Overlap Calculus:

The rules for overlap graph reduction.

Granger Sutton and Clark Mobarry

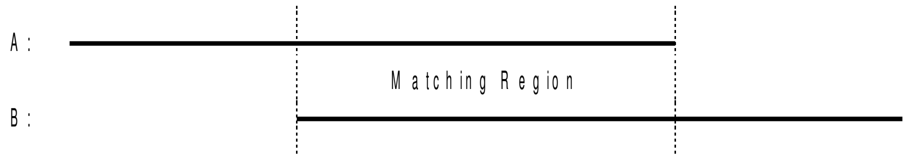
1. Overview

1.1. Overlaps

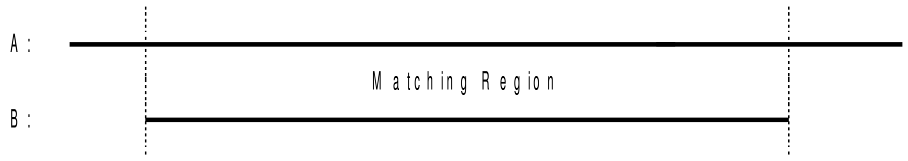
Two sequences, *A* and *B*, are said to *overlap* iff there is a sufficiently long subsequence of *A* that matches a subsequence of *B* to within a specified degree of similarity. This degree of similarity generally reflects the belief that both subsequences were obtained from the same position in the genome being sequenced.

Two major types of overlaps are possible between fragments *A* and *B*:

1. *Dovetail Overlap*: A complete suffix/prefix of *A* matches a complete suffix/prefix of *B*.



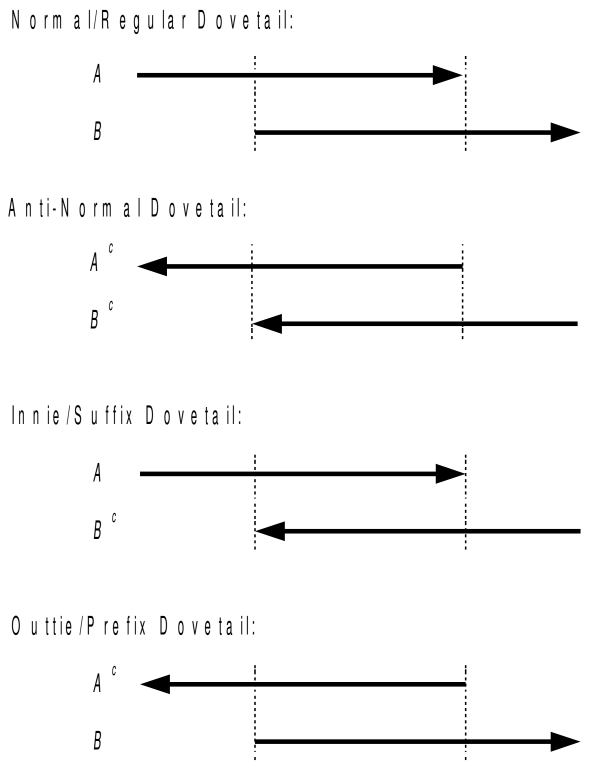
2. *Containment Overlap*: The entirety of *B* matches a subsequence of *A*, or *vice versa*.



The relative orientation of fragments A and B further divide the two major types of overlaps into subtypes. The following subtypes exist for dovetail overlaps:

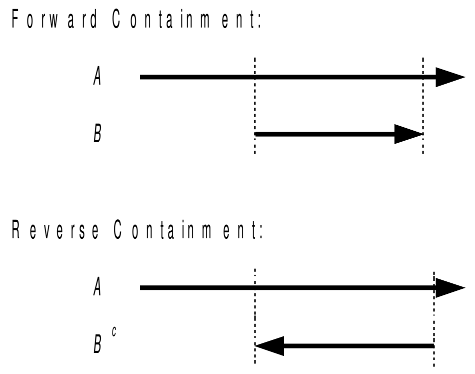
- `Normal Dovetail` — A suffix of *A* matches a prefix of *B*.
- `Antinormal Dovetail` --- A prefix of A^c matches a suffix of *B*.
- `Innie Dovetail` — A suffix of *A* matches a suffix of B^c.

