 1 to claim 13.
15. A computing system that includes at least one computing device having at least one processor and at least one non-transitory computer-readable medium, wherein the at least one computer-readable medium has instructions stored thereon that, in response

to execution by the at least one processor of the at least one computing device, cause the computing system to perform a method as recited in any of claim 1 to claim 13.

16. A computer-implemented method of finding an analog query pattern in a set of analog reference data, the method comprising:

5 generating, by a computing system, a set of binned query values based on the analog query pattern;

 determining, by the computing system, a seed value based on the set of binned query values;

10 searching, by the computing system, for the seed value in the set of analog reference data;

 for each location in the set of analog reference data in which the seed value is found, extending, by the computing system, the seed value into a candidate result; and

 evaluating, by the computing system, the candidate results to determine confidence values that each candidate result represents a true result.

15 17. The computer-implemented method of claim 16, wherein searching for the seed value in the set of analog reference data includes comparing binned query values of the seed value to binned values associated with the analog reference data.

20 18. The computer-implemented method of claim 17, wherein extending the seed value into a candidate result includes comparing analog values of the analog query pattern to analog values of the set of analog reference data.

19. The computer-implemented method of claim 16, wherein the analog query pattern includes a predicted analog query pattern, and wherein the method further comprises generating the predicted analog query pattern based on a discrete value query pattern.

25 20. The computer-implemented method of claim 19, wherein the discrete value query pattern is a nanopore sequence read, and wherein the predicted analog query pattern is a series of predicted ionic current values.

21. The computer-implemented method of claim 20, wherein the set of analog reference data is based on a reference genome.

22. A non-transitory computer-readable medium having computer-executable instructions stored thereon that, in response to execution by one or more processors of a computing system, cause the computing system to perform a method as recited in any of claim 16 to claim 21.
- 5 23. A computing system comprising at least one computing device that includes at least one processor and a non-transitory computer-readable medium having computer-executable instructions stored thereon that, in response to execution by the at least one processor, cause the computing system to perform a method as recited in any of claim 16 to claim 21.
- 10 24. A computer-implemented method for aligning genetic sequence reads to a reference genome, the method comprising:
- receiving, by a computing system, a sequence read generated by a sequencing device; and
 - aligning, by the computing system, the sequence read to a reference genome;
 - 15 wherein aligning the sequence read to the reference genome includes:
 - scanning the reference genome for one or more seeds based on the sequence read; and
 - extending the one or more seeds to generate one or more candidate alignments, wherein each candidate alignment includes an alignment score; and
 - 20 wherein at least one of the scanning of the reference genome and the extending of the one or more seeds is based on a series of predicted analog values associated with the sequence read .
25. The computer-implemented method of claim 24, wherein scanning the reference genome for one or more seeds based on the sequence read includes:
- 25 comparing k-mers of the sequence read to k-mers of the reference genome, wherein values of a scoring matrix used to compare the k-mers are based on similarities of predicted analog values associated with the k-mers.
26. The computer-implemented method of claim 24, wherein scanning the reference genome for one or more seeds based on the sequence read includes:

generating a series of predicted analog values associated with the sequence read and a series of predicted analog values associated with the reference genome;

binning the predicted analog values; and

5 comparing the binned series of predicted analog values associated with the sequence read to the binned series of predicted analog values associated with the reference genome.

27. The computer-implemented method of any of claims 24-26, wherein extending the one or more seeds to generate one or more candidate alignments includes using a gapped alignment technique.

10 28. The computer-implemented method of claim 27, wherein the gapped alignment technique uses a windowed extension method with a finite lookout distance.

29. A non-transitory computer-readable medium having computer-executable instructions stored thereon that, in response to execution by one or more processors of a computing system, cause the computing system to perform a method as recited in
15 any of claim 24 to claim 28.

30. A computing system comprising at least one computing device that includes at least one processor and a non-transitory computer-readable medium having computer-executable instructions stored thereon that, in response to execution by the at least one processor, cause the computing system to perform a method as recited in any of claim
20 24 to claim 28.

