

GENE IDENTIFICATION SIGNATURE (GIS) ANALYSIS FOR TRANSCRIPT MAPPING**Field Of Invention**

The present invention relates generally to a transcript mapping method. In particular,
5 the invention relates to a transcript mapping method for mapping from a transcript to
a compressed suffix array indexed genome sequence.

Background

Since the completion of the genome sequences for human and several other
10 organisms, attention has been drawn towards annotation of genomes for functional
elements including gene coding transcript units and regulatory cis-acting elements
that modulate gene expression levels.

Currently there are three main approaches for genome annotation. The first approach
15 uses existing transcript data to identify gene-coding regions in the genomes, the
second approach uses computational algorithms to statistically predict genes and
regulatory elements and the third approach compares genomic sequences from other
vertebrates for conserved regions based on the view that functional elements in
genomes are conserved during evolution.

20 Despite considerable success, these approaches are unsatisfactory for determining the
complete and precise content of all functional elements in the human genome. As a
result, a complete list of genes in the human genome is still unavailable. In particular,
all the low abundant and cell specific genes have not been identified. Many gene
25 models suggest that the current genome annotation is incorrect, particularly regarding
where the transcription starts and ends.

All the gene predictions have to be validated by experimental means and the
prospective genes are required to be cloned in full-length for further functional
30 studies. It is therefore clear that many challenges surround the field of human
genome annotation.

One of the challenges is the identification of all genes and all transcripts expressed from the genes in human and model organisms. In the annotation of genes, full-length cDNA cloning and sequencing is the most conclusive and is viewed as the gold standard for the analysis of transcripts. However, this approach is expensive and slow when applied to a large number of transcripts across a large number of species and biological conditions. There are short tag based approaches such as SAGE (serial analysis of gene expression) and MPSS (massively parallel signature sequence). These short tag based methods extract a 14-20bp signature for representing each transcript. Though this approach is efficient in tagging and counting transcripts in a given transcriptome, the specificity of the tags is often poor and the information yielded regarding transcript structures are frequently incomplete and ambiguous.

Gene Identification Signature (GIS) ditag sequences, obtained by extracting interlinked 5' and 3' ends of full-length cDNA clones into a ditag structure, provide substantial tag specificity. However, there are no existing computer algorithms that are readily applicable for mapping the GISditag sequences to genome. In the past, SAGE and MPSS tags were analyzed using a two-step approach. The tags were first matched to cDNA sequences and then to the genome. In this approach, novel transcripts that did not exist in cDNA databases would not be mapped. The two most often used sequence alignment tools, BLAST (basic local alignment search tool) and BLAT (BLAST-like alignment tool), are not designed for short tag sequences and often leads to poor or incorrect results.

Hence, this clearly affirms a need for an improved transcript mapping method.

25

Summary

A transcript mapping method according to an embodiment of the invention is described hereinafter and combines short tag based (SAGE and MPSS) efficiency with the accuracy of full-length cDNA (flcDNA) for comprehensive characterization of transcriptomes. This method is also referred to as Gene Identification Signature (GIS) analysis. In this method, the 5' and 3' ends of full-length cDNA clones are initially extracted into a ditag structure, with the ditag concatemers of the ditag being subsequently sequenced in an efficient manner, and finally mapped to the genome for

30

defining the gene structure. In the GIS analysis, each sequence read reveals approximately 15 ditags representing 15 transcripts. This approach increases efficiency by at least 30-folds for identifying and quantifying full-length transcripts compared to the current flcDNA cloning and sequencing approaches. Because each
 5 GISditag sequence contains 36 base pairs (bp) to represent the beginning and the end of a transcript, the specificity of mapping tag-to-genome is greatly increased compared to the 14-21bp SAGE and MPSS tags. In addition, as a GISditag represents the 5' and 3' ends of a transcript, it is more informative than SAGE and MPSS tags.

- 10 To accommodate the GISditag data, a Suffix Array based Tag to Genome (SAT2G) algorithm is used for mapping the GISditag sequences to a genome sequence built and indexed on an advanced data structure Compressed Suffix Array (CSA).