- `Outtie Dovetail` — A prefix of A^c matches a prefix of *B*.



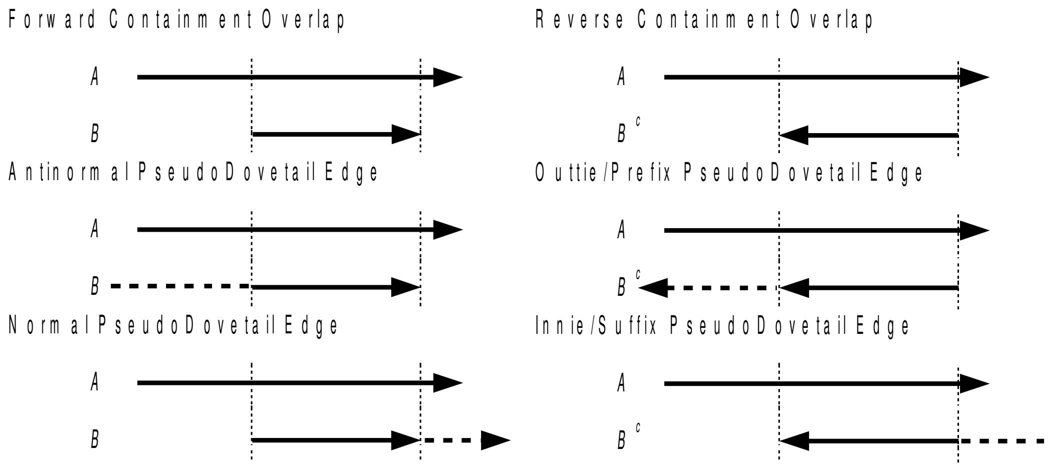
The following subtypes exist for containment overlaps:

- `Forward Containment` — *A* contains *B*.
- `Reverse Containment` — *A* contains B^c.

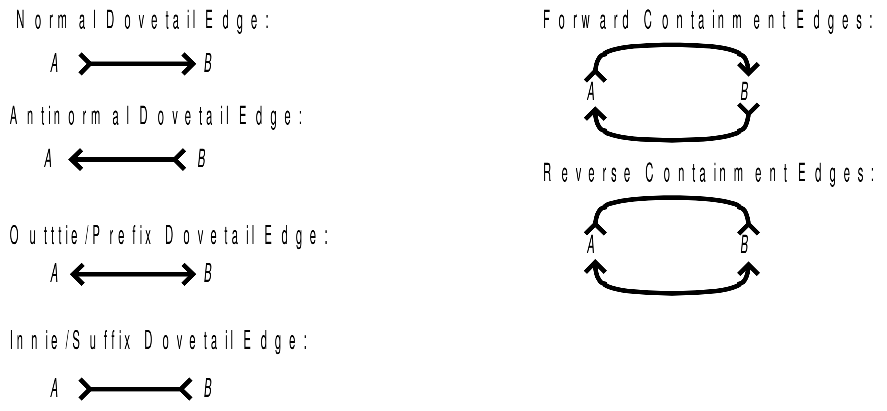


1.2. Overlap Graph

The overlap graph is a representation of a set of fragments and all pairwise overlaps between these fragments. The set of fragments is represented one-to-one by the set of vertices. The set of edges represents both dovetail and containment overlaps. The dovetail overlaps have a one-to-one correspondence to dovetail edges, but each containment overlap is represented by two containment edges. Two containment edges are used to represent each containment overlap in order to simplify the edge reduction rules. The idea is to convert the containment overlap into two pseudodovetail edges where for the most part these edges can be treated as real dovetail edges. We create these containment edges by determining what kind of dovetail edges would result from lengthening the prefix (first edge) and then the suffix (second edge) of the contained fragment.


Forward Containment Overlap
A
B
Reverse Containment Overlap
A
B c
Antinormal Pseudo Dovetail Edge
A
B
Outtie/Prefix Pseudo Dovetail Edge
A
B c
Normal Pseudo Dovetail Edge
A
B
Innie/Suffix Pseudo Dovetail Edge
A
B c

Each edge has two arrowheads - one at each vertex the edge connects. The arrowhead indicates whether the prefix or suffix of a fragment is involved in the overlap. Arrowheads pointing toward a vertex indicate the prefix is involved. Arrowheads pointing away from a vertex indicate the suffix is involved.


