

The Algorithm for the Fan Layout

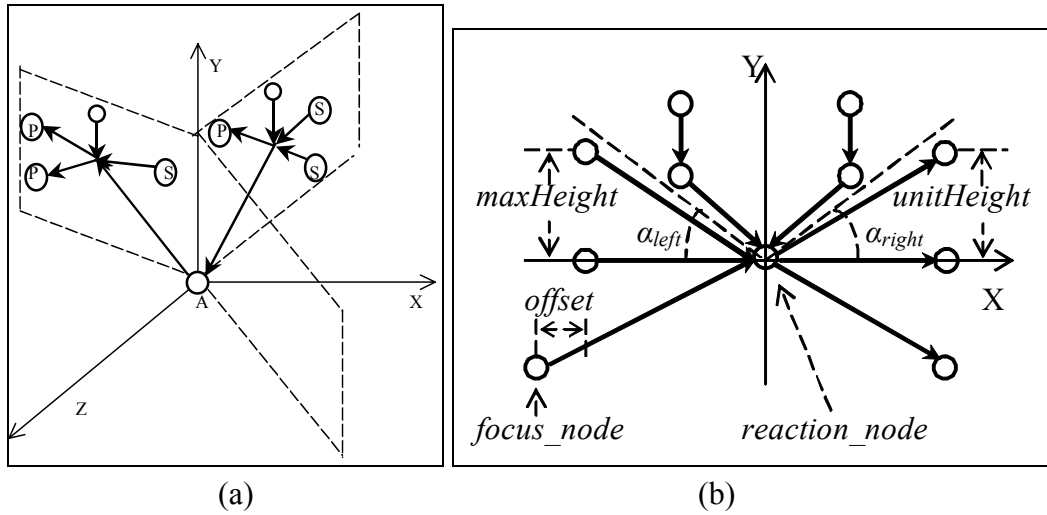
fanLayout(*focus_node*)

```
{
  /* Generate multiple copies for each other node than focus_node if it takes part in
  multiple ones of the reactions that focus_node takes part in. */
  split(focus_node);
  /* Lay out reactions one by one. */
  num = the number of reactions that focus_node takes part in;
  half_num = floor(num÷2); /* the function floor is to round down value */
  i = 0;
  for each reaction r that focus_node takes part in
  {
    /* Lay out reaction r, let the position of reaction_node to be (0, 0, 0). */
    reactionLayout(r);
    /* Translate focus_node along the -X axis with the distance of offset to avoid
    potential overlap of nodes in different reactions (Figure 6(b)). */
    focus_node.position.x = focus_node.position.x - offset;
    /* Translate the layout of reaction r, let the position of focus_node to be (0, 0, 0). */
    for each node n that belongs to r
    {
      n.position = n.position - focus_node.position;
    }
    /* Rotate the layout of reaction r so that all reactions are evenly located in 3D space
    around focus_node (Figure 6(a)) */
    if (i < half_num)
      Rotate the layout of reaction r along the Y axis with the degree of 360÷ half_num
      ×i;
    else
    {
      Rotate the layout of reaction r along the X axis with the degree of 180;
      Rotate the layout of reaction r along the Y axis with the degree of 360÷(num-
      half_num) ×(i- half_num);
    }
  }
}
split(focus_node)
{
  for each pair (i, j) of the reactions that focus_node takes parts in
  {
    /* Find the common nodes that reaction i and reaction j have. i.nodes and j.nodes are
    sets of all nodes in reaction i and reaction j respectively. */
     $\Phi = i.nodes \cap j.nodes$ ;
    /* Generate another copy for each common node except focus_node. */
    for each  $n \in \Phi$ 
    {
      if ( $n == focus\_node$ ) continue;
      make n', a copy of n; /* Split n */
      substitute n' for n in reaction j;
    }
  }
}
```

```

reactionLayout(r)
{
  reaction_node.position = (0, 0, 0);
  lay out the tree rooted at reaction_node;
  calculate ( $\alpha_{left}$ ,  $\alpha_{right}$ ); /* ( $\alpha_{left}$ ,  $\alpha_{right}$ ) is the pie-shape boundary of the tree (Figure 6(b)) */
  /* For each metabolite node, decide whether to draw it at the left side of reaction_node
     or at the right side. The focus_node is always at the left side.  $\Phi_{left}$  is the set of nodes to
     be drawn at the left side.  $\Phi_{right}$  is the set of nodes to be drawn at the right side. */
  if (focus_node is a substrate of reaction r)
  {
     $\Phi_{left}$  = the set of all substrates of reaction r;
     $\Phi_{right}$  = the set of all products of reaction r;
  }
  else /* focus_node is a product of reaction r */
  {
     $\Phi_{left}$  = the set of all products of reaction r;
     $\Phi_{right}$  = the set of all substrates of reaction r;
  }
  /* Lay out the nodes at the left side of reaction_node. Make sure that edges between any
     of these nodes and reaction_node will not intersect the pie-shape boundary ( $\alpha_{left}$ ,  $\alpha_{right}$ ).
     */
   $maxHeight$  = unitHeight × ( $\Phi_{left.size}$  - 1) ÷ 2; /* unitHeight is a constant distance */
   $x$  =  $maxHeight$  ÷ tan( $\alpha_{left} - \delta$ ); /*  $\delta$  is a small constant angle. */
   $i$  = 1;
  for each  $n \in \Phi_{left}$ 
  {
    if ( $n == focus\_node$ )
       $n.position$  = ( $x$ ,  $-maxHeight$ , 0);
    else
    {
       $n.position$  = ( $x$ ,  $-maxHeight + unitHeight \times i$ , 0);
       $i$  ++;
    }
  }
  /* Lay out the nodes at the right side of reaction_node. Make sure that edges between
     any of these nodes and reaction_node will not intersect the pie-shape boundary ( $\alpha_{left}$ ,
      $\alpha_{right}$ ). */
   $maxHeight$  = unitHeight × ( $\Phi_{right.size}$  - 1) ÷ 2;
   $x$  =  $maxHeight$  ÷ tan( $\alpha_{right} - \delta$ );
   $i$  = 0;
  for each  $n \in \Phi_{right}$ 
  {
     $n.position$  = ( $x$ ,  $-maxHeight + unitHeight \times i$ , 0);
     $i$  ++;
  }
}

