Inexact Distance Rectangle Graph

April 10, 2011

1 Background

Let $S = e_1, \ldots, e_n$. Without loss of generality, assume that distance $(S(e_1), S(e_m)) = d$. We draw the grid with horizontal and vertical axes as: $e_1, \ldots, e_n$ and $e_m, \ldots, e_n, \ldots, e_{m-1}$. Each (k, d)-mer is represented as a single point on the grid and the set of all (k, d)-mers forms a diagonal on the grid. Rectangles that are intersected by the diagonal form a **distance constraint set** P (Fig. 1). Each rectangle can be identified by a triple: $(e_i, e_j, \delta_{i,j})$, where e_i is the red edge, e_j is the blue edge and $\delta_{i,j}$ is the genomic distance between the starts of e_i and e_j.

Given the distance constraint set P, we define a **two dimensional quotient graph RBG** as follow: For each edge-triple (e_i, e_j, δ) in P, construct a rectangle with vertices labeled as defined in the previous draft. The two dimensional quotient graph is obtained by gluing all vertices with the same labels into a single vertice. Note that in this graph, there are two types of edges: red and blue edges. Red edges correspond to the first and blue edges correspond to the second edges in the edge-triples. We denote G_b, G_r as subgraphs obtained from RBG by deleting all red and blue edges respectively.

We define the red graph RG as a graph obtained from RBG by contracting each weakly connected component of G_b into a single vertice. The blue graph BG can be defined on the RBG in a similar way (by contracting each weakly connected component of G_r into a single vertice). We call BG, RG one-dimensional quotient graphs.

In general, all graphs defined above can also be defined from **any arbitrary set of rectangles P** in the grid with the assumpsion that the genomic distance between any pair of edges in this set can be estimated exactly (Fig. 3). We call **P red-consistent** if the genome S corresponds to an Eulerian cycle in RG constructed from **P**. We call **P blue-consistent** if the genome S corresponds to an Eulerian cycle in BG constructed from **P**.

Theorem 1 *The set of rectangles P is* red-consistent *iff for any red edges e_i in S there exists at least one rectangle in P that has e_i as red edge and any pair of (end points of) red edges e_i in any rectangles in P are connected by using only the blue edges in P* on the grid. *The same conditions apply for case of blue-consistent.*