Therefore, in accordance with a first aspect of the invention, there is disclosed a
 15 transcript mapping method comprising the steps of:

- obtaining a 5' terminal tag and a 3' terminal tag from a transcript of a gene;
- matching the 5' terminal tag to at least a portion of a genome sequence to thereby identify at least one 5' site therefrom, each of the at least one 5' site having a sequence matching the 5' terminal tag;
- 20 matching the 3' terminal tag to at least a portion of the genome sequence to thereby identify at least one 3' site therefrom, each of the at least one 3' site having a sequence matching the 3' terminal tag;
- determining quantity disparity between the at least one 5' site and the at least one 3' site;
- 25 identifying at least one occurring segment, each of the at least one occurring segment being a sequence segment along the genome sequence extending from one of the at least one 5' site towards one of the at least one 3' site, each of the at least one occurring segment having a sequence length;
- identifying at least one feasible gene location, each of the feasible gene
 30 location being one of the at least one occurring segment having a sequence length not exceeding that of a predefined gene length; and
- generating a data structure for indexing the genome sequence.

In accordance with a second aspect of the invention, there is disclosed a computer
 35 program storage device tangibly embodying a program of instructions, the program of

instructions being readable by a computing device, the program of instructions when executed by the computing device causes the computing device to:

obtain a 5' terminal tag and a 3' terminal tag from a transcript of a gene;

match the 5' terminal tag to at least a portion of a genome sequence to thereby identify at least one 5' site therefrom, each of the at least one 5' site having a sequence matching the 5' terminal tag;

match the 3' terminal tag to at least a portion of the genome sequence to thereby identify at least one 3' site therefrom, each of the at least one 3' site having a sequence matching the 3' terminal tag;

determine quantity disparity between the at least one 5' site and the at least one 3' site;

identify at least one occurring segment, each of the at least one occurring segment being a sequence segment along the genome sequence extending from one of the at least one 5' site towards one of the at least one 3' site, each of the at least one occurring segment having a sequence length;

identify at least one feasible gene location, each of the feasible gene location being one of the at least one occurring segment having a sequence length not exceeding that of a predefined gene length; and

generate a data structure for indexing the genome sequence.

In accordance with a third aspect of the invention, there is disclosed a transcript mapping method comprising the steps of:

obtaining a 5' terminal tag and a 3' terminal tag from a transcript of a gene;

matching the 5' terminal tag to at least a portion of a genome sequence to thereby identify at least one 5' site therefrom, each of the at least one 5' site having a sequence matching the 5' terminal tag;

matching the 3' terminal tag to at least a portion of the genome sequence to thereby identify at least one 3' site therefrom, each of the at least one 3' site having a sequence matching the 3' terminal tag;

determining quantity disparity between the at least one 5' site and the at least one 3' site;

identifying at least one occurring segment, each of the at least one occurring segment being a sequence segment along the genome sequence extending from one of

the at least one 5' site towards one of the at least one 3' site, each of the at least one occurring segment having a sequence length; and

identifying at least one feasible gene location from the at least one occurring segment, each of the at least one feasible gene location being one of the at least one occurring segment with at least one of the sequence length thereof not exceeding that of the predefined gene length, the sequence order thereof and of the at least one 5' site

5

and one of the at least one 3' site corresponding thereto in accordance with a 5'-occurring segment-3' structure matching the sequence order of the corresponding portion of the genome sequence, the 5' site and one of the at least one 5' site and one of the at least one 3' site corresponding thereto having a 5'-3' orientation, and one of
5 the at least one 5' site and one of the at least one 3' site corresponding to each of the occurring segment being located within the same chromosome.

Brief Description Of The Drawings

Embodiments of the invention are described hereinafter with reference to the
10 following drawings, in which:

FIG. 1 shows a schematic diagram of a 5' and 3' terminal tags SAGE technique for use in genome annotation;

15 FIG. 2 shows a process flow chart of a transcript mapping method according to an embodiment of the invention;

FIG. 3 shows a schematic diagram of a GIS ditag for application of the transcript mapping technique of FIG. 2 thereto;

20

FIG. 4 shows a pseudo code "Find_Sites" of the transcript mapping method of FIG. 2 for forward and reverse searching of 5' sites and 3' sites from a genome sequence; and

25 FIG. 5 shows a pseudo code "Match_sites_1" of the transcript mapping method of FIG. 2 for identifying the sequence length of an occurring segment, the sequence length being subsequently compared with a predefined length for identifying of a feasible gene location therefrom; and

30 FIG. 6 shows a pseudo code "Match_sites_2" of the transcript mapping method of FIG. 2 for identifying an occurring segment when a disparity condition is met wherefrom a feasible gene location is subsequently obtained.