```



**Figure 6 (a) An illustration of a fan layout of reactions of interest focusing on node ‘A’
 (b) An illustration of how to draw a reaction**

The Algorithm for the Radial Layout

RadialLayout(*gene_node*)

```
{
  for each reaction r that gene_node takes part in
  {
    /* Lay out reaction r using any one of metabolites in r as focus_node, let the position
       of reaction_node to be (0, 0, 0). */
    reactionLayout(r).
    /* Translate the layout of reaction r, let the position of gene_node to be (0, 0, 0). */
    for each node n that belongs to r
      n.position = n.position - gene_node.position;
    calculate ( $\alpha_{r,left}$ ,  $\alpha_{r,right}$ ); /* ( $\alpha_{r,left}$ ,  $\alpha_{r,right}$ ) is the pie-shape boundary of the layout
       (Figure 7(b)) */
     $\alpha_r = \max(\alpha_{r,left}, \alpha_{r,right})$ .
  }

   $k = 2 \times \pi / \sum_r(\alpha_r)$ ;
  for each reaction r that gene_node takes part in
  {
    /* Scale the Y positions of all nodes in r so that angular sum of all reactions is  $2\pi$ 
       (Figure 7(b)). */
     $H' = W_r / \tan(k \times \alpha_r)$ ;
     $s = H'_r / H_r$ ;
    for each node n that belongs to r
      n.position.y = n.position.y  $\times$  s;
     $R_r = \text{abs}(\text{reaction\_node.position.y})$ ;
  }
   $R = \max_r(R_r)$ ; /* R is the radius for the final radial layout; */
   $\alpha = 0$ ;
  i = 0;
  for each reaction r that gene_node takes part in
  {
    /* Scale the X and Y positions of all nodes in r so that all reactions has the same
       radius R */
     $s = R / R_r$ ;
    for each node n that belongs to r
    {
      n.position.x = n.position.x  $\times$  s;
      n.position.y = n.position.y  $\times$  s;
    }
    /* Rotate r around gene_node so that the reaction_nodes for all reactions on the
       circle around gene_node */
    if (i > 0)  $\alpha = \alpha + \alpha_r/2$ ;
    rotate r around gene_node with the degree of  $\alpha$ ;
     $\alpha = \alpha + \alpha_r/2$ ;
    i ++;
  }
}
```

