CLAIMS

1. A method for encoding genome sequence data, said genome sequence data comprising reads of sequences of nucleotides, said method comprising the steps of:
aligning said reads to one or more reference sequences thereby creating aligned reads,
classifying said aligned reads according to specified matching rules with said one or more reference sequences, thereby creating classes of aligned reads,
encoding said classified aligned reads as a multiplicity of blocks of descriptors, wherein descriptors encode the reads with reference to the one or more reference sequence,
wherein encoding said classified aligned reads as a multiplicity of blocks of descriptors comprises selecting said descriptors according to said classes of aligned reads,
structuring said blocks of descriptors with header information thereby creating successive Access Units.

2. The encoding method of claim 1 further comprising:
further classifying said reads that do not satisfy said specified matching rules into a class of unmapped reads,
constructing a set of reference sequences using at least some unmapped reads,
aligning said class of unmapped reads to the set of constructed reference sequences,
encoding said classified aligned reads as a multiplicity of blocks of descriptors,
encoding said set of constructed reference sequences,
structuring said blocks of descriptors and said encoded reference sequences with header information thereby creating successive Access Units.

3. The method of claim 2, wherein said classifying comprises identifying genomic reads without any mismatch in the reference sequence as first “Class P” when no mismatches are present in the mapped read with respect to the reference sequence used for mapping.

4. The method of claim 3, wherein said classifying further comprises identifying genomic reads as a second “Class N” when mismatches are only found in the positions where the sequencing machine was not able to call any “base” and the number of mismatches in each read does not exceed a given threshold.

5. The method of claim 4, wherein said classifying further comprises identifying genomic reads as a third “Class M” when mismatches are found in the positions where the sequencing machine was not able to call any “base”, named “n type” mismatches, and/or it called a different “base” than the reference sequence, named “s type” mismatches, and the number of mismatches does not exceed given thresholds for the number of mismatches of “n type”, of “s type” and a threshold obtained from a given function (f(n,s)).

6. The method of claim 5, wherein said classifying further comprises identifying genomic reads as a fourth “Class I” when they can possibly have the same type of mismatches of “Class M”, and in addition at least one mismatch of type: “insertion” (“i type”) “deletion” (“d type”) soft clips (“c type”), and wherein the number of mismatches for each type does not exceed the corresponding given threshold and a threshold provided by a given function (w(n,s,i,d,c)).

7. The method of claim 6, wherein said classifying further comprises identifying genomic reads as a fifth “Class U” as comprising all reads that do not find any classification in the Classes P, N, M, I.

8. The encoding method of claim 7 wherein the reads of the genomic sequence to be encoded are paired.

9. The method of claim 8, wherein said classifying further comprises identifying genomic reads as a sixth “Class HM” as comprising all reads pairs where one read belong to Class P, N, M or I and the other read belong to “Class U”.

10. The encoding method of claim 9 further comprising the steps of:
Identifying if the two mate reads are classified in the same class (each of: P, N, M, I, U), then assigning the pair to the same identified class,
Identifying if the two mate reads are classified in different classes, and in case none of them belongs to the “Class U”, then assigning the pair of reads to the class with the highest priority defined according to the following expression:
P < N < M < I
in which “Class P” has the lowest priority and “Class I” has the highest priority;
identifying if only one of the two mate reads has been classified as belonging to “Class U” and classifying the pair of reads as belonging to the “Class HM” sequences.

11. The method of claim 10 where each Class of reads N, M, I is further partitioned into two or more subclasses (296, 297, 298) according to a vector of thresholds (292, 293, 294) defined respectively for each class N, M and I, by the number of “n type” mismatches (292), the function f(n,s) (293) and the function w(n,s,i,d,c) (294).

12. The encoding method of claim 11 further comprising the steps of:

identifying if the two mate reads are classified in the same subclass, then assigning the pair to the same sub-class,

identifying if the two mate reads are classified into sub-classes of different Classes, then assigning the pair to the subclass belonging to the Class of higher priority according to the following expression:

$N < M < I$

where N has the lowest priority and I has the highest priority;

identifying if the two mate reads are classified in the same class, and such class is N or M or I, but in different sub-classes, then assigning the pair to the sub-class with the highest priority according to the following expressions:

$N_1 < N_2 < \ldots < N_k$

$M_1 < M_2 < \ldots < M_j$

$I_1 < I_2 < \ldots < I_h$

where the highest index has the highest priority.