Normal Dovetail Edge:
A
B
Antinormal Dovetail Edge:
A
B
Outttie/Prefix Dovetail Edge:
A
B
Innie/Suffix Dovetail Edge:
A
B
Forward Containment Edges:
A
B
Reverse Containment Edges:
A
B

The arrowheads do not represent enough information about an edge but are helpful for visualization. One reason the arrowheads are not enough information is that there is not necessarily only one type of edge between a given pair of fragments or even only one edge of a given type/subtype (the graph may be a multigraph). We need to define edge attributes and some notation. We will indicate some edge π between fragments *f* and *g* as $f\pi g$. We will indicate the arrowhead information as $\pi.suf_f$ and as $\pi.suf_g$ where each is a boolean value and $\pi.suf_f$ is true iff the suffix of *f* is in the overlap. We also need to indicate the amount of each fragment which is not within the overlap. For dovetail edges, only one of the prefix or suffix of each fragment is not in the overlap and needs to be measured. The notation $\pi.\mathit{hang}_f$ is the amount of fragment *f* (the overhang) not in the overlap. This is unambiguous for dovetail edges and whether the overhang involves the prefix or suffix is indicated by $\pi.\mathit{suf}_f$. The notation $\pi.\mathit{hang}_g$ is defined analogously. These four attributes ($\pi.\mathit{suf}_f$, $\pi.\mathit{suf}_g$, $\pi.\mathit{hang}_f$, $\pi.\mathit{hang}_g$) define a dovetail overlap/edge. The same four attributes can be defined for each pseudodovetail edge of a containment overlap. The one significant difference is that the contained edge does not have an overhang but rather an underhang. If the contained fragment is *f* then $\pi.\mathit{hang}_f$ by convention will be nonpositive and have a magnitude equal to the length the prefix (if $\pi.\mathit{suf}_f$ is true) or suffix (if $\pi.\mathit{suf}_f$ is false) of *f* needs to be extended to reach the appropriate end of *g*. Even though it is convenient to conceptualize a containment overlap as two edges in the overlap graph the actual implementation only needs to maintain a pair of 4-tuples for the edge reduction rules. In addition, either 4-tuple is directly computable from the other: $\pi^1.\mathit{suf}_f = \neg\pi^2.\mathit{suf}_f$, $\pi^1.\mathit{suf}_g = \neg\pi^2.\mathit{suf}_g$,$\pi^1.\mathit{hang}_f = -\pi^2.\mathit{hang}_g$, and $\pi^1.\mathit{hang}_g = -\pi^2.\mathit{hang}_f$.

1.3. Graph Reduction Rules

An overlap graph has on average approximately *O(n-1)* edges per vertex for a sequence devoid of repeats and sequenced to coverage *n*. The goal is to remove edges from the graph so that there is approximately one edge per vertex for this same sequence (because containment overlaps are represented by two edges there will actually be some extra overhead). The method for removing edges is to find edges of the graph which are inferable by other edges remaining in the graph. The *Transitive Edge Removal* principle: if $f\tau g\sigma h$ and $f\pi h$ are mutually consistent overlaps among fragments *f*, *g*, and *h*, then the edge representing π is removed. Informally overlaps are mutually consistent if the overlap between *f* and *h* implied by the overlaps τ and σ is the same as that of π within error rate ε, i.e. $\pi \approx_\varepsilon \tau \circ \sigma$.

The are two flavors of graph reduction used by the Celera Assembler. The first flavor is to define the edge marking rules for triplets of fragments. Edges are marked for removal but are still available to remove still more edges. Only after every edge is considered and marked for removal or retention are any edges permanently removed from the graph. The second flavor is to incrementally remove edges under consideration as they are determined inferable. Edge inference is available in the to removal any remaining edges.

