

CLAIMS:

1. A computer-implemented method of determining at least one optimal alignment of at least part of a first map to at least part of a second map or a plurality of second maps, the first map being a physical genome map and the or each second map being a physical genome map, the method comprising:

receiving first map data indicative of a first ordered list of distances between features of the first map;

receiving second map data indicative of a second ordered list of distances between features of the second map or second maps;

generating, from the second map data, seed data indicative of a plurality of seeds, each seed comprising at least one of the distances in the second ordered list;

generating a plurality of candidate alignments from the seed data by searching at least part of the first ordered list to find at least approximate matches for respective seeds, and extending the approximate matches by dynamic programming;

determining respective alignment scores for respective candidate alignments; and

selecting one or more of the candidate alignments as an optimal alignment or optimal alignments, based on the alignment scores,

wherein the respective alignment scores comprise Z-scores; and

wherein the alignment score for a candidate alignment π is determined according to

$$\vartheta(\pi \in \Pi) = Z\text{-score}(-Z\text{-score}(\pi, \#matches) + Z\text{-score}(\pi, \#cuterrors) + Z\text{-score}(\pi, \text{WHT}(\chi^2, \#matches))),$$

where $\#matches$ is the number of matching distances in the candidate alignment, $\#cuterrors$ is the number of cut errors identified by the alignment in the first map and/or the second map(s), and $\text{WHT}(\chi^2, \#matches)$ is the Wilson-Hilferty Transformation of the χ^2 score for sizing errors.

2. The method according to claim 1, wherein the first map and/or the second map is a restriction map, the features are restriction sites, and the distances are fragment sizes.

3. The method according to any one of the preceding claims, wherein the second map or maps is or are generated from one or more nucleotide sequences.

4. The method according to claim 3, wherein the second map or maps is or are generated by searching for one or more patterns in the one or more nucleotide sequences, and determining distances between successive matches from said searching.

5. The method according to claim 4, wherein each pattern is a restriction enzyme recognition sequence.

6. The method according to any one of the preceding claims, wherein the seeds are composite seeds each comprising a plurality of c-tuples, each c-tuple comprising one or more successive distances and/or one or more sums of successive distances in the second ordered list.

7. The method according to claim 6, wherein c is greater than or equal to 2 for at least some of the c-tuples.

8. The method according to any one of the preceding claims, wherein the dynamic programming and/or the searching of the first ordered list comprises finding a feasible match between a subset of distances of the second ordered list and a subset of distances of the first ordered list.

9. The method according to claim 8, wherein a feasible match is found if the following is satisfied:

$$\left| \frac{\sum_{i=k}^s o_i - \sum_{j=l}^t r_j}{\sqrt{\sum_{i=k}^s \sigma_i^2}} \right| \leq C_\sigma,$$

25

where r_j is the subset of distances of the second ordered list, o_i is the subset of distances of the first ordered list, k and s are beginning and end indices of the match in the first ordered list, l and t are beginning and end indices of the match in the second

ordered list, σ_i are respective standard deviations of the distances in the subset of the first ordered list, and C_σ is a match stringency threshold.

10. The method according to claim 8, wherein a feasible match is found if the following is satisfied:

$$\left| \frac{\sum_{i=k}^s o_i - \sum_{j=l}^t r_j}{\sqrt{\sum_{j=l}^t \sigma_j^2}} \right| \leq C_\sigma$$

- where r_j is the subset of distances of the second ordered list, o_i is the subset of distances of the first ordered list, k and s are beginning and end indices of the match in the first ordered list, l and t are beginning and end indices of the match in the second ordered list, σ_j are respective standard deviations of the distances in the subset of the second ordered list, and C_σ is a match stringency threshold.

11. The method according to claim 9 or 10, wherein C_σ is different for the dynamic programming and the searching of the first ordered list.

12. The method according to any one of the preceding claims, comprising converting the alignment scores to p-values; returning one or more candidate alignments as the optimal alignment(s) if the one or more candidate alignments meet an alignment score threshold and/or a p-value threshold; otherwise, returning no candidate alignments.

13. The method according to claim 12, further comprising assessing statistical significance of the optimal alignment(s).

14. The method according to claim 13, wherein statistical significance is assessed by determining a false discovery rate (FDR) q-value for the optimal alignment and each other candidate alignment.

15. The method according to any one of the preceding claims, further comprising:

generating a plurality of sub-maps from the first map, the sub-maps being overlapping windows of the first ordered list;

for each sub-map, determining one or more optimal alignments of the sub-map to the one or more second maps; and

5 if an optimal alignment for a sub-map is statistically significant, extending said statistically significant optimal alignment by dynamic programming.

16. A system for determining at least one optimal alignment of at least part of a first map to at least part of a second map or a plurality of second maps, the first map being
10 a physical genome map and the or each second map being a physical genome map, the system comprising an alignment software module which is configured to:

receive first map data indicative of a first ordered list of distances between features of the first map;

15 receive second map data indicative of a second ordered list of distances between features of the second map or second maps;

generate, from the second map data, seed data indicative of a plurality of seeds, each seed comprising at least one of the distances in the second ordered list;

20 generate a plurality of candidate alignments from the seed data by searching at least part of the first ordered list to find at least approximate matches for respective seeds, and extending the approximate matches by dynamic programming;

determine respective alignment scores for respective candidate alignments; and

select one or more of the candidate alignments as an optimal alignment or optimal alignments, based on the alignment scores,

wherein the respective alignment scores comprise Z-scores; and

25 wherein the alignment score for a candidate alignment π is determined according to

$$\vartheta(\pi \in \Pi) = Z\text{-score}(-Z\text{-score}(\pi, \#matches) + Z\text{-score}(\pi, \#cuterrors) + Z\text{-score}(\pi, \text{WHT}(\chi^2, \#matches))),$$

where $\#matches$ is the number of matching distances in the candidate alignment, $\#cuterrors$ is the number of cut errors identified by the alignment in the first map and/or the second map(s), and $\text{WHT}(\chi^2, \#matches)$ is the Wilson-Hilferty Transformation
30 of the χ^2 score for sizing errors.