

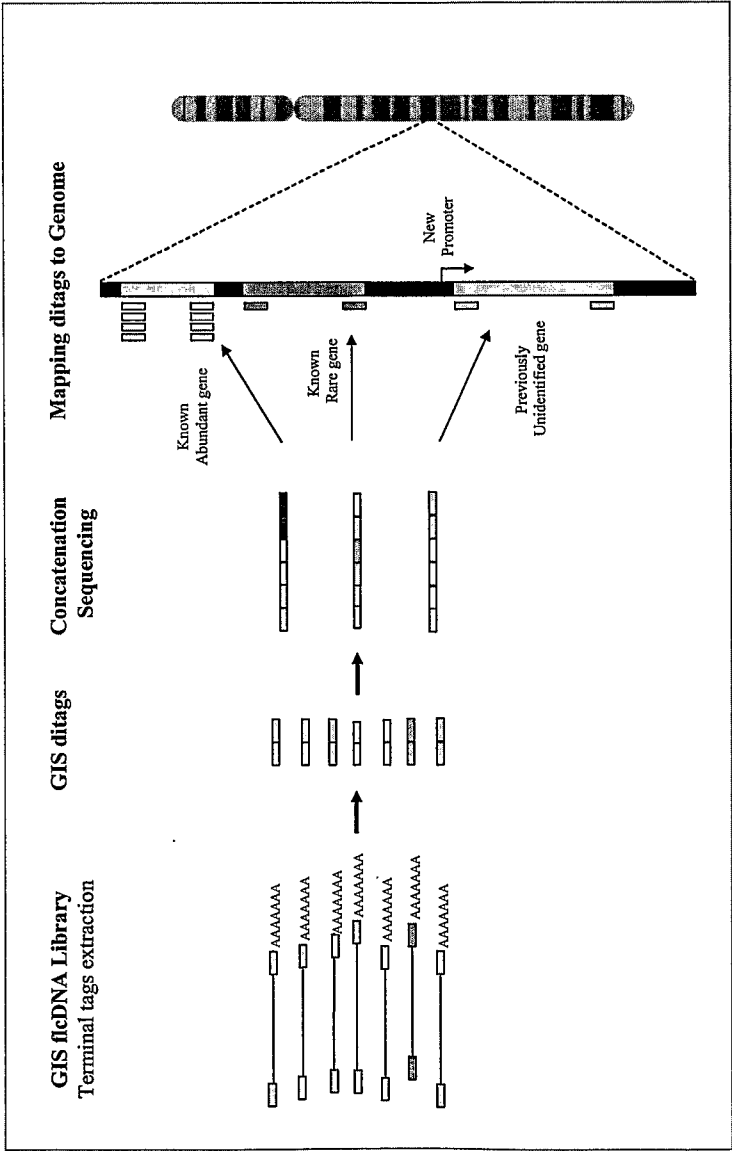


FIG. 1

100

2/6

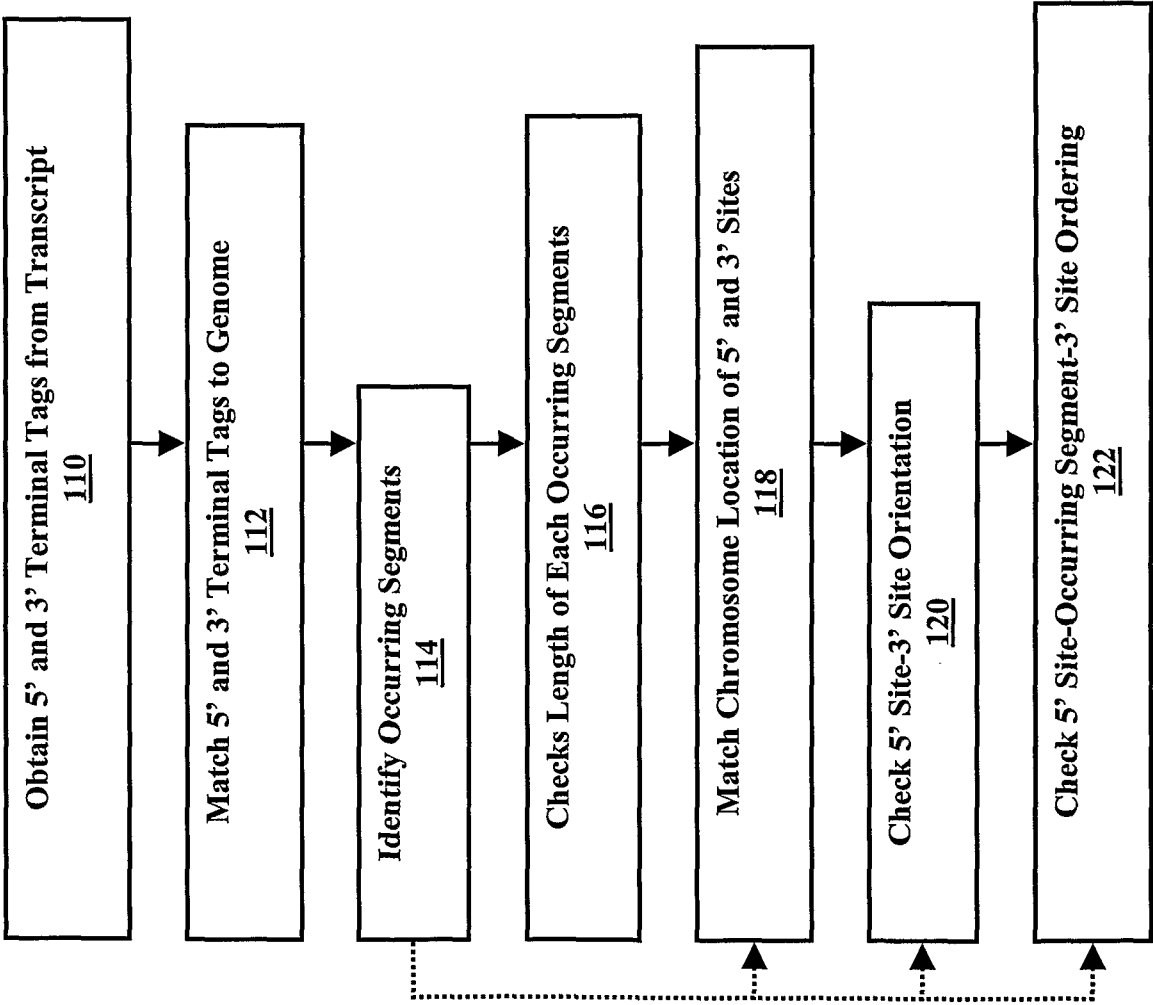


FIG. 2

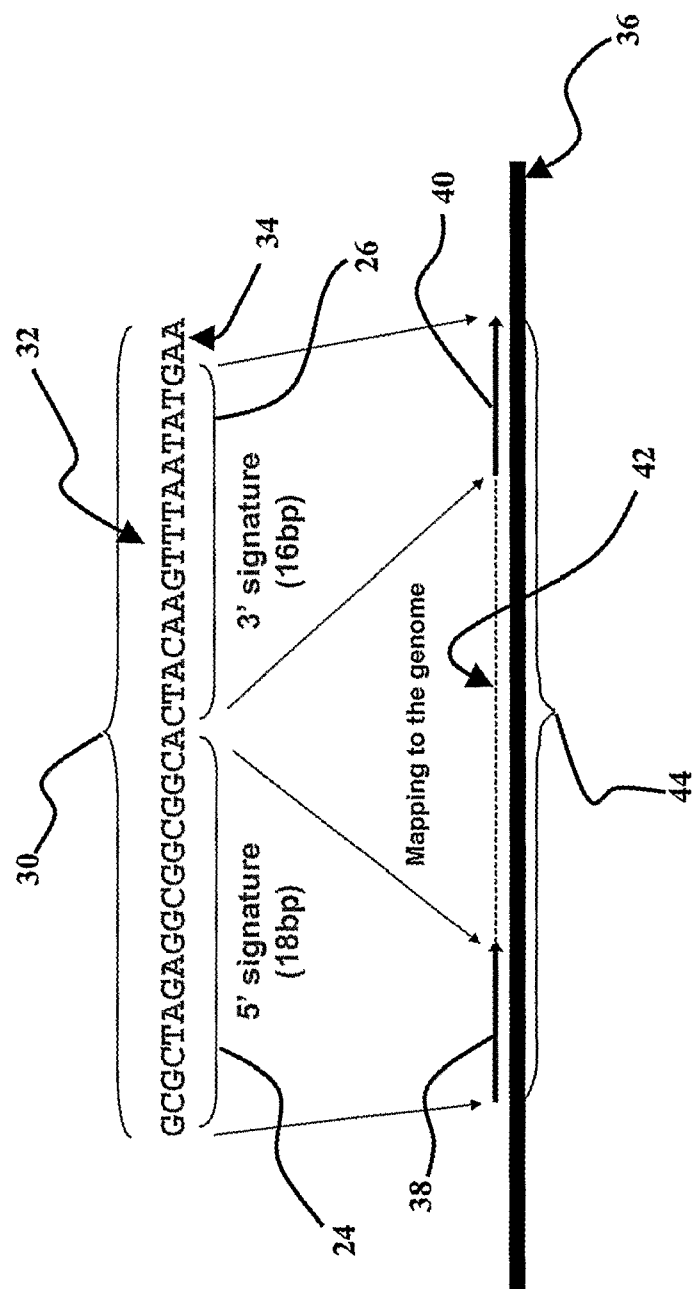


FIG. 3

Algorithm Find_Sites (5' sites)

Input: A DNA segment $P[1..m]$, a genome $G[1..n]$, and a threshold $minlength_5$ of minimum contiguous exact match.

Output: A set of 5' sites.

for $i = minlength_5$ **downto** 1 **do**

 Given the occurrence range of $P[i+1..minlength_5]$, we find the occurrence range of $P[i..minlength_5]$ by Backward search (Lemma 2).

 Extend and report all the occurrences of $P[1..minlength_5]$

for $i = 2$ **to** $boundary_5 - minlength_5 + 1$ **do**

for $j = minlength_5 + 1$ **to** $i + minlength_5 - 1$ **do**