1.3.1 Transitive graph reduction for set intersection using triplets

Consider graph reduction for a familiar operator such as set intersection. The set intersection operator is binary, associative, symmetric, and reflexive, which helps to simplify the discussion. Consider an "all-against-all" graph where the vertices represent sets and the edges represent all non-empty pair-wise set of the sets. Edges of the all-against-all graph that are inferable by traversing an appropriate path of edges of the graph are removed to form the transitively reduced graph. Later, the transitively reduced graph can be completed by adding back in all inferable edges to obtain the all-against-all graph. The reduction and completion operators must be consistent so that (1) established intersection relationships are not lost and (2) incorrect intersection relationships are not inferred.

To be concrete, the inference rules are defined for triplets of sets. Take three sets: f, g, and h, with three pair-wise non-empty set intersections: $\tau = f \cap g \neq \emptyset$, $\sigma = g \cap h \neq \emptyset$, and $\pi = f \cap h \neq \emptyset$. We define the set intersection π as weakly inferable by τ and σ, when $\pi = \tau \cap \sigma$. We define the set intersection π as strongly inferable by τ and σ, when $\pi = \tau \cap \sigma$, $\tau \neq \pi \cap \sigma$, and $\sigma \neq \tau \cap \pi$. The relevant difference between strong and weak inference is that strong inference does not allow cycles of inference. If there are no cycles of inference, then edge inference can be used to reduce the graph. As discussed later, weak inference and an appropriate tie-breaker can also prevent cycles of inference. The ties exist in this weak inference rule when $\pi = \sigma$ or $\pi = \tau$. When there is a tie, then it is possible to use τ and σ to remove π, and then use π and $\sigma(\tau)$ to remove $\tau(\sigma)$. If both edges are removed, then the information in the all-against-all graph is not conserved in the reduced graph. Note that if when $\pi = \tau \cap \sigma$, and $\tau = \pi \cap \sigma$, then $\tau = \pi$. Thus, an alternate way to state strong inference is that $\pi = \tau \cap \sigma$, $\tau \neq \pi$, and $\sigma \neq \pi$. We define the set intersection $\pi = f \cap h$ as interlocking inferable, when $f \cap h \subset g$ and $f \cup h \supset g$. Interlocking inferable generalizes to a sequence of sets. A sequence of sets is interlocking when the common intersection/union of any sub-sequence of sets is the intersection/union of the end-sets of the sub-sequence of sets.

The removed edges are inferred in triplet manner. Given three sets: f, g, and h, and two non-empty, pair-wise set intersections: $\tau = f \cap g \neq \emptyset$, and $\sigma = g \cap h \neq \emptyset$, then we know that $\pi = f \cap h$ is possibly non-empty. We know that $\pi \supseteq \tau \cap \sigma$. The edge inference rule is that if an edge representing $\pi = f \cap h$ is not present in the reduced graph and $\tau \cap \sigma \neq \emptyset$, then π is inferred to be $\tau \cap \sigma$.

The following figure is all possible set intersection topologies of three sets, where we differentiate between containment and non-containment intersections. There is a Venn diagram and a graph for each possibility. Containment intersections are represented by a double directed edge and non-containment intersections are represented by an undirected edge.

I. Zero contained sets

A. All disjoint

B. One non-containment

C. Two non-containment

D. Three non-containment: one strongly inferable edge because one set contains the intersection of the other two sets. There is an interlocking inferable edge only if the middle set is a subset of the two outer sets.

E. Three non-containment: no inferable intersections

II. One contained set

A. Divorced contained

B. Dating contained

C. Step contained

The non-containment edge to the contained set is strongly inferable. However, this edge is not interlocking inferable.

D. Joint contained

Two containment edges are only weakly inferable. Neither the flattening nor stacking tie-breaker rules resolve this degeneracy. In the fragment overlap case, the pseudo-overlap pair technique can be used. Also containment edges from a set of three or more interlocking sets to a common contained set can possibly be trimmed.

III. Two contained sets

A. Cousin contained

B. Sibling contained

One strongly inferable intersection because the container set contains the intersection of the other two sets.

C. Grand contained

There are two weakly inferable intersections to the multiply contained set. The flattening or stacking tie-breaker can be used to eliminate the degeneracy. The sequence of set from biggest to smallest is interlocking, so the intersection between the biggest and smallest set is interlocking inferable.