Detailed Description

A transcript mapping method is described hereinafter for addressing the foregoing problems.

5

Complete genome annotation relies on precise identification of transcription units bounded by a transcription initiation site (TIS) and a polyadenylation site (PAS). To facilitate this, a pair of complementary methods, namely 5'LongSAGE (long serial analysis of gene expression) and 3'LongSAGE, was developed. These methods are based on the original SAGE (serial analysis of gene expression) and LongSAGE methods that utilize typical full-length cDNA cloning technologies to enable high-throughput extraction of the first and the last 20 base pairs (bp) of each transcript. Mapping of 5' and 3' LongSAGE tags to the genome allows the localization of the TIS and the PAS.

15

However, matching of 5' and 3' tags derived from same transcripts in genome sequences are not always straightforward and can sometimes be very ambiguous. A solution is to clone the 5' and 3' tags of the same transcript by inter-linking the 5' and 3' tags. To achieve this, a specially designed device comprising cloning adapters and a vector links the 5' tag and the 3' tag derived from the same transcript into a ditag.

20

A plurality of ditags can be concatenated for cloning and sequencing, with each ditag representing an individual transcript. Unlike single tag sequences, the paired ditag sequences can be specifically multiplied with a frame of transcripts being precisely definable when being mapped to the genome sequences. This approach, named Gene Identification Signature (GIS) analysis, can accurately map the 5' and 3' ends of transcription units encoded by genes as shown in FIG. 1.

25

In the GIS analysis, the conventional cap-trapper method is applied to enrich a full-length cDNA and incorporated adapter sequences that bear an MmeI restriction site at each end of the cDNA fragments. The cDNA fragments are then cloned in a cloning vector to construct a GIS flcDNA (full-length cDNA) library. The plasmid prepared from the library is digested by MmeI (a type II restriction enzyme) and cleaved 20bp

30

downstream of its binding site. After digestion, the flcDNA inserts of the library were dropped from the plasmid to leave 18bp signatures of 5' and 3' ends with the learned cloning vector. Re-circling the vector would create a GIS single ditag library. The ditags of the library were then sliced out and purified for concatenating and cloning to
5 generate the final GIS ditag library for sequence analysis. Typically, each sequence read of the GIS ditag clones reveals 15 ditags. Each unit of the ditag sequence contains 18bp of 5' and 18bp of 3' signatures with a 12bp spacer to separate one ditag sequence from another.

10 Mapping ditags to the genome is akin to searching occurrences of a pattern in the genome sequence. Approaches for pattern searching include the conventional BLAST (basic local alignment search tool) and BLAT (BLAST-like alignment tool) method. Both the BLAST and BLAT methods are very slow because each thereof requires a pattern to be searched by scanning through the whole genome. Moreover,
15 conventional full-text indexing is usually employed if exact occurrences of a pattern with a small mismatch margin are required. Efficient full-text indexing data-structures include a suffix tree and a suffix array.

A suffix tree is a tree-like data-structure having branches stemming from a root with
20 each branch terminating at a leaf that encodes a suffix of the genome sequence. The suffix array is a sorted sequence of all suffices of the genome according to lexicographic order. The suffix array is represented as an array $SA[i]$ where $i=1\dots n$ and that $SA[i] = j$ means that the j -suffix (suffix starting from character j) is the i -th smallest suffix in the lexicographic order.

25 Both the suffix tree and the suffix array allow for fast pattern searching. Given a pattern of length x , its occurrences in the genome $G[1\dots n]$ can be reported in $O(x)$ time and $O(x \log n)$ time for the suffix tree and the suffix array respectively. Although the query time is fast, it is not always feasible to build the suffix tree or the
30 suffix array due to large space requirements thereof. For example, for a mouse genome, the suffix tree and the suffix array require 40 Gigabytes (GB) and 13 GB respectively. Such memory requirement far exceeds the memory space capacity of ordinary computers. To solve the memory space problem, we apply the space-

efficient compressed suffix array (CSA) indexing data structure. CSA is a compressed form of the suffix array. It can be built efficiently without need for enormous memory requirements using known algorithms. Also, the built CSA is very small. For example, a CSA for the mouse genome (mm3) occupies approximately 1.3 GB. Additionally, CSA is also able to support searching efficiently. Searching a pattern of length x requires only $O(x \log n)$ time.