13. The method of claim 12 wherein information on the mapping position of each read is encoded by means of a “pos” descriptor block.

14. method of claim 13 wherein information on the strandedness (i.e. the DNA strand the read was sequences from) of each read is encoded by means of a rcomp descriptor block.

15. The method of claim 14 wherein pairing information of paired-end reads is encoded by means of a “pair” descriptor block.

16. The method of claim 15 wherein additional alignment information such as if the read is mapped in proper pair, it fails platform/vendor quality checks, it is a PCR or optical duplicate or it is a supplementary alignment is encoded by means of a “flags” descriptor block.

17. The method of claim 16 wherein information on unknown bases is encoded by means of a “nmis” descriptor block.

18. The method of claim 17 wherein information on the position of substitutions is encoded by means of a “snpp” descriptor block.

19. The method of claim 18 wherein information on the type of substitutions is encoded by means of a specific “snpt” descriptor block.

20. The method of claim 19 wherein information on the position of mismatches of type substitutions, insertions or deletions is encoded by means of a “indp” descriptor block.

21. The method of claim 20 wherein information on the type of mismatches such as substitutions, insertions or deletions is encoded by means of a “indt” descriptor block.

22. The method of claim 21 wherein information on clipped bases of a mapped read is encoded by means of a “indc” descriptor block.

23. The method of claim 22 wherein information on unmapped reads is encoded by means of a “ureads” descriptor block.

24. The method of claim 23 wherein information on the type of reference sequence used for encoding is encoded by means of a “rtype” descriptor block.

25. The method of claim 24 wherein information on multiple alignments of the mapped reads is encoded by means of a “mmap” descriptor block.

26. The method of claim 25 wherein information on spliced alignments and multiple alignments of the same read is encoded by means of a “msar” descriptor block and a “mmap” descriptor block.

27. The method of claim 26 wherein information on read alignment scores is encoded by means of a “mscore” descriptor block.

28. The method of claim 27 wherein information on the groups reads belong to is encoded by means of a “rgroup” descriptor block.

29. The method of claim 28 wherein Access Units of class P are built using blocks of descriptors of type “pos”, “rcomp” and “flags”.

30. The method of claim 29 wherein said Access Units of class P encodes pairing information of paired-end using a block of “pair” descriptors.

31. The method of claim 30 wherein Access Units of class N are built using the same blocks of descriptors of an Access Unit of class P plus a “nmis” descriptor block for the information on the position of unknown bases.

32. The method of claim 30 wherein Access Units of class M are built using the same blocks of descriptors of Access Units of class P plus blocks of the “snpp” and “snpt” descriptors for the information on position and type of substitutions.

33. The method of claim 30 wherein Access Units of class I are built using the same blocks of descriptors of Access Units of class P plus blocks of the “indp”, “indt” and “indc” descriptors for the information on position and type of substitutions, insertions, deletions and clipped bases.

34. The method of claim 33 wherein Access Units of class HM are built using the same blocks of descriptors of Access Units of class I for the mapped reads, and using blocks of the “ureads” descriptor for the unmapped reads.

35. The method of claim 33 wherein information on multiple alignments is conveyed using blocks of the “mmap” and “msar” descriptor.

36. The method of claim 35 wherein information on spliced alignments is conveyed using an extended cigar string comprising:

- the symbol = to indicated matching bases
- the symbol + to indicate insertions
- the symbol - to indicate deletions

- the symbol / to indicate a splice on the forward strand
- the symbol % to indicate a splice on the reverse strand
- the symbol * to indicate an undirected splice
- a textual character from the IUPAC codes for DNA to indicate a substitution
- the symbol (n) to indicate n soft clipped bases where n is an integer number
- the symbol [n] to indicate n hard clipped bases where n is an integer number

37. The method of claim 36 wherein said blocks of descriptors comprise a “master index table”, containing one section for each Class and sub-class of aligned reads, said section comprising the mapping positions on said one or more reference sequences of the first read of each Access Unit of each Class or sub-class of data; jointly coding said “master index table” and said Access Unit data.