From the previous figure, note that the triplet topologies 1D, 2C, and 3B each have a strongly inferable non-containment edge that can be removed from the graph. The triplet topologies 2D and 3C each have weakly two inferable containment edges to the same multiply contained set, that would need a tie breaker to allow conservative edge removal of one of the edges.

The grand-contained and the joint-contained topologies are the only relationships involving three sets that have a cycle of weak inference. In the grand-contained topology, suppose that g is contained by f, and h is contained by f and g, then $\tau \cap \sigma = (f \cap g) \cap (g \cap h) = h = (f \cap h) = \pi \cap g$ and $\tau \cap \pi = (f \cap g) \cap (f \cap h) = h = g \cap h = \sigma$. This would imply that τ and σ weakly infers π, and τ and π weakly infers σ. This cycle of inferences can cause problems in transitively inferable edge graph reduction. A parallel implementation of edge removal would find both of the edges corresponding to σ and π inferable and removable, which would orphan h.

The flattening tie-breaking rule is similar to that used by the Drosophila assembler. The flattening tie-breaking rule for weak inference requires that the "middle" set, g, must not be contained by either of the two outer sets, f or h. Manipulate the weak inference rule: $\tau \cap \sigma = (f \cap g) \cap (g \cap h) = (f \cap h) \cap (g) = \pi \cap g$. Thus $\pi \subseteq g$ is a requirement for π to be weakly inferable from τ and σ. The middle rule is more restrictive than weak inference since it implies that $\pi \neq g$, $g \not\subset f$, and $g \not\subset h$, but less restrictive than strong inference.

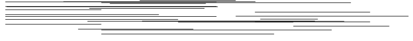
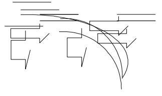
The stacking tie-breaking rule for weak inference requires that when a set, h, is contained by the other two fragments in the triplet, f and g, then the edge between f and h is removable when f, g, and h are an interlocking sequence of sets.

1.3.2 Transitive graph reduction for fragment overlaps

The graph induced by fragment overlaps is a sub-graph of the graph induced by fragment overlaps. In the absence of zero sequence errors and mutations, the fragment overlaps are equivalent to set intersections that are restricted to be contiguous sub-fragments. Dovetail and containment overlaps are further restrictions of possible overlaps.

A fragment overlap is represented by (1) the overlap hangs and orientations of the two fragments necessary to specify the relative layout of the fragments, and (2) the sub-sequence of each fragment actually in the overlap. Specifying the sub-sequence of each fragment in the overlap is unnecessary for dovetail and containment overlaps, since it is inferred by the relative layout and the knowledge that only dovetail and containment overlaps are to be considered. For the overlaps that are inferred from a path of dovetail and containment overlaps, it is necessary to include information specifying the sub-sequence of two outer fragments of the inferred overlap.

The possible topologies of three fragments overlapping without worrying about

fragment orientation are shown below with a fragment layout and graph representation.

2. Zero contained fragments

A. All disjoint

B. One disjoint
one overlap

C. Two overlaps

D. Three overlaps
one removable

E. Three overlaps
none removable

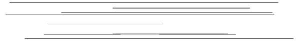
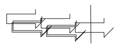
2. One contained fragment

A. Divorced contained

B. Dating contained

C. Step contained

D. Joint contained

3. Two contained fragments

A. Cousin contained

B. Sibling contained

C. Grand contained

The fragment triplet topologies 1D, 2C, and 3B have a strongly inferable non-containment edge. The fragment triplet topologies 2D and 3C have weakly inferable containment edges that need a tie-breaker to allow an edge to be removed.

.