A first embodiment of the invention, a transcript mapping method 20 is described with reference to FIG. 2, which shows a process flow chart of the transcript mapping method 100. The transcript mapping method 100 is for application to a transcript obtained from a gene. The transcript mapping method 100 is preferably implemented using a computer-based system. In a step 110 of the transcript mapping method 100, a 5' terminal tag 24 and a 3' terminal tag 26 are obtained from the transcript.

In combination, the 5' terminal tag 24 and the 3' terminal tag 26 forms a GIS ditag 30 as described above and as shown in FIG. 3. The GIS ditag 30 has a ditag length 32 of 36bp with 18bp nucleotide sequence being derived from the 5' terminal tag 24 and another 18bp of nucleotide sequence being derived from the 3' terminal tag 26. Due to some enzymatic variations during molecular cloning, the ditag length 32 of the GIS ditag 30 may vary from 34bp to 38bp.

This variation often occurs proximate to the extremities of the 5' terminal tag 24 and the 3' terminal tag 26 with the internal nucleotides remaining structurally conserved. In the 3' terminal tag 26, two residual nucleotides 34 (AA) are retained during poly-A tail removal therefrom. The AA residual nucleotides 34 are subsequently for use as an orientation indicator. Therefore, only 16 bp of the 3' terminal tag 26 in the GIS ditag 30 is usable for mapping to a genome sequence 36.

Following the step 110, each of the 5' terminal tag 24 and the 3' terminal tag 26 is matched to the genome sequence 36 in a step 112. In the step 112, 5' sites 38 and 3' sites 40 are identified when the 5' terminal tag 24 and the 3' terminal tag 26 are respectively matched to the genome sequence 36. Each of the 5' sites 38 and each of

the 3' sites 40 is a portion of the genome sequence 36 that has a sequence that substantially matches the 5' terminal tag 24 and the 3' terminal tag 26 respectively.

In a step 114, at least one occurring segment 42 is identified from the genome sequence 36. Each of the at least one occurring segment 42 is a sequence segment along the genome sequence 36 situated between one 5' site 38 and one 3' site 40. Each of the at least one occurring segment 42 has a sequence length 44.

Given the GIS ditag 30 (P) for the transcript (R), the computational problem of locating R in the genome sequence 36 (G) is referred to as a transcript location identification problem. Therefore, given $G[1...n]$ and $P[1...m]$, the occurring segment 42 is identified as being a feasible gene location of P when: the sequence length 44 ($j-i$) is smaller than the predefined gene length (*maxlength*), which is typically less than 1 million base pairs in length for known genes; the 5' terminal tag 24 and the 3' terminal tag 26 are longer than predefined *minlength₅* and *minlength₃* respectively (where *minlength₅* = 16 bp and *minlength₃* = 14 bp); and the 5' terminal tag 24 and the 3' terminal tag 26 of R are the substrings of $P[1...boundary_5]$ and $P[boundary_3...m]$ respectively (where *boundary₅* = 19 and *boundary₃* = 18).

The genome sequence 36 is preferably indexed using a compressed suffix array (CSA). The 5' terminal tag 24 and the 3' terminal tag are matched to the genome sequence 36 preferably by applying binary search to the compressed suffix array. The binary search for matching the 5' terminal tag 24 and the 3' terminal tag 26 are dependent on two lemmas, namely, lemma 1 for performing a forward search on the compressed suffix array and lemma 2 for performing a reverse search on the compressed suffix array.

Lemma 1 (forward search): given the CSA for the genome $G[1..n]$ and a set of occurrences of a pattern Q in G, for any base $c \in \{\text{adenine (A), cytosine (C), guanine (G), thymine (T)}\}$, a set of occurrences of the pattern Qc is obtainable in $O(\log n)$ time. A forward binary search is achieved by modifying a conventional binary search algorithm to use values in the compressed suffix array and suffix array instead of

explicit text for the suffixes in the genome sequence 36 when comparing with pattern Q in the binary search.