1/21

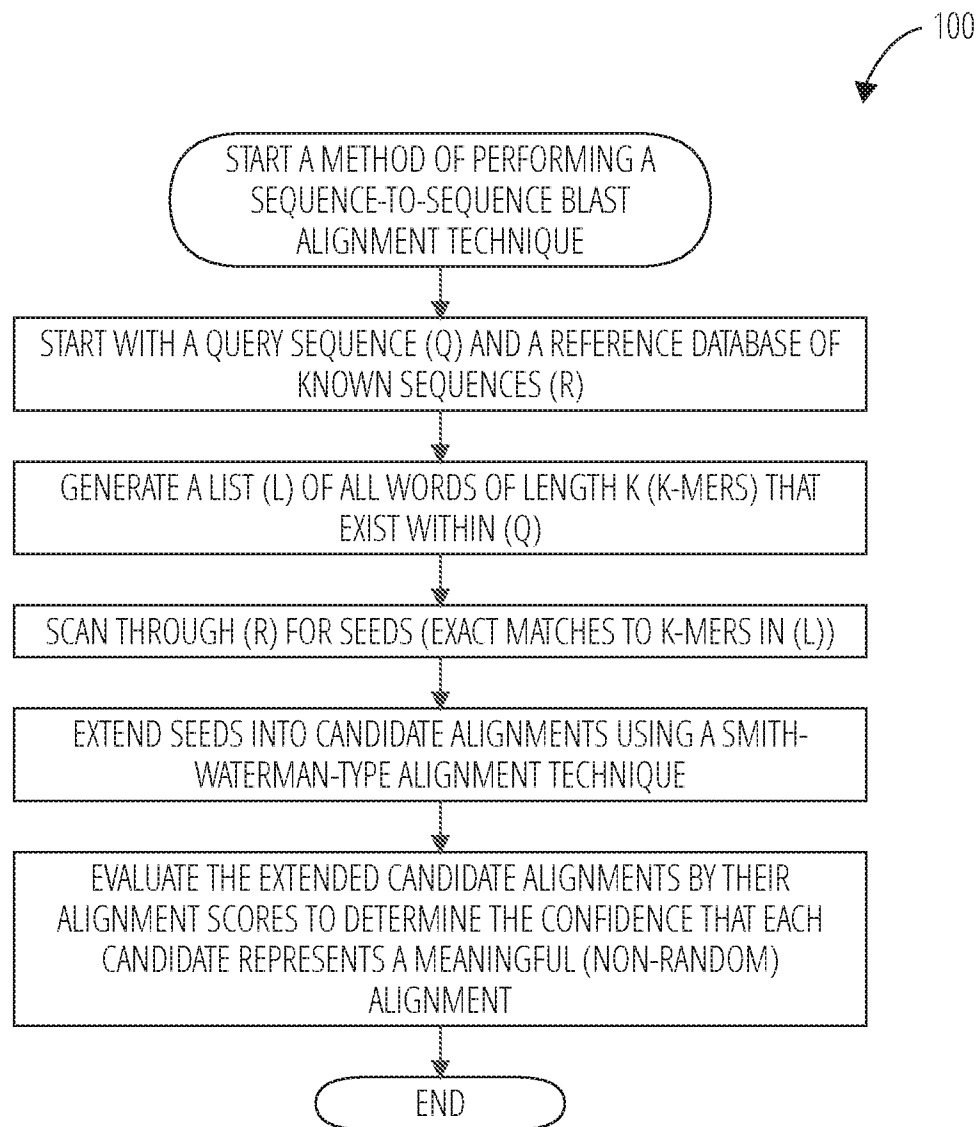


FIG. 1

2/21

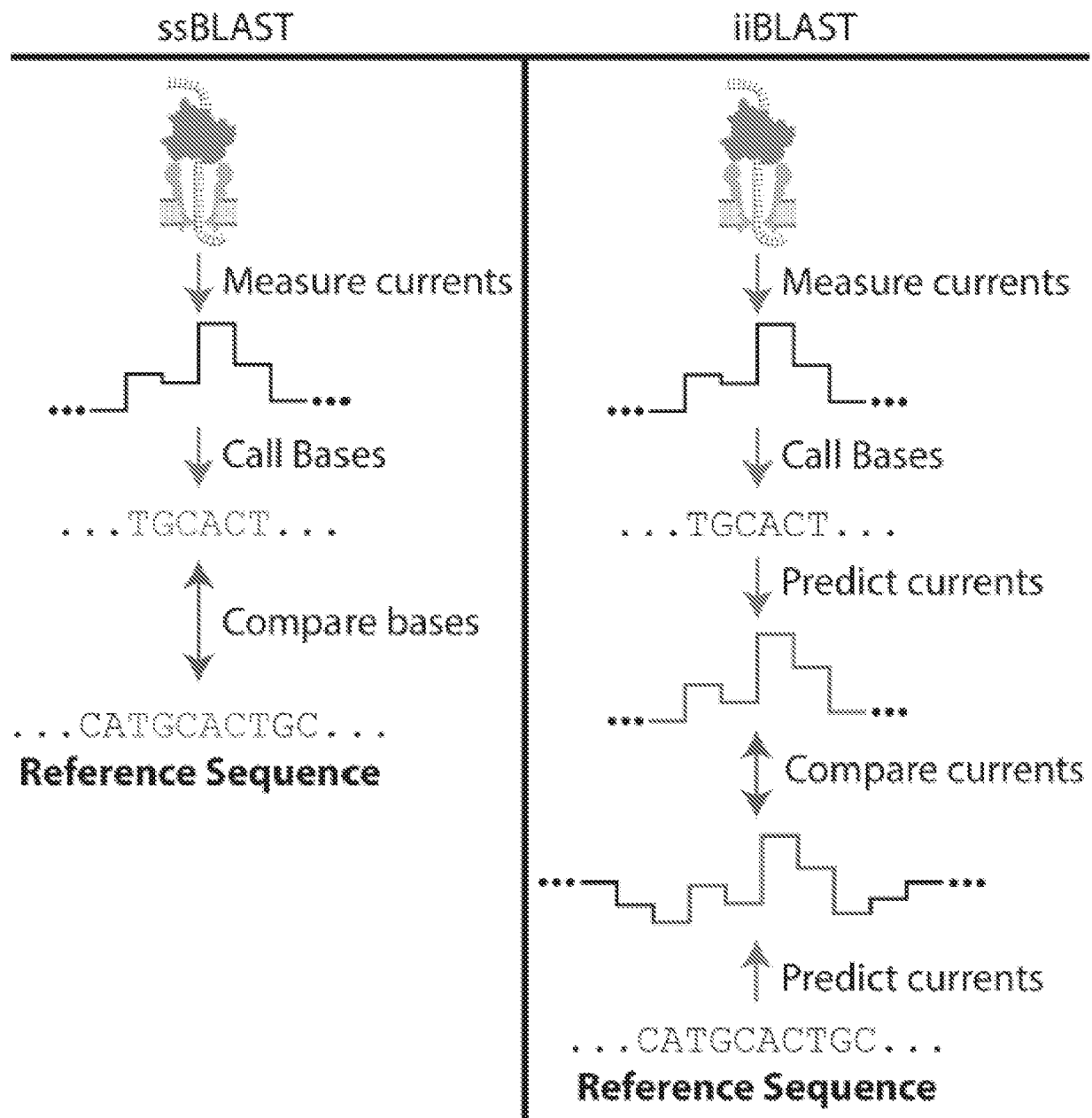


FIG. 2

3/21