2.1.1 Graph reduction for our implementation of fragment triplets

Topological sieve:

$$\pi.\mathrm{suf}_f = \tau.\mathrm{suf}_f,$$

$$\pi.\mathit{suf}_h = \sigma.\mathit{suf}_h,$$

and

$$\tau.\mathit{suf}_g \neq \sigma.\mathit{suf}_g;$$

and a geometrical sieve:

$$\mathrm{abs}(|\pi| + |g| - |\tau| - |\sigma|) <= \alpha + \varepsilon\,|\sigma|.$$

A simple, position-based formulation for practical purposes is use the subsitutions:

$$\tau.\mathit{hang}_f + \sigma.\mathit{hang}_g \in \pi.\mathit{hang}_f \pm (\,\varepsilon \cdot \mathit{length}(\tau) + \alpha\,),$$

and

$$\sigma.\mathit{hang}_h + \tau.\mathit{hang}_g \in \pi.\mathit{hang}_h \pm (\,\varepsilon \cdot \mathit{length}(\sigma) + \alpha\,).$$

If in equation 5, $\tau.\mathit{hang}_f = |f| - |\tau|$, $\sigma.\mathit{hang}_g = |g| - |\sigma|$, and $\pi.\mathit{hang}_f = |f| - |\pi|$, then we have

$$\mathrm{abs}(|\pi| + |g| - |\tau| - |\sigma|) <= \alpha + \varepsilon\,|\tau|.$$

If in equation 5, $\tau.\mathit{hang}_g = |g| - |\tau|$, $\sigma.\mathit{hang}_h = |h| - |\sigma|$, and $\pi.\mathit{hang}_h = |h| - |\pi|$, then we have

Notice that the topological sieve and the geometrical sieve have some ambiguity in their implementation.

This rule is sufficient for dovetail edges but needs to be augmented for containment (pseudodovetail) edges to require:

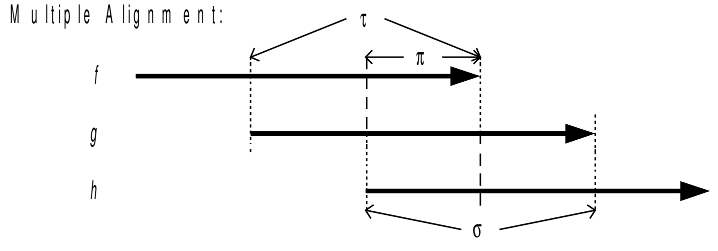
$\tau.hang_g > 0$ or ($\tau.hang_g = 0$ and $\tau.hang_f < 0$) or ($\tau.hang_g = 0$ and $\tau.hang_f = 0$ and a tie breaker),

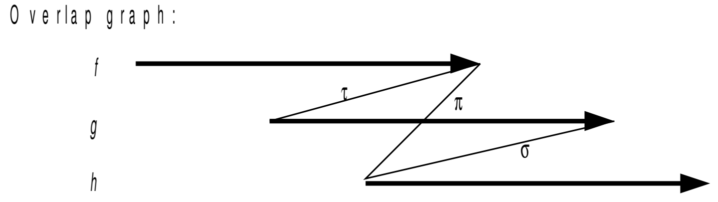
and

$\sigma.hang_g > 0$ or ($\sigma.hang_g = 0$ and $\sigma.hang_h < 0$) or ($\sigma.hang_g = 0$ and $\sigma.hang_h = 0$ and a tie breaker).

In plain english, a topological constraint is that the middle fragment (*g)* must not contained in either of the outer fragments (*f* or *h*, subject to tie breaking) when removing edge π.

The transitive edge removal rule is just one way to implement the idea of removing edges that can be inferred from other edges. One obvious extension would be to consider more than three edges at a time but this would complicate the rule. Two figures will help illustrate what the rule is encoding.



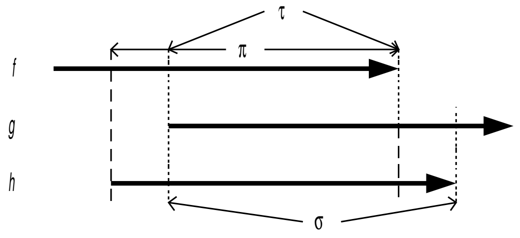


The multiple alignment view of three fragments *f*, *g*, and *h* shows how the overlap π can be inferred from the overlaps τ and σ. (The reader should confirm that the transitive edge removal rule is satisfied based on this configuration). One important observation is that for any three fragments and corresponding three edges there are six possible labelings with *f*, *g*, and *h* (the π, τ, and σ labels are implied). However, three of these labelings are isomorphic with the other three in terms of satisfying the transitive edge removal rule. This is because one fragment is chosen as the middle fragment *(g)* and the other two as the outer fragments. Once the choice of the middle fragment *(g)* is made the *f* and *h* labeling is arbitrary because the transitive edge removal rule is symmetric in the outer fragments *f* and *h,* and their overlaps τ and σ.