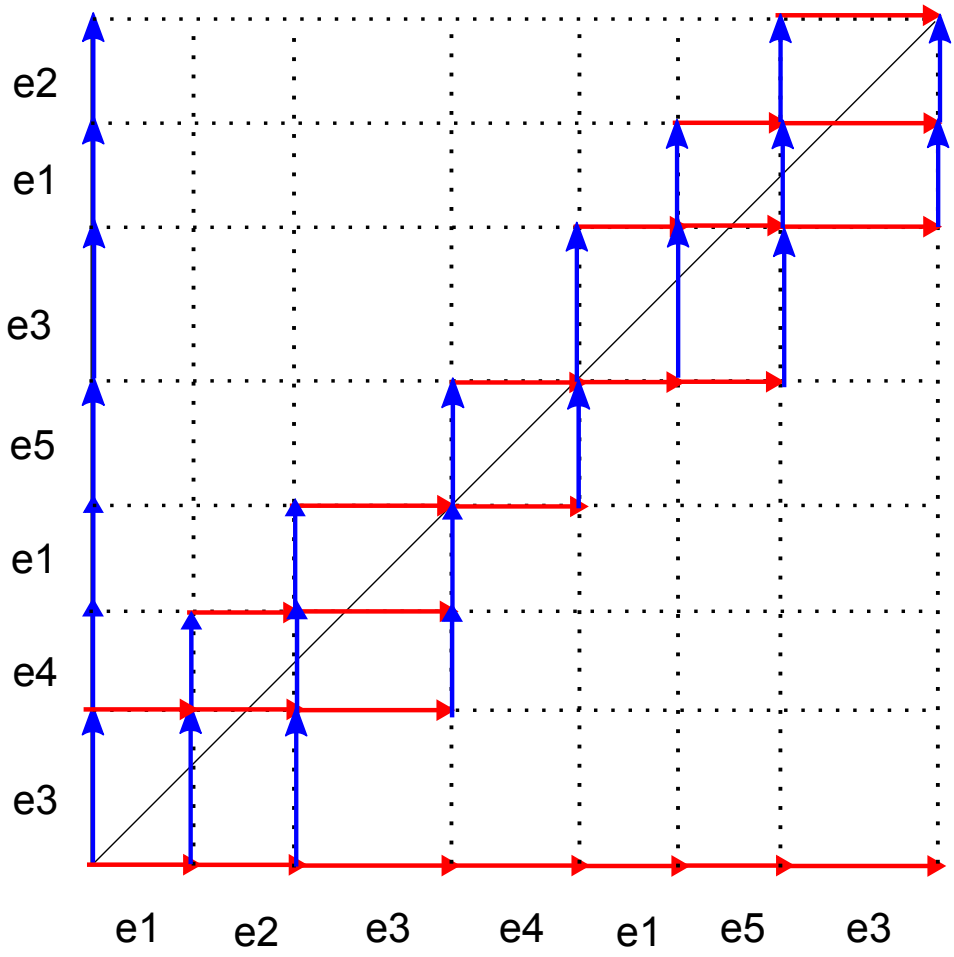

Figure 1: Rectangles intersected by the diagonal form a **distance constraint set**

The set of rectangles in figure 1 is both red and blue-consistent, while the set of rectangles in figure 3 is only red-consistent.

Corollary 1 *The distance constraint set (rectangles intersected by the diagonal) is both red and blue consistent.*

In reality, S is unknown but we can use the mate-pairs to create a set of rectangles P (A rectangle $(e_i, e_j, \delta_{i,j}) \in P$ if there is a mate-pair (a,b) that $a \in e_i$ and $b \in e_j$). If the insert size is exact, P is the distance constraint set and therefore is both red and blue consistent. When distance between mate-pairs are not exact, P may differ from the distance constraint set and therefore not guaranteed to be red or blue consistent. The following section describes the algorithm for inexact insert distance.

2 Algorithm

Let assume that for every edge e_i in S, there exists at least one rectangle in P that contains e_i as red edge. Figure 4 illustrates this case, where for every edge e_i in S, there exists at least one rectangle that contains the red edge e_i[1]. However, these rectangles do not form a red-consistent constraint set since not all red edges e_2 are connected by blue edges. To make it red-consistent, we should modifly the gluing rule (Fig. 5).

Gluing rule: Two vertices $A = (red\_edge\_end_1, blue\_edge\_end_1, distance_1)$,$B = (red\_edge\_end_2, blue\_edge\_end_2, distance_2)$ are glued if

- $red\_edge\_end_1 = red\_edge\_end_2$
- distance between $blue\_edge\_end_1$ and $blue\_edge\_end_2$ in the de Bruijn graph is less then Δ
- $|distance_1 - distance_2| < \Delta$

In this approach, only the RG can be used to spell the contigs. If we want to use the BG to spell the contigs, the gluing rule should be as follow:

Gluing rule: Two vertices $A = (red\_edge\_end_1, blue\_edge\_end_1, distance_1)$,$B = (red\_edge\_end_2, blue\_edge\_end_2, distance_2)$ are glued if

- $blue\_edge\_end_1 = blue\_edge\_end_2$
- distance between $red\_edge\_end_1$ and $red\_edge\_end_2$ in the de Bruijn graph is less then Δ
- $|distance_1 - distance_2| < \Delta$

[1]This does not hold for blue edges, for instance blue edge e_4 is not contained in any rectangle in P