Lemma 2 (reverse search): given the CSA for the genome $G[1...n]$ and a set of
5 occurrences of a pattern Q in G, for any base $c \in \{A, C, G, T\}$, we can find the set of occurrences of the pattern cQ using $O(\log n)$ time.

The pseudo code "Find_Sites" for both the forward search and the reverse search is shown in Fig. 4. Instead of applying both the forward search and the reverse search in
10 tandem in the step 114, an alternative approach is to apply either only the forward search using lemma 1 or only the reverse search using lemma 2 to the genome sequence 36 for identifying the at least one occurring segment 42.

The GIS ditag 30 may appear in the genome sequence 36 in sense or anti-sense. To
15 address this issue, an index is created for each of the sense genome sequence and the anti-sense genome sequence. Instead of creating two separated indexing arrays, an anti-sense GIS ditag is created. The suffix array is searched twice in the step 110 for each of the 5' terminal tag 24 and the 3' terminal tag 26, once using the sense GIS ditag 30 and a second time using the anti-sense GIS ditag (not shown).

20 Additionally, the genome sequence 36 is naturally partitioned into chromosomes. This enables a compressed suffix array to be created for the sequence segment of each chromosome. By doing so, 5' sites 38 and 3' sites 40 are obtainable for specific chromosomes instead of the entire genome sequence 36.

25 Besides the compressed suffix array, a suffix array, a suffix tree, a binary or the like indexing data structure is usable for indexing the genome sequence 36 as abovementioned.

30 Following the step 114, the 5' sites 38 and the 3' sites 40 undergo a series of checks to identify a feasible gene location. The checks comprises of length, locality, orientation and ordering checks.

In a step 116, the length check is performed by comparing the sequence length 44 of each of the at least one occurring segment 42 with a predefined gene length in a step 116. Initially, the 5' sites 38 and 3' sites 40 are sorted preferably in an ascending order. Next, each of the at least one occurring segment 42 has a sequence length 44
5 that does not exceed that of the predefined gene length (*maxlength*) is identified as a potential feasible gene location. The pseudo code "Match_sites_1" for step 116 is shown in Fig. 5.

In a step 118, the locality check is performed whereby the 5' site 38 and the 3' site 40
10 corresponding to each of the at least one occurring segment 42 are analysed to identify which chromosome each of them are located within. The occurring segment 42 identifies a potential feasible gene location only when the 5' site 38 and the 3' site 40 thereof belongs to the same chromosome.

In a step 120, the orientation check is performed by identifying the orientation of the
15 5' site 38 and the 3' site 40 that corresponds to each occurring segment 42. The orientation of the 5' site 38 and the 3' site is identifiable by locating the position of the residual nucleotide 34. Preferably, the 5' site 38 and the 3' site 40 should have a 5'-3' orientation for the occurring segment 42 thereof to identify a potential feasible
20 gene location.

In a step 122, the ordering check is performed by comparing each of the occurring segments 42 and the corresponding 5' site 38 and 3' site 40 to the genome sequence 36. Preferably, the ordering of each of the occurring segments 42 and its
25 corresponding 5' site 38 and 3' site 40 should follow a 5'-occurring segment-3' structure for it to be a potential feasible site.

Steps 116-122 of the transcript mapping method can occur in any sequence in combination or independently.

30

In a situation where the feasible gene location is not found from the GIS ditag 30, the constraints are relaxed to allow at least one mismatch when matching the 3' terminal tag 26 to the genome sequence 36 in the step 112.

Alternatively, the quantity of the 5' sites 38 and the quantity of the 3' sites 40 are initially obtained before the 5' sites 38 and the 3' sites 40 are matched to the genome sequence 36 in the step 112. This enables identification of quantity disparity between the 5' sites 38 and the 3' sites 40, for example, when there only exist less than ten of the 5' sites 38 and more than tens of thousand of the 3' sites 40, or vice versa.

When large quantity disparity between the 5' sites 38 and the 3' sites 40 exists, the transcript mapping method 20 undergoes multiple iterations of redundant mapping to the genome sequence 36. Therefore, a modified approach is required for the transcript mapping method 100 when a large quantity disparity arises. To identify the quantity disparity, a disparity condition is established as:

$$\frac{1}{threshold_{5,3}} \geq \frac{count_5}{count_3} \geq threshold_{5,3}$$

15

where $count_5$ is the quantity of 5' sites 38, $count_3$ is the quantity of 3' sites 40, and $threshold_{5,3}$ is a pre-defined threshold, for example $threshold_{5,3} = 10,000$, for limiting the quantitative disparity between $count_5$ and $count_3$. The CSA enables both $count_5$ and $count_3$ to be obtained without enumerating either any of the 5' sites 38 or any of the 3' sites.