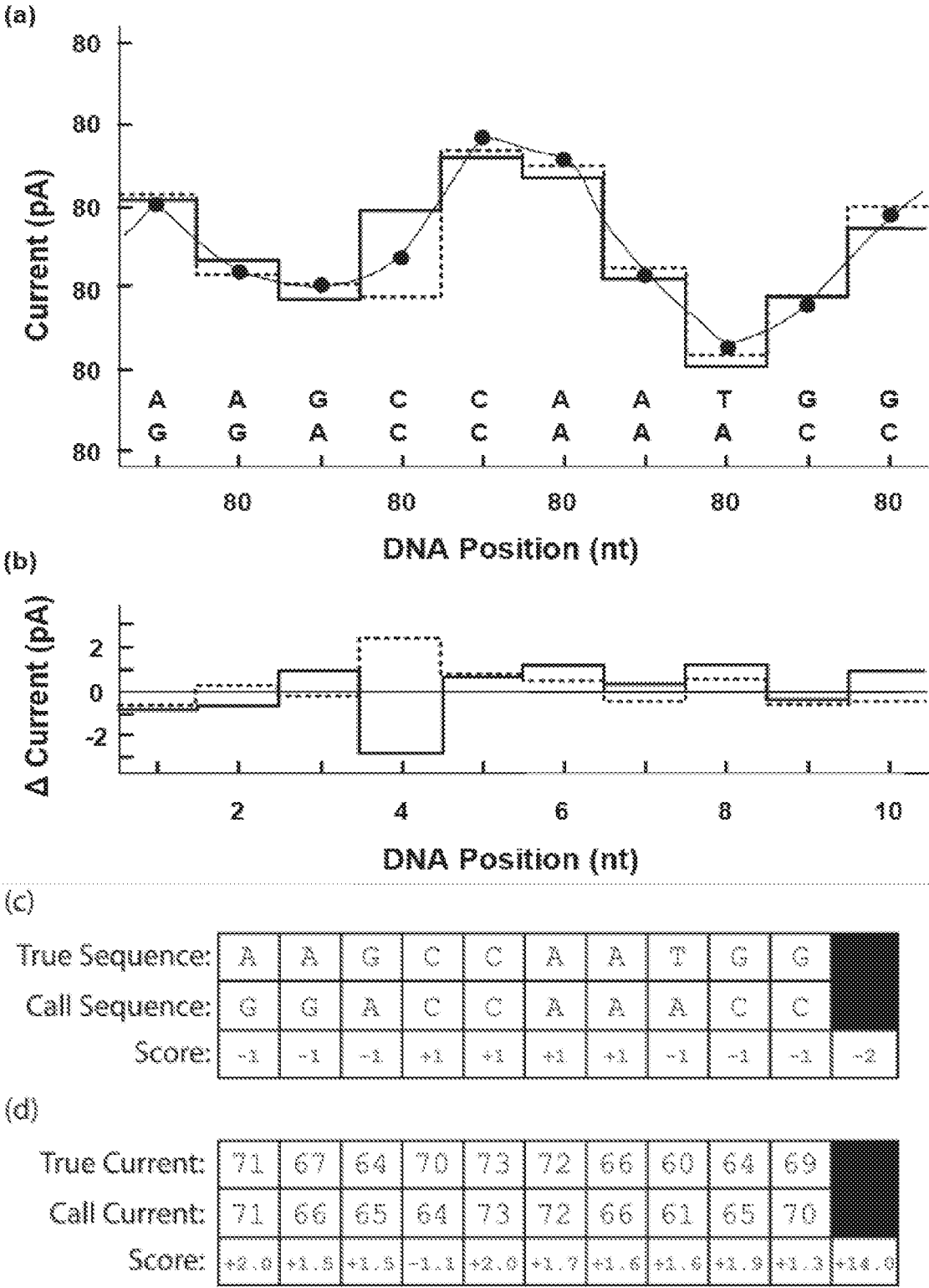


FIG. 3

4/21

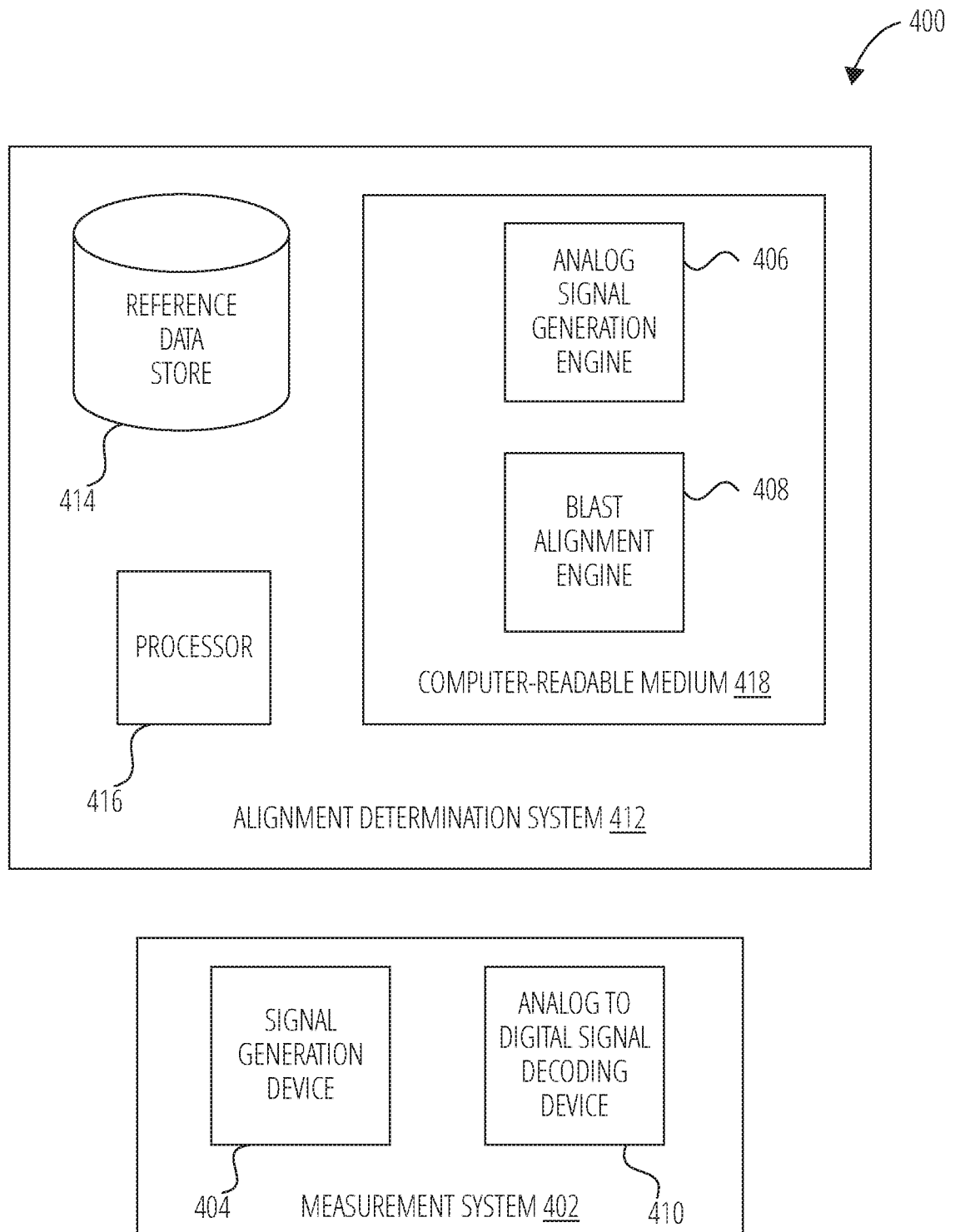


FIG. 4

5/21