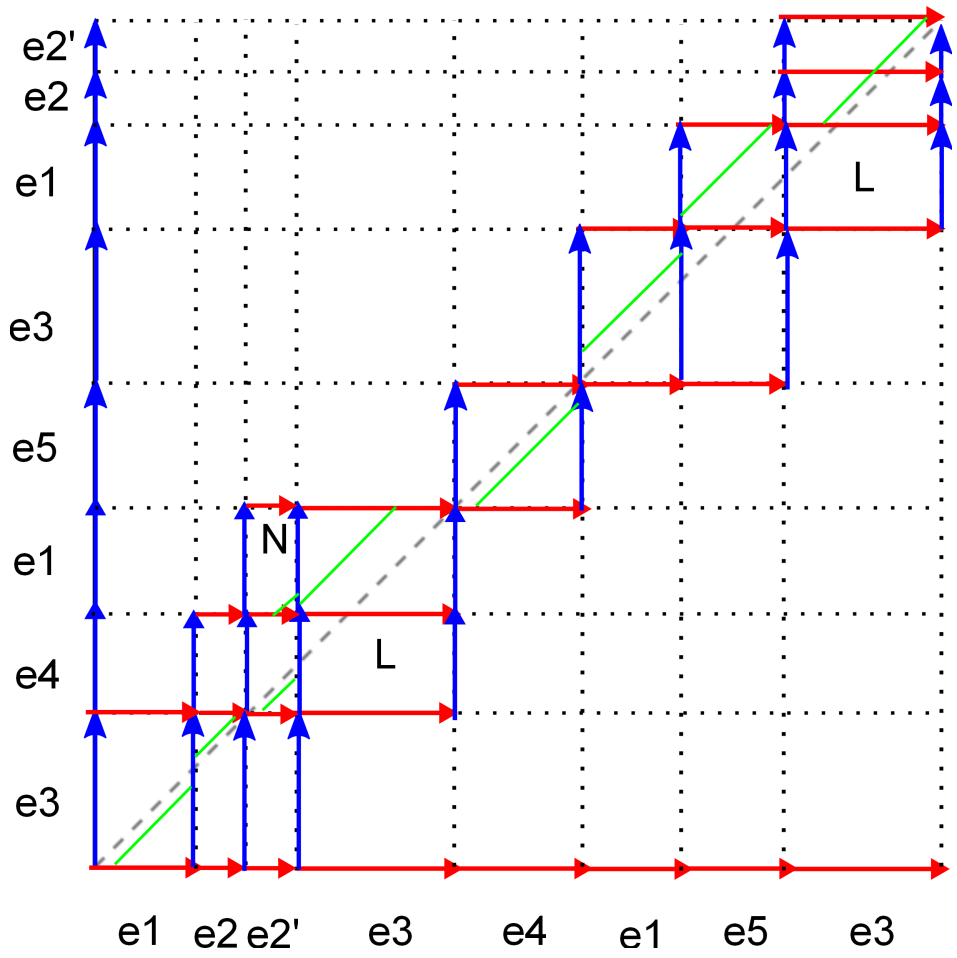

Figure 2: Distances between pair of edges can not be estimated correctly. The dot diagonal lines shows the exact distance case while the perturbed green diagonal lines are the results of the inexact distance estimation. Letter N written inside the rectangle shows that this is a new (wrong) rectangle added, corresponds to an in-correct edge pair added into the set P. The rectangle with letter L denoted that it is lost in the set of considered rectangles.