38. The method of claim 37, wherein said blocks of descriptors further comprise information on the type of reference used (pre-existing or constructed), and the segments of the read that do not map on the reference sequence.

39. The method of claim 38, wherein said reference sequences are first transformed into different reference sequences by applying substitutions, insertions, deletions and clipping, then the encoding of said classified aligned reads as a multiplicity of blocks of descriptors refers to the transformed reference sequences.

40. The method of claim 39 wherein the same transformation is applied to the reference sequences used for all classes of data.

41. The method of claim 40 where different transformations are applied to the reference sequences used for each class of data.

42. The methods of claims 41 where the reference sequences transformations are encoded as blocks of descriptors and structured with header information thereby creating successive Access Units.

43. The method of claims 42, wherein the encoding of said classified aligned reads and the related reference sequences transformations as multiplicity of blocks of descriptors comprises

the step of associating a specific source model and a specific entropy coder to each descriptor block.

44. The method of claim 43, wherein said entropy coder is one of a context adaptive arithmetic coder, a variable length coder or a golomb coder.

45 A method for decoding encoded genomic data comprising the steps of:

parsing Access Units containing said encoded genomic data to extract multiple blocks of descriptors by employing header information, wherein descriptors encode reads with reference to one or more reference sequence,

decoding said multiplicity of blocks of descriptors to extract reads according to specific matching rules defining their classification with respect to one or more reference sequences, wherein the matching rules define a lower bound of the number of unknown bases, a lower bound of a number of substitutions, a lower bound of a number of insertions, a lower bound of a number of clipped bases and a set of matching accuracy constraints that indicate upper and/or lower bounds for the number of unknown bases, number of substitutions, number of insertions and number of clipped bases, each defined per class.

46. The decoding method of claim 45 further comprising decoding a master index table containing one section for each class of reads and the associated relevant mapping positions.

47. The decoding method of claim 46 further comprising decoding information related to the type of reference used: pre-existing, transformed or constructed.

48. The decoding method of claim 47 further comprising decoding information related to one or more transformations to be applied to the pre-existing reference sequences.

49. The decoding method of claims 48 wherein said block of descriptors are entropy decoded.

50. The decoding method of claim 49 wherein:

wherein a first “Class P” of reads is defined, which includes genomic reads without any mismatch in the reference sequence, wherein no mismatches are present in the mapped read with respect to the reference sequence used for mapping,

wherein each read of the first “Class P” is obtained by decoding one or more of the following:

a “pos” descriptor block, which encodes information on the mapping position of the read;

a “rcomp” descriptor block, which encodes information on the strandedness (i.e. the DNA strand the read was sequences from) of the read;

a “flags” descriptor block, which encodes additional alignment information such as if the read is mapped in proper pair, it fails platform/vendor quality checks, it is a PCR or optical duplicate or it is a supplementary alignment;

wherein a second “Class N” of reads is defined, which includes genomic reads for which mismatches are only found in the positions where the sequencing machine was not able to call any “base” and the number of mismatches in each read does not exceed a given threshold,

wherein each read of the second “Class N” is obtained by decoding one or more of the following:

a “pos” descriptor block;

a “rcomp” descriptor block;

a “flags” descriptor block;

a “nmis” descriptor block, which encodes information on unknown bases;

wherein a third “Class M” of reads is defined, which includes genomic reads for which mismatches are found in the positions where the sequencing machine was not able to call any “base”, named “n type” mismatches, and/or it called a different “base” than the reference sequence, named “s type” mismatches, and the number of mismatches does not exceed given thresholds for the number of mismatches of “n type”, of “s type” and a threshold obtained from a given function (f(n,s)),

wherein each read of the third “Class M” is obtained by decoding one or more of the following:

a “pos” descriptor block;

a “rcomp” descriptor block;

a “flags” descriptor block;

a “snpp” descriptor block, which encodes information on the position of substitutions;
a “snpt” descriptor block, which encodes information on the type of substitutions;