20

The method described in the pseudo code "Match_sites_2" of FIG. 6 is applied when the above disparity condition is met. In the pseudo code "Match_sites_2", the number of iterations required for mapping to the genome sequence 36 is determined by the smaller one of $count_5$ and $count_3$. For example, should there be only two 5' sites 38, the mapping to or traversal along the genome sequence 36 for obtaining the corresponding one of the 3' sites 40 is only iterated twice, once for each of the two 5' sites 38, for obtaining the occurring segments 42 therefrom.

25

However, should the above disparity condition be unmet, the quantity disparity between $count_5$ and $count_3$ is not large and therefore the transcript mapping method

30

100 reverts to the method described in “Match_sites_1” for obtaining the occurring segments 42.

5 In the foregoing manner, a transcript mapping method is described according to one embodiment of the invention for addressing the foregoing disadvantages of conventional mapping methods. Although only one embodiment of the invention are disclosed, it will be apparent to one skilled in the art in view of this disclosure that numerous changes and/or modification can be made without departing from the scope and spirit of the invention.

Claims

1. A transcript mapping method comprising the steps of:
 - obtaining a 5' terminal tag and a 3' terminal tag from a transcript of a gene;
 - 5 matching the 5' terminal tag to at least a portion of a genome sequence to thereby identify at least one 5' site therefrom, each of the at least one 5' site having a sequence matching the 5' terminal tag;
 - matching the 3' terminal tag to at least a portion of the genome sequence to thereby identify at least one 3' site therefrom, each of the at least one 3' site having a sequence matching the 3' terminal tag;
 - 10 determining quantity disparity between the at least one 5' site and the at least one 3' site;
 - identifying at least one occurring segment, each of the at least one occurring segment being a sequence segment along the genome sequence extending from one of the at least one 5' site towards one of the at least one 3' site, each of the at least one occurring segment having a sequence length;
 - 15 identifying at least one feasible gene location, each of the feasible gene location being one of the at least one occurring segment having a sequence length not exceeding that of a predefined gene length; and
 - 20 generating a data structure for indexing the genome sequence.
2. The transcript mapping method as in claim 1, the step of identifying a 5' terminal tag and a 3' terminal tag comprising the step of:
 - providing a nucleotide sequence with at least 16 base pairs for forming
 - 25 the 5' terminal tag; and
 - providing a nucleotide sequence with at least 16 base pairs for forming the 3' terminal tag.
3. The transcript mapping method as in claim 1, the step of matching the 5' terminal tag to at least a portion of a genome sequence comprising the step of:
 - 30 matching the 5' terminal tag to a chromosome sequence.
4. The transcript mapping method as in claim 3, the step of matching the 3' terminal tag to at least a portion of the genome sequence comprising the step of:
 - 35

matching the 3' terminal tag to a chromosome sequence.

5. The transcript mapping method as in claims 1, the step of generating a data structure for indexing the genome comprising the step of:
5 generating at least one of a tree structure and an ordered array for indexing the genome sequence.
6. The transcript mapping method as in claims 5, further comprising the step of generating at least one of a suffix array, a suffix tree, a binary tree and a
10 compressed suffix array for indexing the genome sequence.
7. The transcript mapping method as in claim 6, the step of the matching the 5' terminal tag to at least a portion of a genome sequence comprising the step of:
 at least one of forward traversing and reverse traversing the genome
15 sequence for comparing the 5' terminal tag to at least a portion of the genome sequence to obtain the at least one 5' site.
8. The transcript mapping method as in claim 7, the step of the matching the 3' terminal tag to at least a portion of a genome sequence comprising the step of:
20 at least one of forward traversing and reverse traversing the genome sequence for comparing the 3' terminal tag to at least a portion of the genome sequence to obtain the at least one 3' site.
9. The transcript mapping method as in claim 1, the step of identifying at least
25 one feasible gene location comprising the step of comparing sequence order of each of the at least one occurring segment and one of the at least one 5' site and one of the at least one 3' site corresponding thereto to at least a portion of the genome sequence for obtaining the at least one feasible gene location therefrom.
- 30 10. The transcript mapping method as in claim 9, the step of comparing sequence order of each of the at least one occurring segment and one of the at least one 5' site and one of the at least one 3' site corresponding thereto comprising the

step of comparing the sequence order of each of the at least one occurring segment and one of the at least one 5' site and one of the at least one 3' site corresponding thereto being in accordance with a 5'-occurring segment-3' structure.