An Illustration of what violating each of the clauses of the transitive edge removal rule implies is shown below.

For,

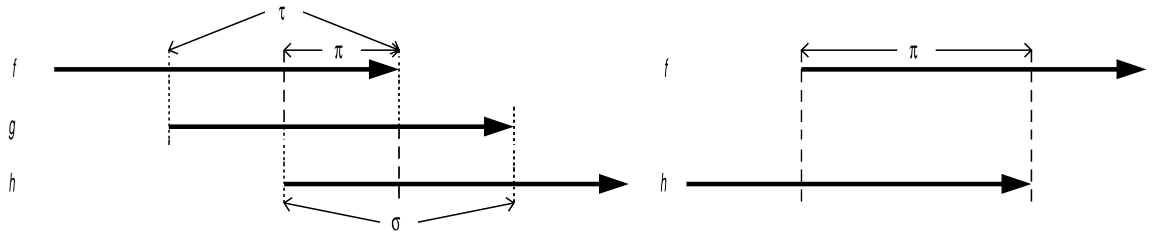
$$\tau.suf_g = \sigma.suf_g$$



The above clause of the rule tells you that if you have a multiple alignment of three fragments there is only one correct one to choose the middle fragment *(g).*

For,

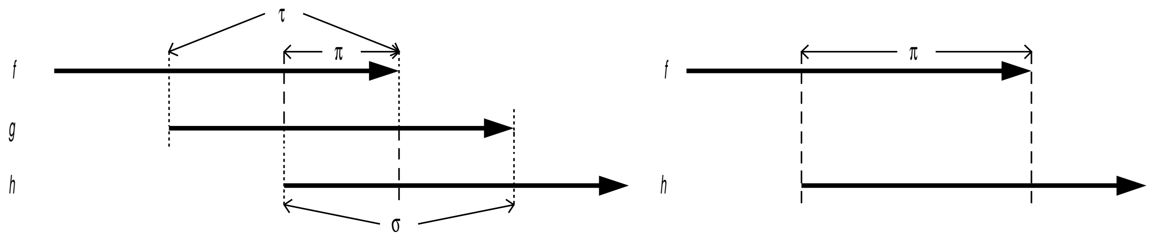
$$\pi.suf_f = \tau.suf_f$$
$$\pi.suf_h = \sigma.suf_h$$



These two clauses of the rule protect against the case where there are two or more edges between some of the fragments creating a multi-graph (in the example above two overlaps between fragments *f* and *h*). The clauses are only satisfied for edges of the correct topology as encoded in the *suf* attributes.

For,

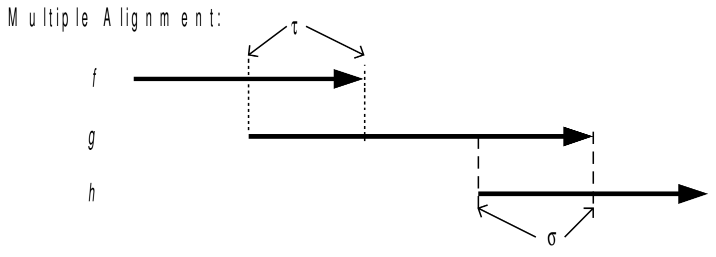
$$\tau.hang_f + \sigma.hang_g \in \pi.hang_f \pm (\varepsilon \cdot length(\tau) + \alpha)$$
$$\tau.hang_g + \sigma.hang_h \in \pi.hang_h \pm (\varepsilon \cdot length(\sigma) + \alpha)$$