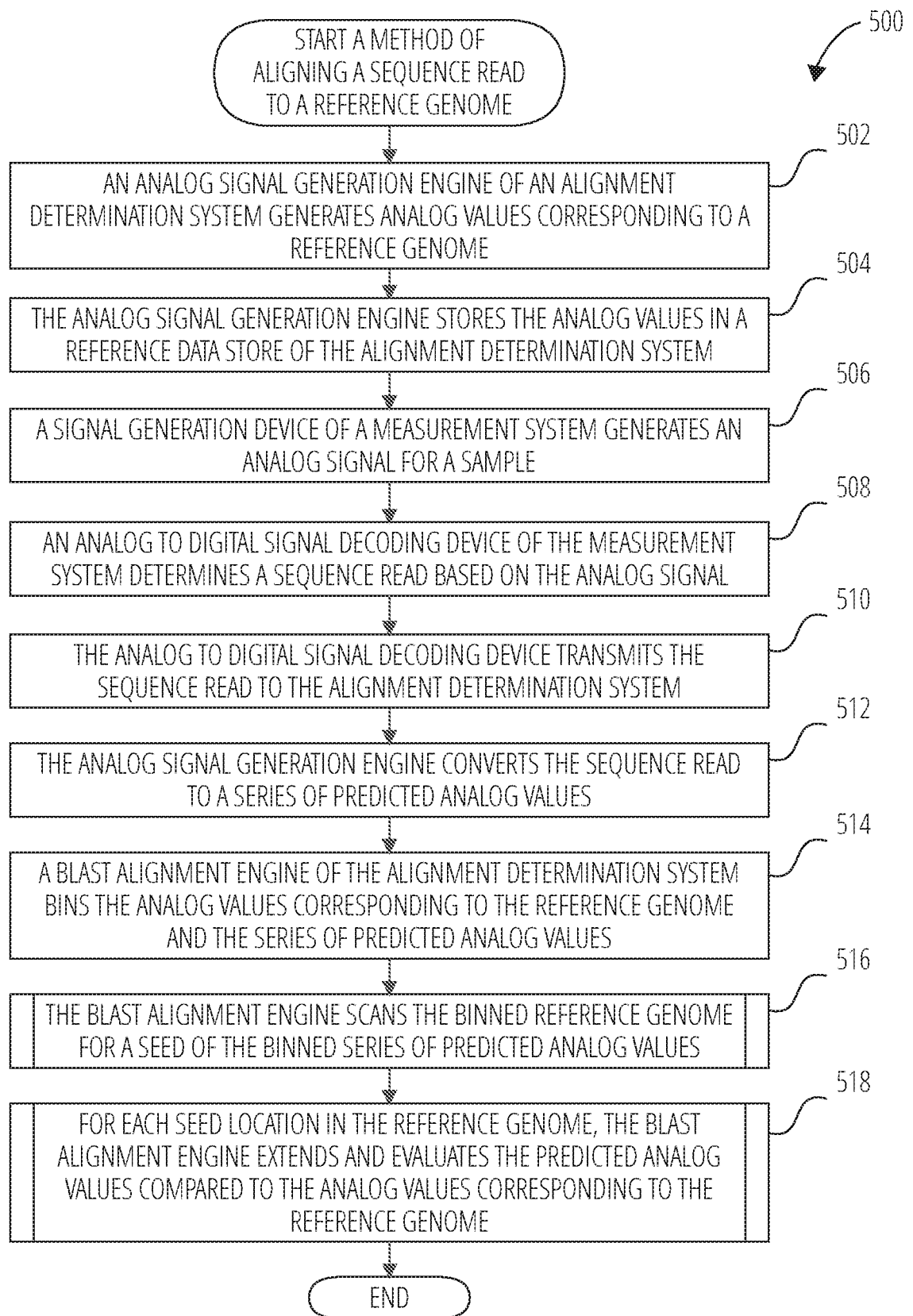


FIG. 5

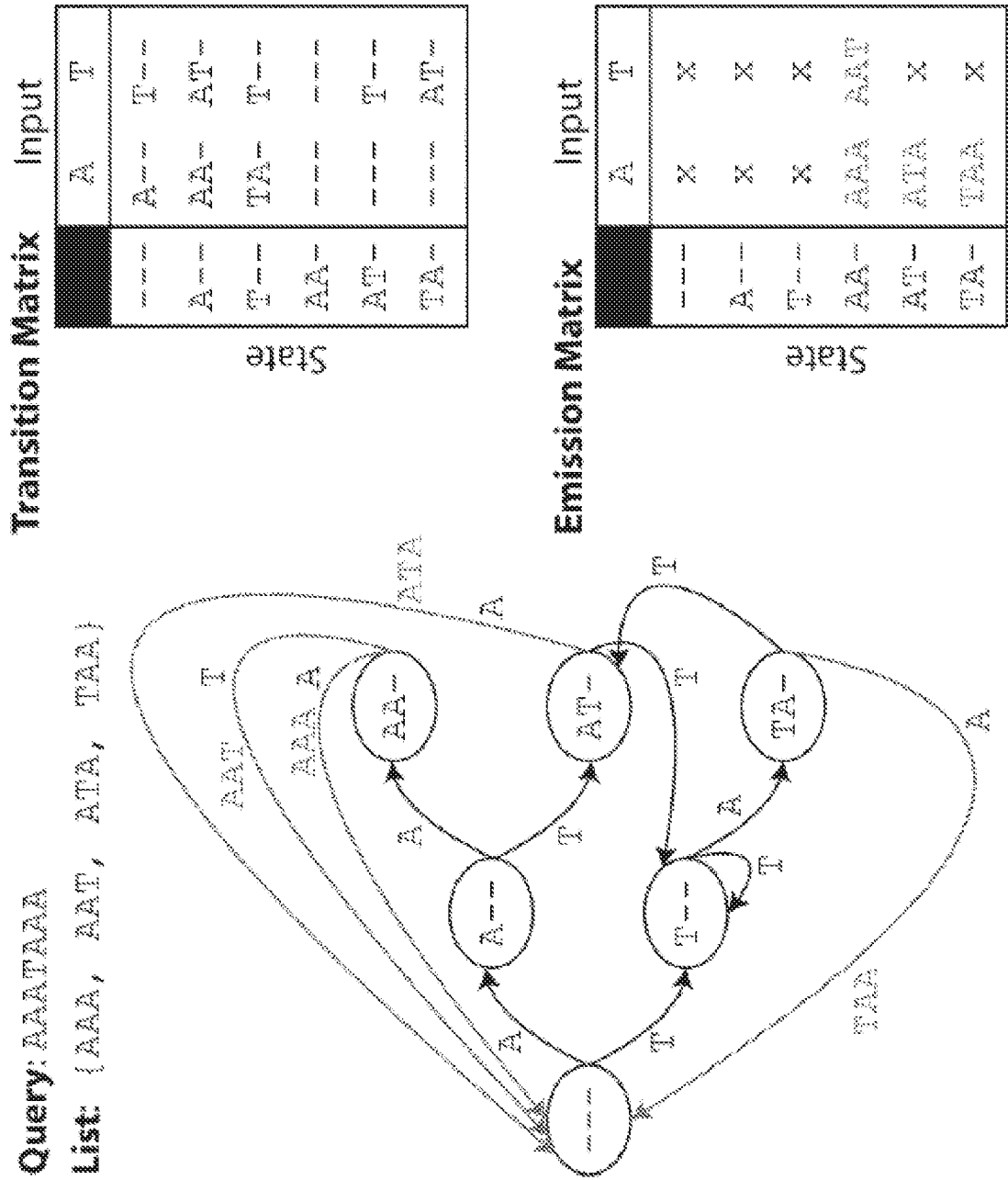
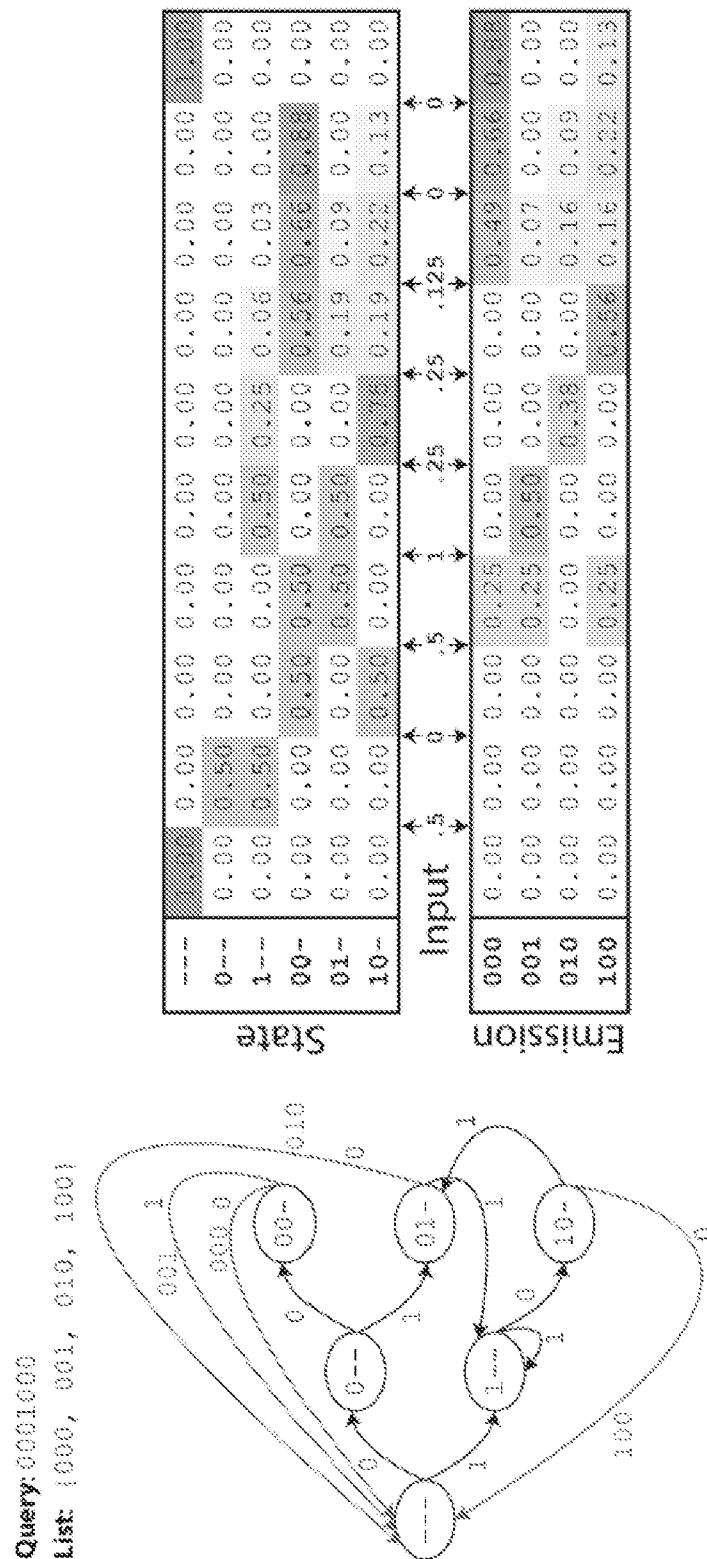