5

11. The transcript mapping method as in claim 1, the step of identifying at least one feasible gene location comprising the step of identifying the 5'-3' orientation of each of the at least one occurring segment for obtaining the at least one feasible gene location therefrom.

10

12. The transcript mapping method as in claim 11, the step of identifying the 5'-3' orientation comprising the step of identifying a residual AA nucleotide, the residual AA nucleotide constituting a portion of the 3' terminal tag.

15

13. The transcript mapping method as in claim 1, the step of identifying at least one feasible gene location comprising the step of:

identifying the chromosome wherein each of one of the at least one 5' site and one of the at least one 3' site corresponding to each of the occurring segment is located for identifying the at least one feasible gene location therefrom.

20

14. The transcript mapping method as in claim 1, the step of matching the 5' terminal tag to at least a portion of a genome sequence comprising the step of:

identifying quantity of the at least one 5' site, and

25

the step of matching the 3' terminal tag to at least a portion of a genome sequence comprising the step of:

identifying quantity of the at least one 3' site.

30

15. The transcript mapping method as in claim 14, the step of identifying at least one occurring segment comprising the step of:

traversing along the genome sequence towards one of the extremities thereof from each of the at least one 5' site for identifying at least one of the at least one 3' site.

16. The transcript mapping method as in claim 15, the step of identifying the at least one feasible gene location comprising the step of:
terminating traversal along the genome sequence in response to one of the at least one feasible gene location being identified for each of the at least one 5' site.
17. The transcript mapping method as in claim 14, the step of identifying at least one occurring segment comprising the step of:
traversing along the genome sequence towards one of the extremities thereof from each of the at least one 3' site for identifying at least one of the at least one 5' site.
18. The transcript mapping method as in claim 17, the step of identifying the at least one feasible gene location comprising the step of:
terminating traversal along the genome sequence in response to one of the at least one feasible gene location being identified for each of the at least one 3' site.
19. A computer program storage device tangibly embodying a program of instructions, the program of instructions being readable by a computing device, the program of instructions when executed by the computing device causes the computing device to:
obtain a 5' terminal tag and a 3' terminal tag from a transcript of a gene;
match the 5' terminal tag to at least a portion of a genome sequence to thereby identify at least one 5' site therefrom, each of the at least one 5' site having a sequence matching the 5' terminal tag;
match the 3' terminal tag to at least a portion of the genome sequence to thereby identify at least one 3' site therefrom, each of the at least one 3' site having a sequence matching the 3' terminal tag;
determine quantity disparity between the at least one 5' site and the at least one 3' site;
identify at least one occurring segment, each of the at least one occurring segment being a sequence segment along the genome sequence

extending from one of the at least one 5' site towards one of the at least one 3' site, each of the at least one occurring segment having a sequence length;

identify at least one feasible gene location, each of the feasible gene location being one of the at least one occurring segment having a sequence length not exceeding that of a predefined gene length; and

5 generate a data structure for indexing the genome sequence.

20. The computer storage device as in claim 19, the program of instructions when executed by the computing device, further causes the computing device to:
- 10 provide a nucleotide sequence with at least 16 base pairs for forming the 5' terminal tag; and
- provide a nucleotide sequence with at least 16 base pairs for forming the 3' terminal tag.

21. The computer program storage device as in claim 19, match the 5' terminal tag to at least a portion of a genome sequence comprises:
- 15 match the 5' terminal tag to a chromosome sequence.

22. The computer program storage device as in claim 21, match the 3' terminal tag to at least a portion of the genome sequence comprises:
- 20 match the 3' terminal tag to a chromosome sequence.

23. The computer program storage device as in claim 19, generate a data structure for indexing the genome sequence comprises:
- 25 generate at least one of a tree structure and an ordered array for indexing the genome sequence.