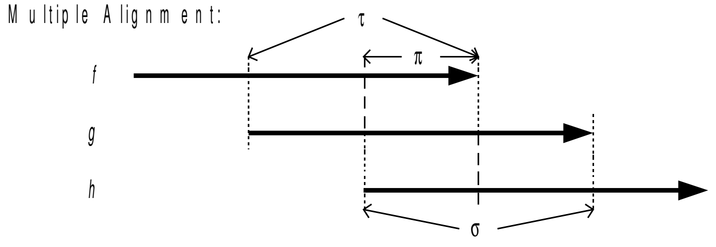
These two clauses of the rule also protect against the case where there are two or more edges between some of the fragments creating a multi-graph (in the example above two overlaps between fragments *f* and *h*). In this case, both of the alternate edges are of the right type as encoded by the *suf* attributes, but only the one with the overlap thickness consistent with the multiple alignment as encoded by the *hang* attributes satisfies these clauses.

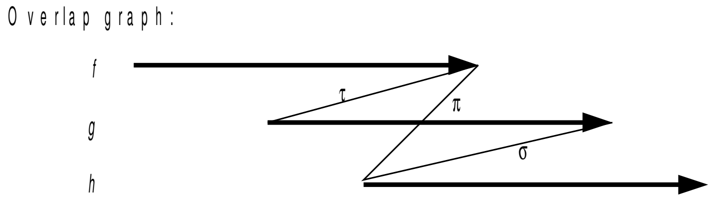
For valid overlaps, we can classify the topological possibilities.

Case 0: Fragments f, g, and h are needed to span the interval, thus fπh is null and there is no π edge to remove.

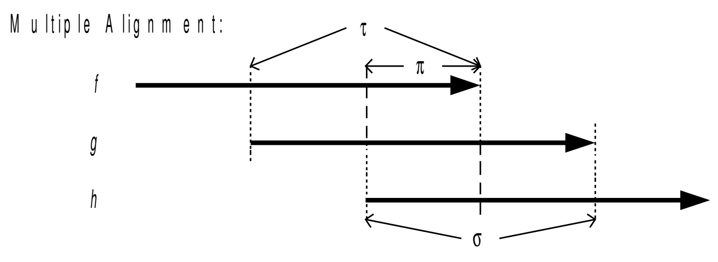


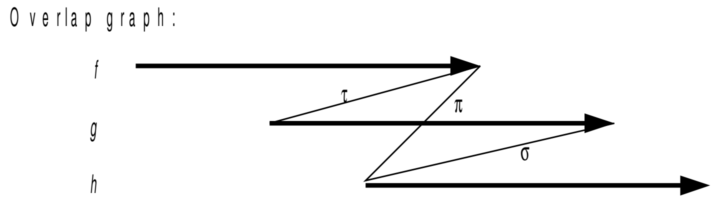
Case 1: Fragments f and h are needed to span the interval and g is the middle fragment and thus can not be contained by f or h.



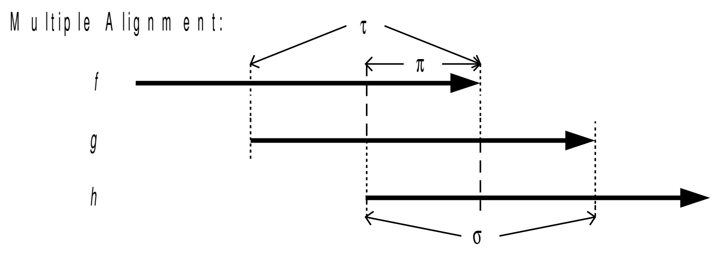


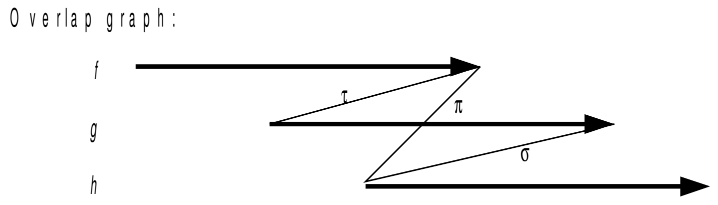
Case 2: Fragments f and g (the middle fragment) span the interval. The fragment h is contained by g.





Case 2: The fragment f spans the interval and fragments g and h are contained by f but not each other


Multiple Alignment:
τ
π
f
g
h
σ


Overlap graph:
f
τ
π
g
σ
h