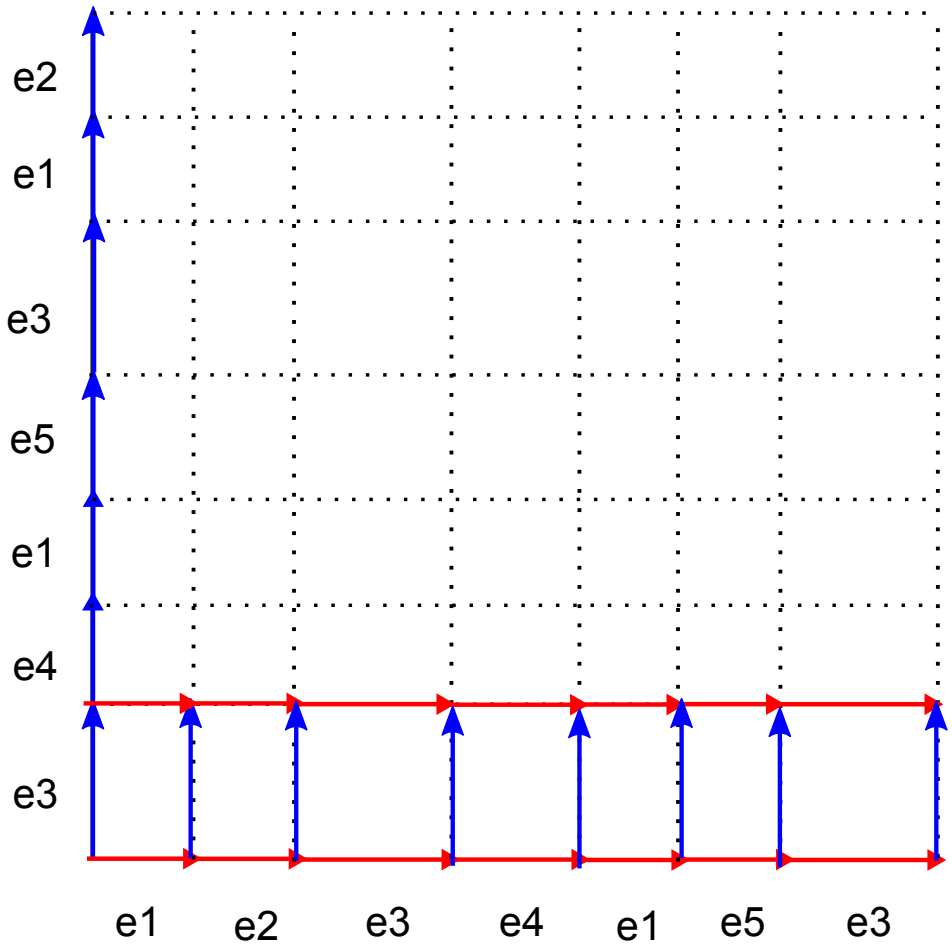

Figure 3: A red-consistent set of rectangles

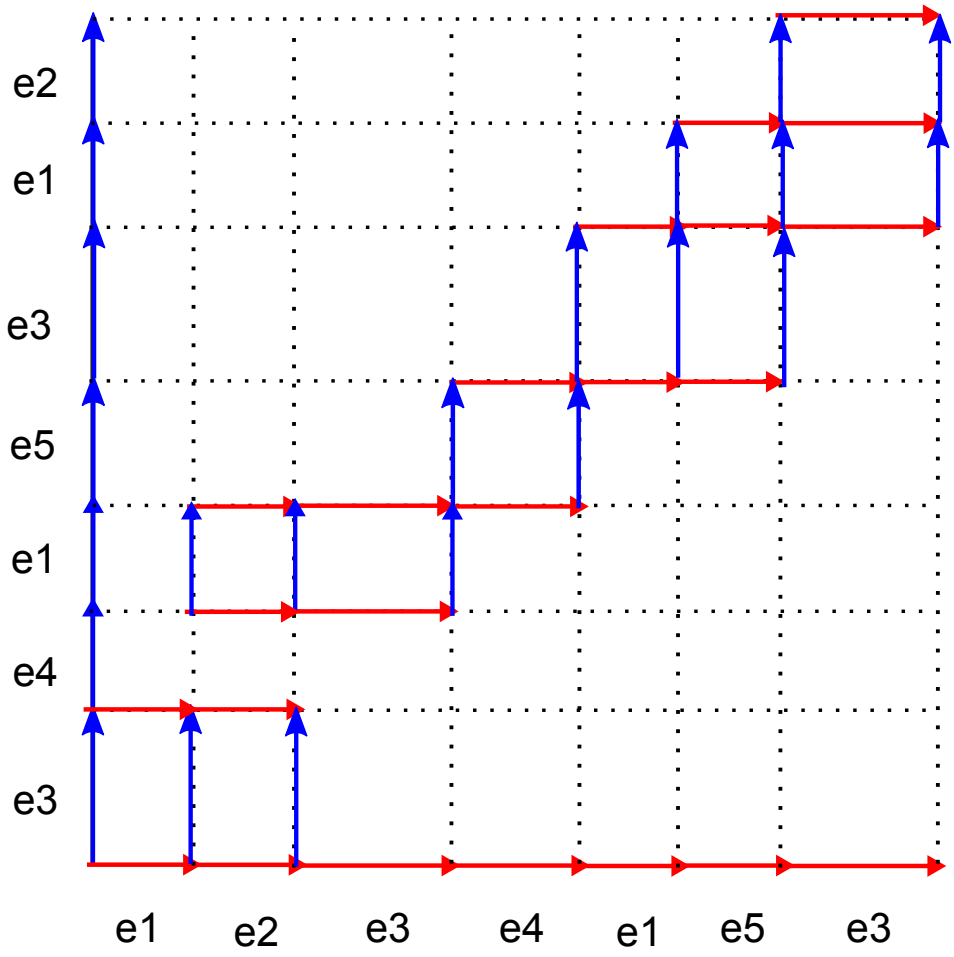

Figure 4: Neither red nor blue-consistent set

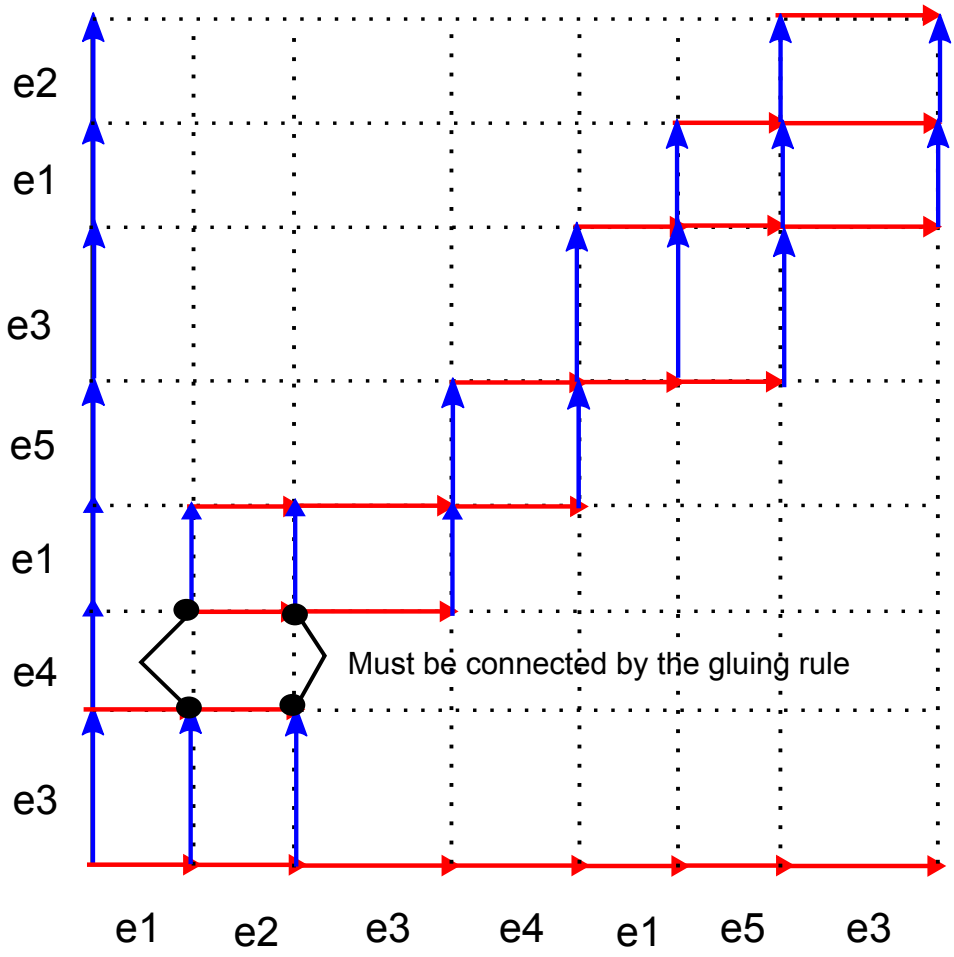

Figure 5: red-consistent by using the gluing rule

3 Discussion

- While this approach is assymetric, I think that it is easy to implement. The symmetric approaches that I described in the previous document turn out to be either (1) Introducing erroneous contigs or (2) Making it difficult to spell the contigs.
- The assumption that for every edge e_i in S, there exists at least one rectangle in P that contains e_i as red edge is reasonable for long edges. For short edges, this assumption may not hold but can be resolved by checking neighboring edges in the de Bruijn graph and adding additional rectangles into P.