FIG. 6



70

8/21

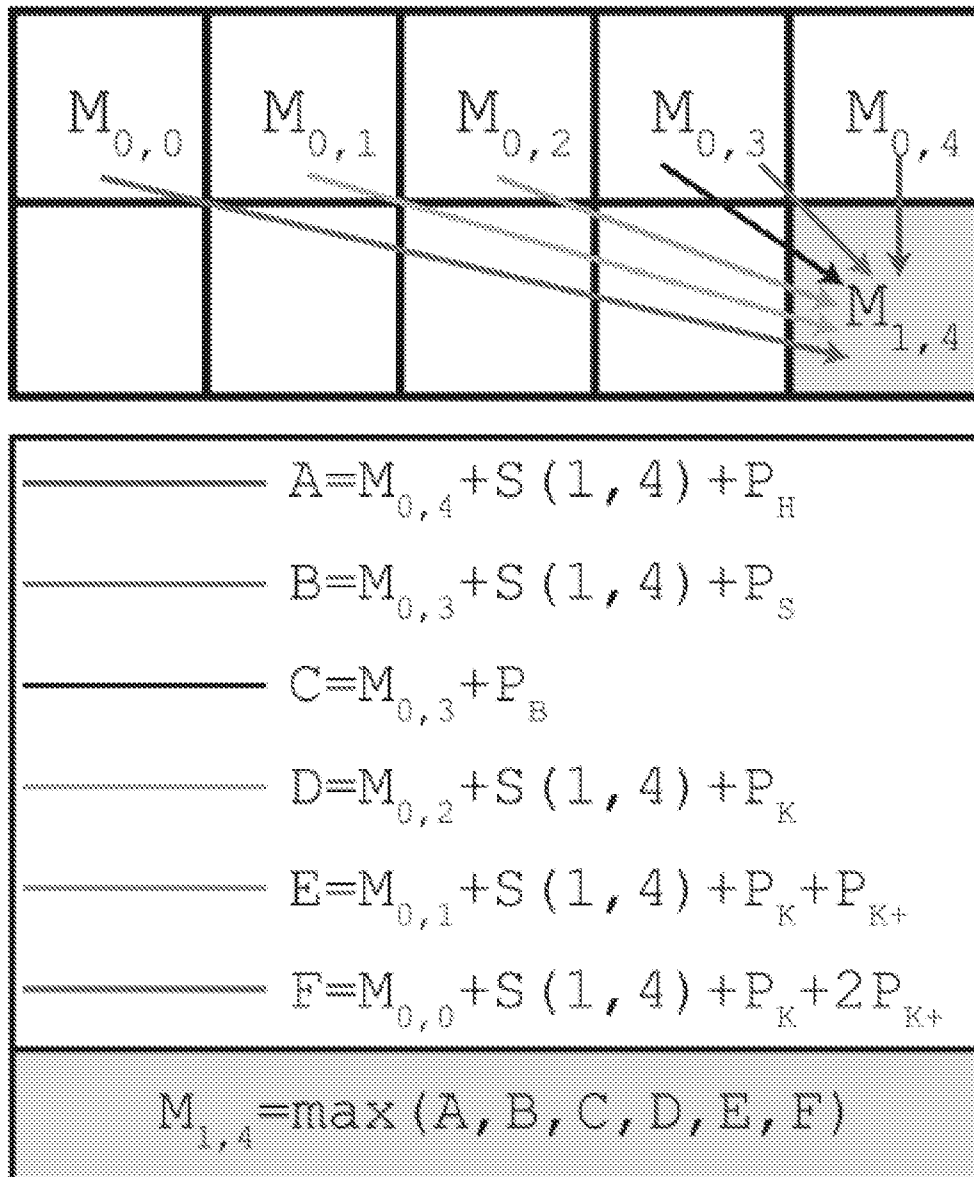


FIG. 8

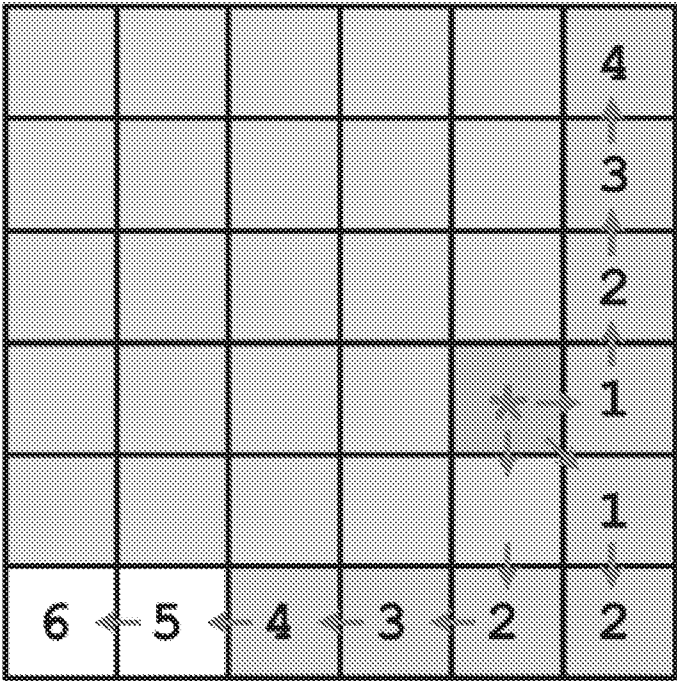


FIG. 9

10/21

```

1: Assume there is a seed found at query position  $q_i$  and reference position  $r_j$ 
2: Initialize variables
    $M \leftarrow [0]$                                 ▷ alignment matrix
    $S_{best} \leftarrow 0$                             ▷ best score in M
    $S_{rank} \leftarrow 0$                             ▷ best score in outermost row/column of M
    $loc_{best} \leftarrow (0, 0)$                     ▷ indices of  $S_{best}$ 
    $loc_{rank} \leftarrow (0, 0)$                     ▷ indices of  $S_{rank}$ 
    $q \leftarrow q_i$                                 ▷ current position in query
    $r \leftarrow r_i$                                 ▷ current position in reference
    $misses \leftarrow 0$                             ▷ counter for consecutive misses
    $STOP \leftarrow FALSE$                           ▷ flag to terminate alignment
3: while  $\sim STOP$  do
4:   if  $q > \text{length}(\text{query})$  or  $r > \text{length}(\text{reference})$  then    ▷ make sure we have not
   overrun the end of the query or reference
5:      $STOP \leftarrow TRUE$  return  $S_{best}, loc_{best}$ 
6:   end if
7:   Add new cells  $\{(x, y)\}_{new}$  to  $M$  forming a new right-most row and new bottom-most
   column.
8:   for cell  $(x, y) \in \{(x, y)\}_{new}$  do    ▷ check to see if the cell is within the lookout
   distance from  $loc_{rank}$ 
9:     if  $\text{distance}((x, y), loc_{rank}) \leq \text{lookout}$  then    ▷ for distance, see fig. G.3b
10:       $M(x, y) \leftarrow \max\{\text{all alignment paths in}\}$     ▷ for paths in, see fig. G.3a
11:    else
12:       $M(x, y) \leftarrow -\infty$ 
13:    end if
14:  end for

```

FIG. 10

11/21

```

15:    $S_{rank} \leftarrow \max\{M(\{x, y\}_{new})\}$      $\triangleright$  update  $S_{rank}$  with the best score of the newly filled
      cells
16:    $loc_{rank} \leftarrow loc(S_{rank})$   $\triangleright$  update  $loc_{rank}$  with the location of the best score of the newly
      filled cells
17:   if  $S_{rank} < S_{best} - T_{termination}$  then           $\triangleright$  have fallen too far below best score, stop
      extension
18:      $STOP \leftarrow TRUE$  return  $S_{best}, loc_{best}$ 
19:   else if  $S_{rank} \geq S_{best}$  then                       $\triangleright$  new best score, update and continue
20:      $S_{best} \leftarrow S_{rank}$ 
21:      $loc_{best} \leftarrow loc_{rank}$ 
22:      $q \leftarrow q + 1$ 
23:      $r \leftarrow r + 1$ 
24:   else                                               $\triangleright$  intermediate score, record miss and continue
25:      $misses \leftarrow misses + 1$ 
26:     if  $misses > misses_{max}$  then  $\triangleright$  have exceeded the maximum allowed consecutive
      steps without improving alignment, stop extension
27:        $STOP \leftarrow TRUE$  return  $S_{best}, loc_{best}$ 
28:     end if
29:      $q \leftarrow q + 1$ 
30:      $r \leftarrow r + 1$ 
31:   end if
32: end while

```

FIG. 11

12/21

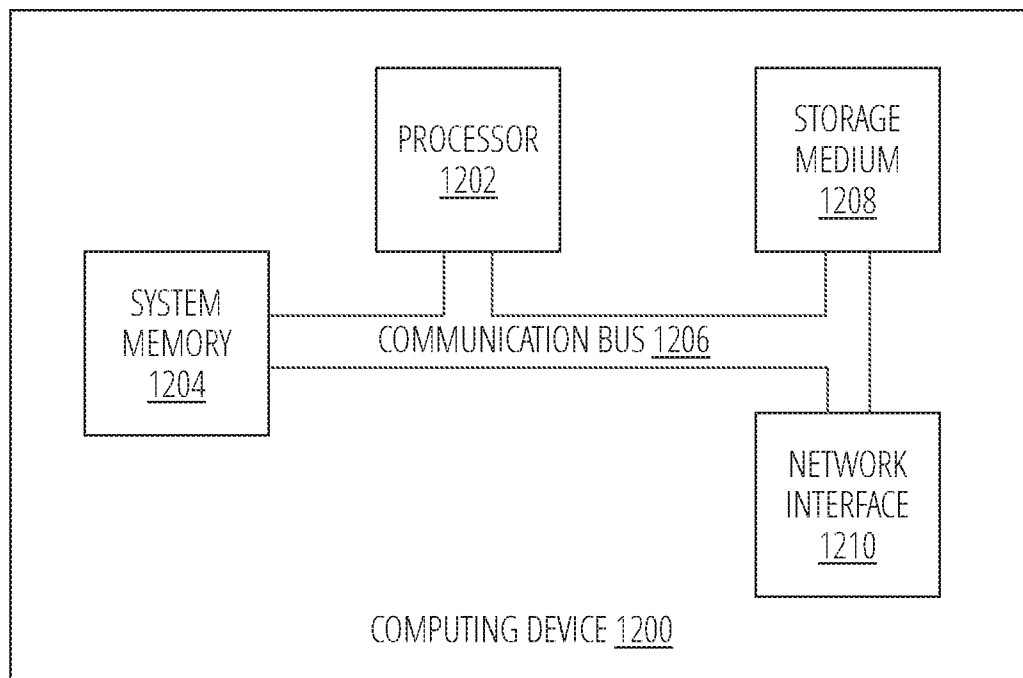


FIG. 12

13/21

Parameter Name	ssBLAST Value	iiBLAST Value
seed size (w_0)	7	7
number of bins (N_{bins})	N/A	8
match penalty (p_{match})	1	2
mismatch penalty ($p_{mismatch}$)	-1	N/A
step penalty (p_S)	0	0
skip penalty (p_K)	-1	-3
skip extension penalty (p_{K+})	-1	-1
bad penalty (p_{bad})	-10	-5
hold penalty (p_{hold})	-10	-2.5
longest skip (K_{max})	4	4
lookout (L)	7	7
max misses (M_{max})	25	25
termination threshold (T_{term})	-12.5	-7.5