24. The computer storage device as in claim 19, the program of instructions when executed by the computing device, further causes the computing device to:

generate at least one of a suffix array, a suffix tree, a binary tree and a compressed suffix array for indexing the genome sequence.

25. The computer program storage device as in claim 24, match the 5' terminal tag
5 to at least a portion of a genome sequence comprises:

at least one of forward traverse and reverse traverse the genome sequence for comparing the 5' terminal tag to at least a portion of the genome sequence to obtain the at least one 5' site.

5

26. The computer program storage device as in claim 25, match the 3' terminal tag to at least a portion of a genome sequence comprises:

at least one of forward traverse and reverse traverse the genome sequence for comparing the 3' terminal tag to at least a portion of the genome sequence to obtain the at least one 3' site.

10

27. The computer program storage device as in claim 19, identify at least one feasible gene location comprises:

compare sequence order of each of the at least one occurring segment and one of the at least one 5' site and one of the at least one 3' site corresponding thereto to at least a portion of the genome sequence for obtaining the at least one feasible gene location therefrom.

15

28. The computer program storage device as in claim 27, compare sequence order of each of the at least one occurring segment and one of the at least one 5' site and one of the at least one 3' site corresponding thereto comprises:

compare the sequence order of each of the at least one occurring segment and one of the at least one 5' site and one of the at least one 3' site corresponding thereto being in accordance with a 5'-occurring segment-3' structure.

20

25

29. The computer program storage device as in claim 19, identify at least one feasible gene location comprises:

identify the 5'-3' orientation of each of the at least one occurring segment for obtaining the at least one feasible gene location therefrom.

30

30. The computer program storage device as in claim 29, identify the 5'-3' orientation comprises:

identify a residual AA nucleotide, the residual AA nucleotide constituting a portion of the 3' terminal tag.

31. The computer program storage device as in claim 19, identify at least one
5 feasible gene location comprises:
identify the chromosome wherein each of one of the at least one 5' site
and one of the at least one 3' site corresponding to each of the occurring
segment is located for identifying the at least one feasible gene location
therefrom.
10
32. The computer program storage device as in claim 31, identify at least one
occurring segment comprises:
15 traverse along the genome sequence towards one of the extremities
thereof from each of the at least one 5' site for identifying at least one of the at
least one 3' site.
33. The computer program storage device as in claim 32, identify the at least one
feasible gene location comprises:
20 terminate traversal along the genome sequence in response to one of
the at least one feasible gene location being identified for each of the at least
one 5' site.
34. The computer program storage device as in claim 31, identify at least one
occurring segment comprises:
25

traverse along the genome sequence towards one of the extremities thereof from each of the at least one 3' site for identifying at least one of the at least one 5' site.

5

35. The computer program storage device as in claim 34, identify the at least one feasible gene location comprises:

terminate traversal along the genome sequence in response to one of the at least one feasible gene location being identified for each of the at least one 3' site.

10

36. A transcript mapping method comprising the steps of:

obtaining a 5' terminal tag and a 3' terminal tag from a transcript of a gene;

15

matching the 5' terminal tag to at least a portion of a genome sequence to thereby identify at least one 5' site therefrom, each of the at least one 5' site having a sequence matching the 5' terminal tag;

matching the 3' terminal tag to at least a portion of the genome sequence to thereby identify at least one 3' site therefrom, each of the at least one 5' site having a sequence matching the 5' terminal tag;

20

identifying at least one occurring segment, each of the at least one occurring segment being a sequence segment along the genome sequence extending from one of the at least one 5' site towards one of the at least one 3' site, each of the at least one occurring segment having a sequence length; and

25

identifying at least one feasible gene location from the at least one occurring segment, each of the at least one feasible gene location being one of the at least one occurring segment with at least one of the sequence length thereof not exceeding that of the predefined gene length, the sequence order thereof and of the at least one 5' site and one of the at least one 3' site corresponding thereto in accordance with a 5'-occurring segment-3' structure matching the sequence order of the corresponding portion of the genome sequence, the 5' site and one of the at least one 5' site and one of the at least one 3' site corresponding thereto having a 5'-3' orientation, and one of the at

30

least one 5' site and one of the at least one 3' site corresponding to each of the occurring segment being located within the same chromosome; and
generating a data structure for indexing the genome sequence.