5

wherein a fourth “Class I” of reads is defined, which includes genomic reads that can possibly have the same type of mismatches of “Class M”, and in addition at least one mismatch of type: “insertion” (“i type”), “deletion” (“d type”), soft clips (“c type”), and wherein the number of mismatches for each type does not exceed the corresponding given threshold and a threshold provided by a given function (w(n,s,i,d,c)),
wherein each read of the fourth “Class I” is obtained by decoding one or more of the following:
a “pos” descriptor block;
a “rcomp” descriptor block;
a “flags” descriptor block;
a “indp” descriptor block, which encodes information on the position of mismatches of type substitutions, insertions or deletions;
a “indt” descriptor block, which encodes information on the type of mismatches such as substitutions, insertions or deletions;
a “indc” descriptor block, which encodes information on clipped bases of the mapped read;

wherein a fifth “Class U” is defined, which includes all genomic reads that do not find any classification in the Classes P, N, M, I,
wherein each read of the fifth “Class U” is obtained by decoding one or more of the following:
a “pos” descriptor block;
a “rcomp” descriptor block;
a “flags” descriptor block;
a “snpp” descriptor block;
a “snpt” descriptor block;
a “indc” descriptor block;
a “ureads” descriptor block, which encodes information on unmapped reads;
a “rtype” descriptor block, which encodes information on the type of reference sequence used

for encoding.

51. The decoding method of claim 50 wherein paired reads of:

Class P, N, M and I are obtained by also decoding blocks of descriptors of type: “pair”, which encodes pairing information of paired-end reads;

Class HM are obtained by decoding one or more of the following:

a “pos” descriptor block;

a “rcomp” descriptor block;

a “flags” descriptor block;

a “indp” descriptor block;

a “indt” descriptor block;

a “indc” descriptor block;

a “ureads” descriptor block.

52. A genomic encoder (2010) for the compression of genome sequence data 209, said genome sequence data 209 comprising reads of sequences of nucleotides,

said genomic encoder (2010) comprising:

an aligner unit (201), configured to align said reads to one or more reference sequences thereby creating aligned reads,

a constructed-reference generator unit (202), configured to produce constructed reference sequences

a data classification unit (204), configured to classify said aligned reads according to specified matching rules with the one or more pre-existing reference sequences or constructed reference sequences thereby creating classes of aligned reads (208);

one or more blocks encoding units (205-207), configured to encode said classified aligned reads as blocks of descriptors by selecting said descriptors according to said classes of aligned reads,

a multiplexer (2016) for multiplexing the compressed genomic data and metadata.

53. The genomic encoder of claim 52 further comprising

a reference sequence transformation unit (2019) configured to transform the pre-existing references and data classes (208) into transformed data classes (2018).

54. The genomic encoder of claim 53 where the data classification unit (204) contains encoders of data classes N, M and I configured with vectors of thresholds generating sub-classes of data classes N, M and I.

55. The genomic encoder of claim 54, wherein the reference transformation unit (2019) applies the same reference transformation (300) for all classes and sub-classes of data.

56. The genomic encoder of claim 54, wherein the reference transformation unit (2019) applies different reference transformations (301, 302, 303) for the different classes and sub-classes of data.

57. A genomic decoder (218) for the decompression of a compressed genomic stream (211) said genomic decoder (218) comprising:

a demultiplexer (210) for demultiplexing compressed genomic data and metadata,

parsing means (212-214) configured to parse said compressed genomic stream into genomic blocks of descriptors (215),

one or more block decoders (216-217), configured to decode the genomic blocks of descriptors into classified reads of sequences of nucleotides (2111),

genomic data classes decoders (219) configured to selectively decode said classified reads of sequences, of nucleotides on one or more reference sequences so as to produce uncompressed reads of sequences of nucleotides.

58. The genomic decoder of claim 57 further comprising a reference transformation decoder (2113) configured to decode reference transformation descriptors (2112) and produce a transformed reference (2114) to be used by genomic data class decoders (219).

59. The genomic decoder of claims 58, wherein the one or more reference sequences are stored in the compressed genome stream (211).

60. The genomic decoder of claim 58, wherein the one or more reference sequences are provided to the decoder via an out of band mechanism.

61. The genomic decoder of claim 58, wherein the one or more reference sequences are built at the decoder.

62. The genomic decoder of claim 58, wherein one or more reference sequences are transformed at the decoder by a reference transformation decoder (2113).

63. A computer-readable medium comprising instructions that when executed cause at least one processor to perform the encoding method of claim 12.

64. A computer-readable medium comprising instructions that when executed cause at least one processor to perform the decoding method of claim 58.