 Given the occurrence range of $P[i..j-1]$, we find the occurrence range of $P[i..j]$ by Forward search (Lemma 1).

 Extend and report all occurrences of $P[i..i+minlength_5-1]$

End Find_Sites (5' sites)

FIG. 4

```

Algorithm Match_Sites_1
Input: A set of 5' sites and 3' sites.
Output: A set of occurring segments (feasible gene locations).

Sort the 5' sites in ascending order and let A[1], A[2], ..., A[p] be the list of 5' sites.
Sort the 3' sites in ascending order and let B[1], B[2], ..., B[q] be the list of 3' sites.
Let i=1 and j=1.
while ( $i \leq p$  and  $j \leq q$ ) do
  if ( $A[i] > B[j]$ ) then
     $j = j + 1$ ;
  elseif ( $B[j] - A[i] \leq \text{maxlength}$ )
    report ( $A[i], B[j]$ ) as an occurring segment and  $j = j + 1$ ;
  else
     $i = i + 1$ ;
End Match_Sites_1

```

FIG. 5

6/6

```
Algorithm Match_Sites_2
Input: A set of 5' sites and 3' sites.
Output: A set of occurring segments.

Let i=1 and j=1.

if (count5 << count3) then
  while (i ≤ p) do
    find 3' site located within maxlength downstream from A[i]
    report found 3' site and A[i] as an occurring segment
    i = i + 1;;
  elseif (count5 >> count3)
    while (j ≤ q) do
      find 5' site located within maxlength upstream from B[j]
      report found 5' site and B[j] as an occurring segment
      j = j + 1;;
    else
      try Match_Sites_1
  End Match_Sites_2
```

FIG. 6