FIG. 13

14/21

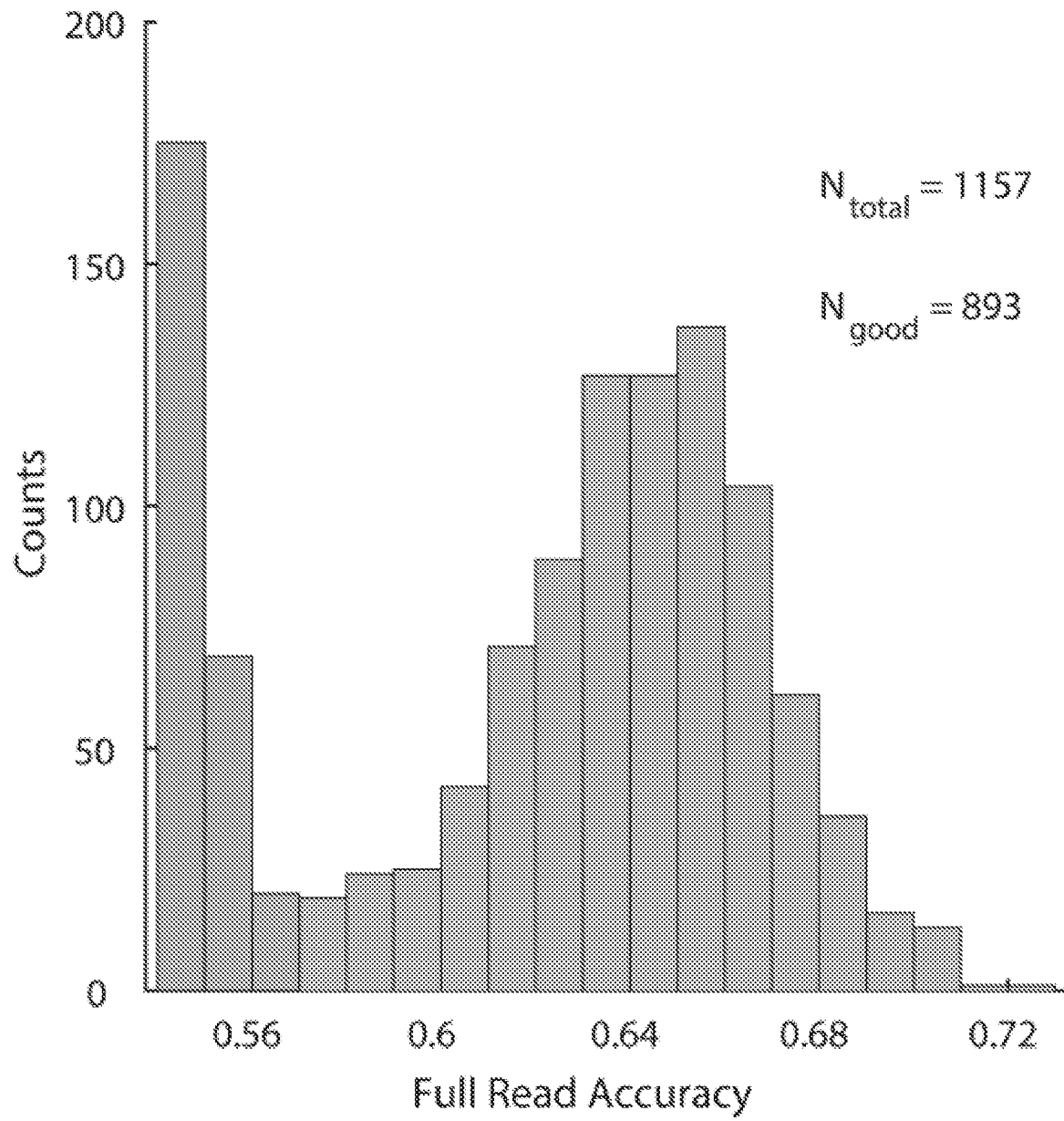


FIG. 14

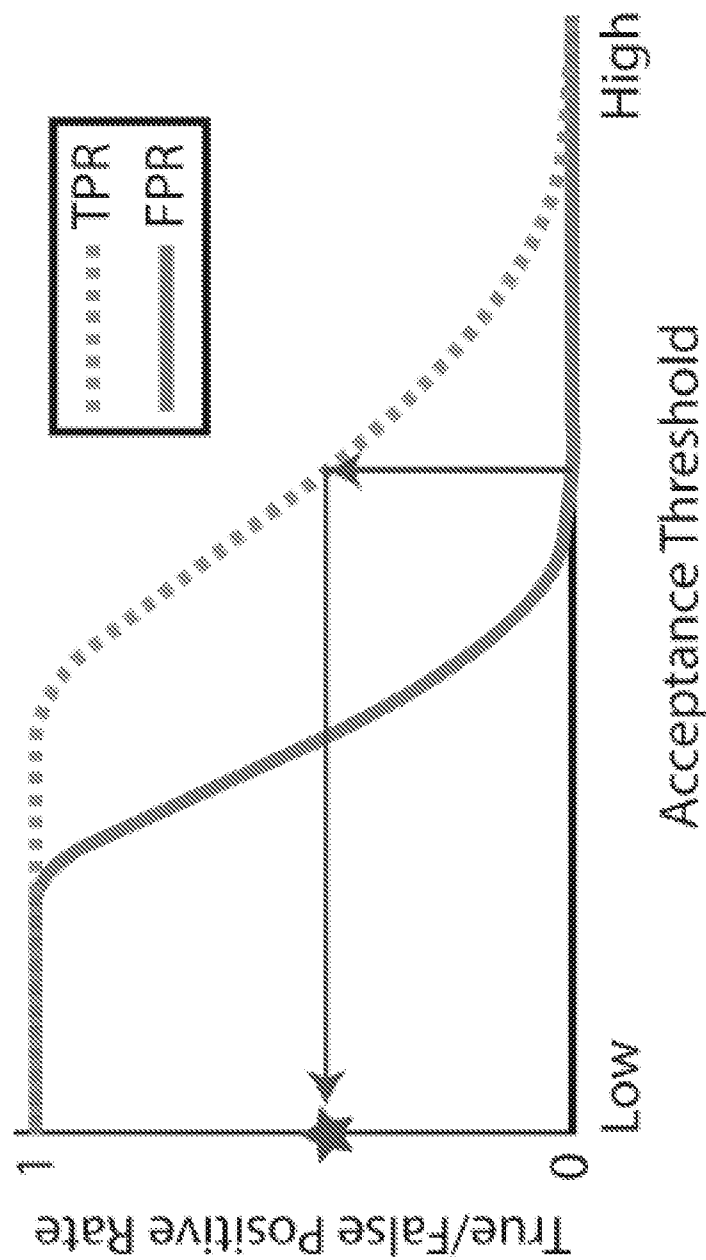


FIG. 15

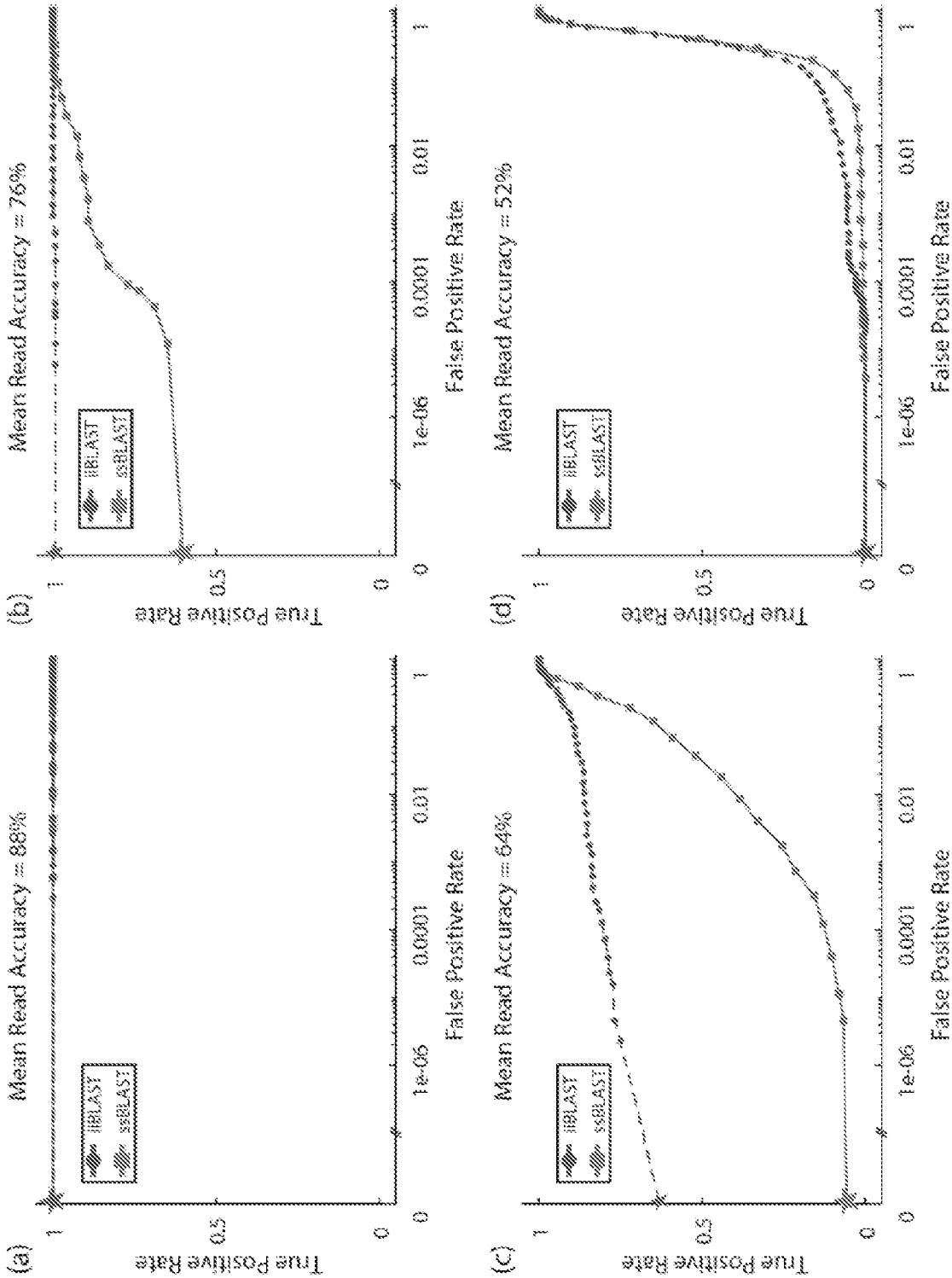


FIG. 16

17/21

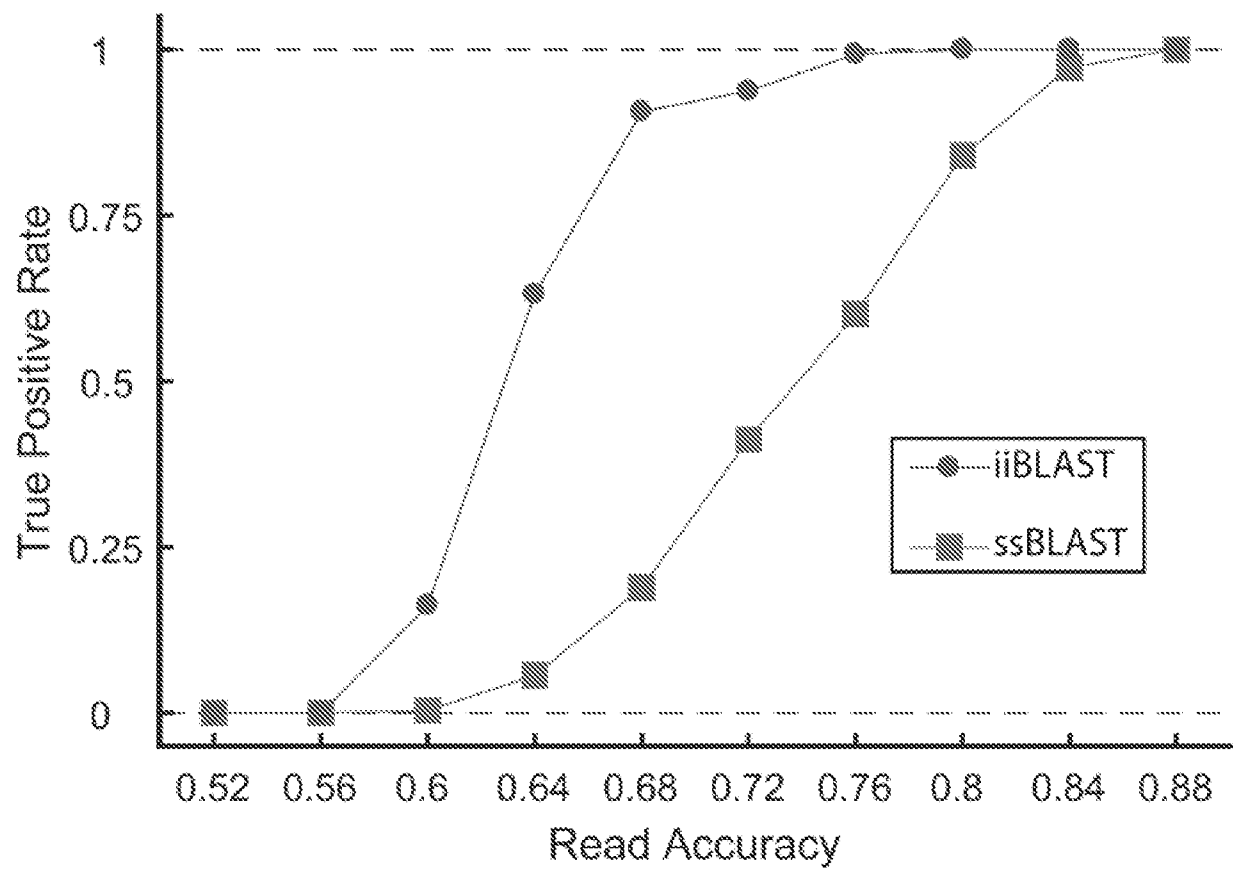


FIG. 17

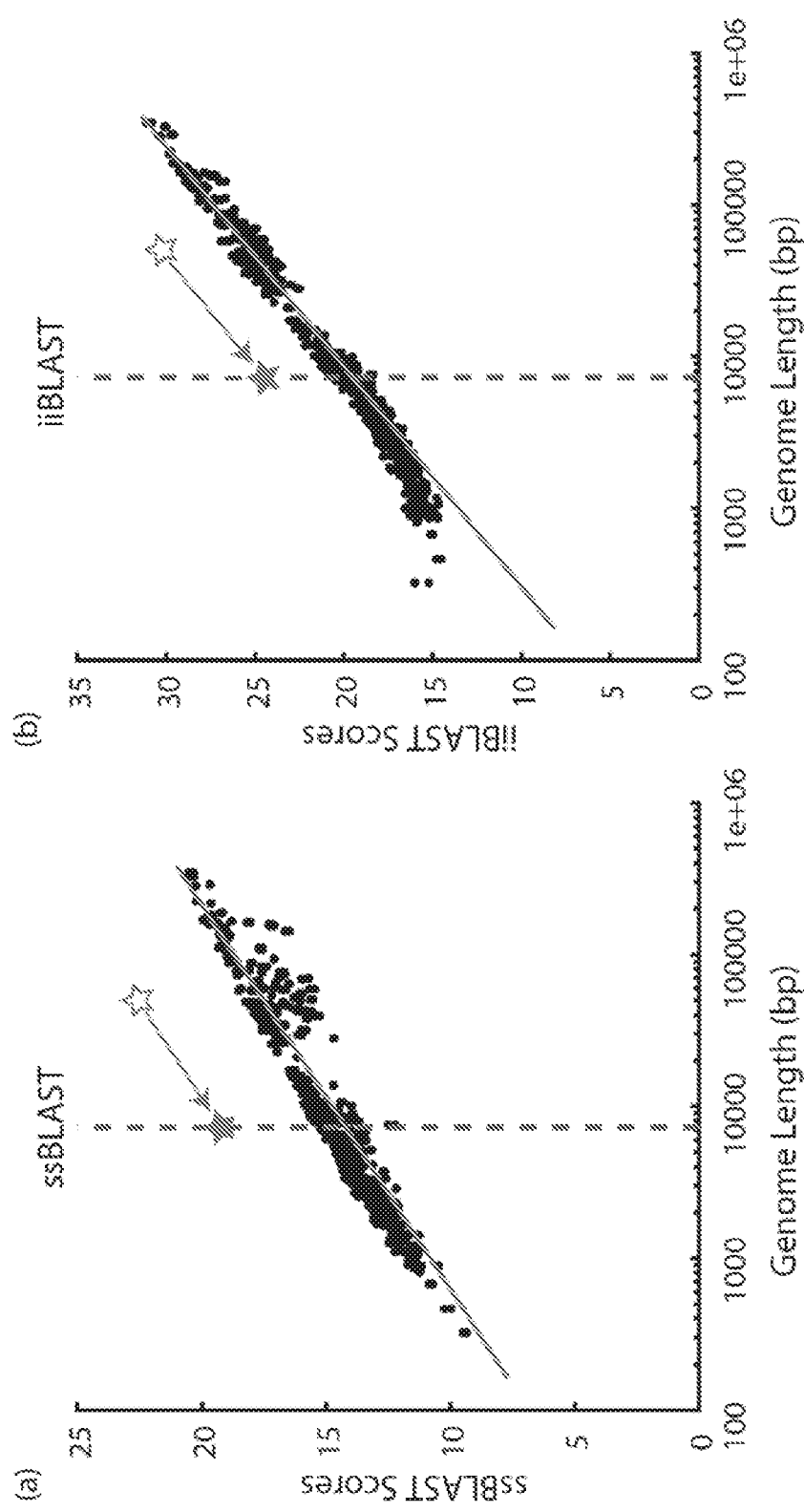


FIG. 18

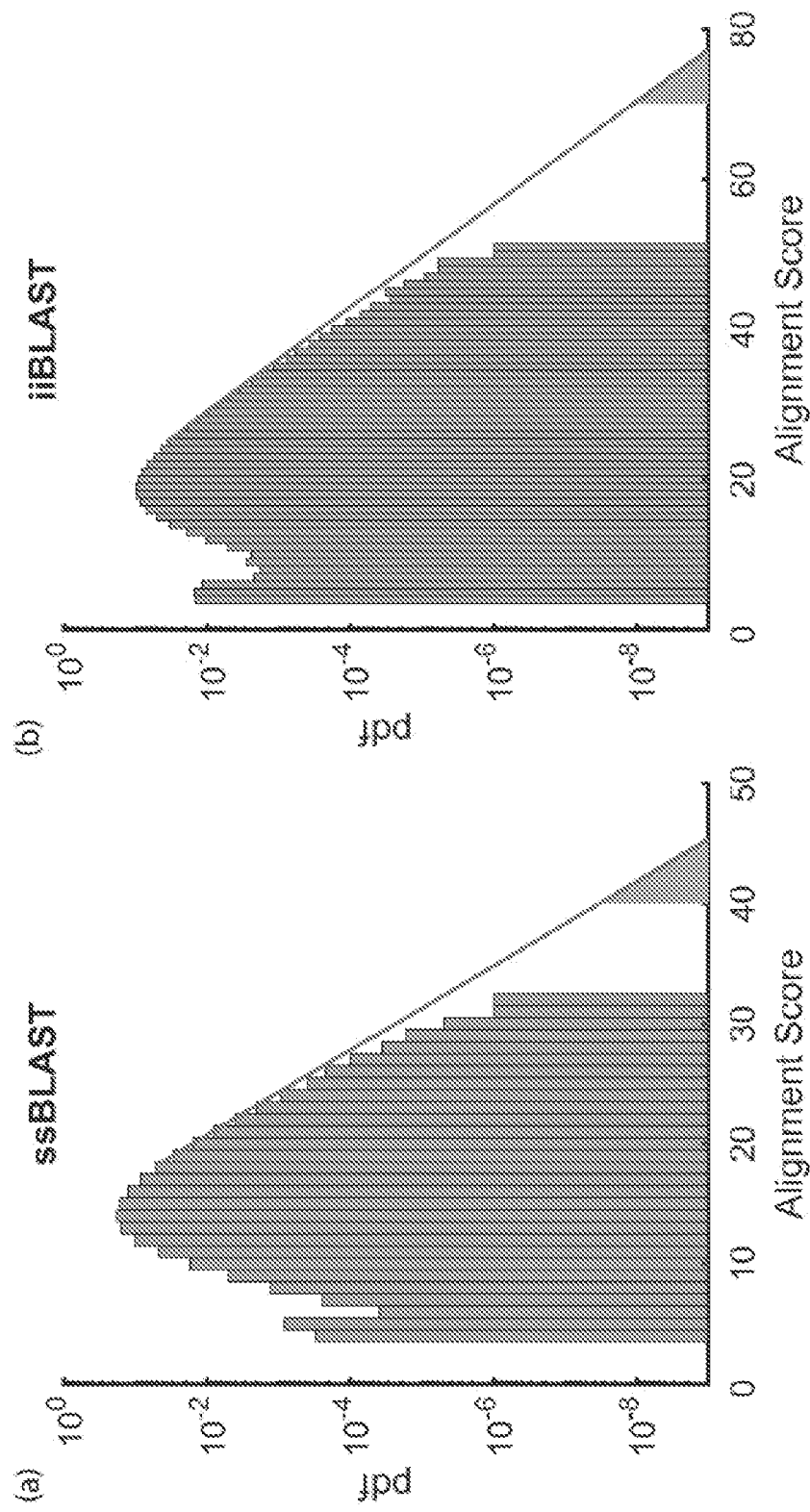


FIG. 19

Phase	iiBLAST (sec)	ssBLAST (sec)
Generate list of all seed words in the query	$9.0 * 10^{-4}$	$3.9 * 10^{-4}$
Build Mealy FSM for the seed search	0.23	0.034
Put currents into bins	0.0066	N/A
Scan database for seeds	0.45	20
Locate the found seeds in the query	0.058	0.19
Extend seeds into alignments	393.5	1787
Total	394	1807

FIG. 20

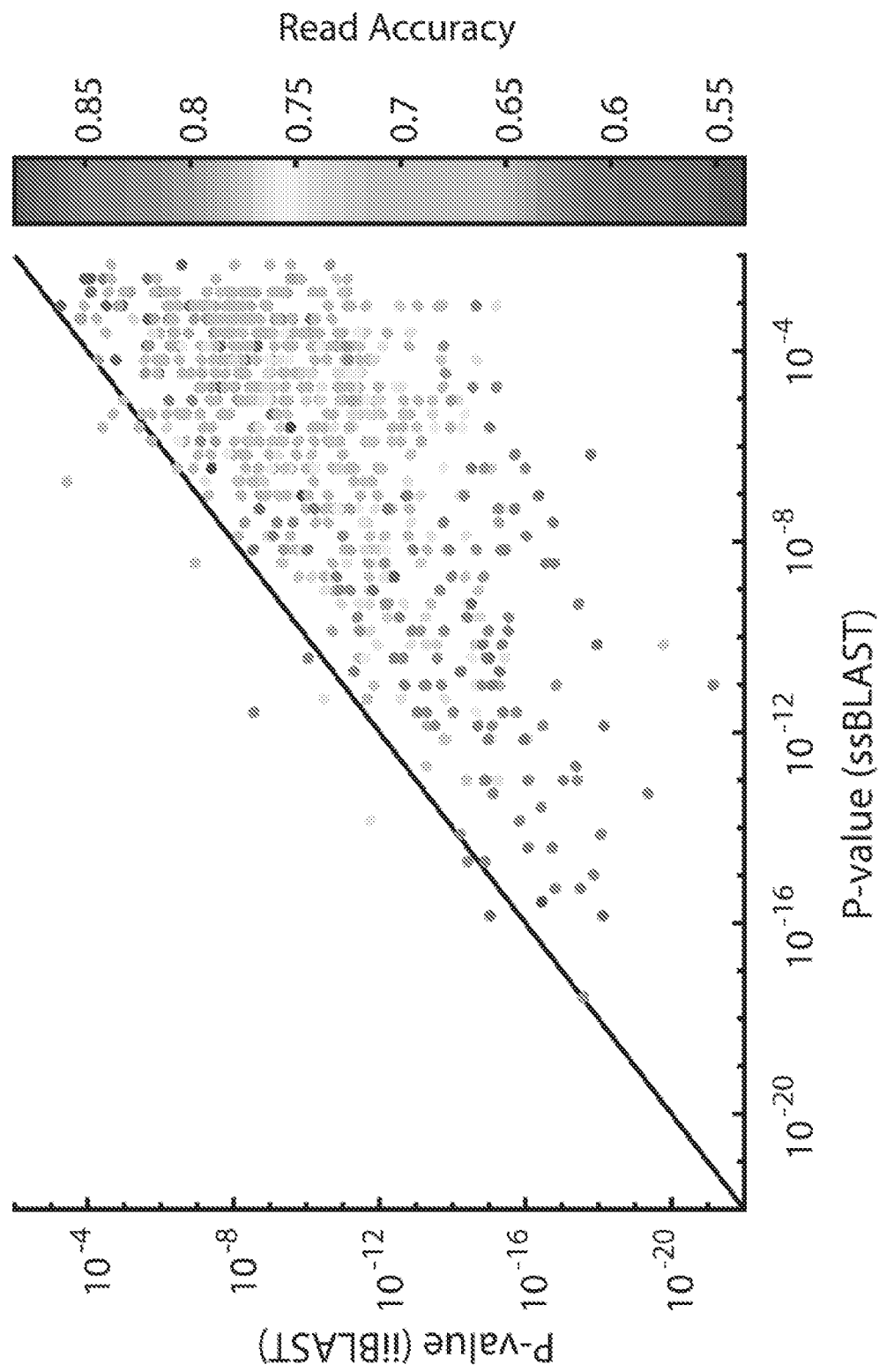


FIG. 21

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 19/65721

A. CLASSIFICATION OF SUBJECT MATTER

IPC - C12Q 1/68, C12Q 1/00 (2020.01)

CPC - C12Q 1/68, C12Q 1/6806, C12Q 1/6827, C12Q 1/6869, C12Q 1/6874, C12Q 2565/631, C12Q 2525/161, C12Q 2525/197

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2016/0259880 A1 (Seven Bridges Genomics Inc.) 08 September 2016 (08.09.2016), entire document especially Figure 15; paras [0093], [0140], [0141], [0196], [0197], [0198], [0199], [0200]	1-10 and 24-28
Y	US 2016/0306923 A1 (Edico Genome, Inc.) 20 October 2016 (20.10.2016), entire document especially paras [0609], [0610]	1-10 and 24-28
A	US 2015/0227685 A1 (Kural) 13 August 2015 (13.08.2015), entire document	1-10 and 24-28
A	US 2013/0059740 A1 (Drmanac et al.) 07 March 2013 (07.03.2013), entire document	1-10 and 24-28
A	US 2013/0217006 A1 (Pacific Biosciences of California, Inc.) 22 August 2013 (22.08.2013), entire document	1-10 and 24-28
A	US 2017/0308644 A1 (Edico Genome Corp.) 26 October 2017 (26.10.2017), entire document	1-10 and 24-28

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

26 March 2020 (26.03.2020)

Date of mailing of the international search report

21 APR 2020

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450

Facsimile No. 571-273-8300

Authorized officer

Lee Young

Telephone No. PCT Helpdesk: 571-272-4300

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 19/65721

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☒ Claims Nos.: 11-15 and 29-30
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be searched, the appropriate additional search fees must be paid.

Group I - Claims 1-10, 24-28 are directed to a method for aligning genetic sequence reads to a reference genome.

Group II - Claims 16-23 are directed to a method of finding an analog query pattern in a set of analog reference data.

-- continued on extra sheet --

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
1-10 and 24-28

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.