Automated Composite Depth Scale Construction and Estimates of Sediment Core Extension

Lorraine E. Lisiecki

Department of Earth Sciences, Boston University, Boston, Massachussetts, USA

Timothy D. Herbert

Department of Geological Sciences, Brown University, Providence, Rhode Island, USA

Appendix B: Technical Modifications to LL02

Here we present a technical description of the modifications made to the LL02 algorithm to perform realistic graphic correlation across core breaks. (Source code is available as supplemental material and at http://lorraine-lisiecki.com.) The LL02 algorithm divides the data from each hole into many small, evenly sized depth intervals (∼25 cm for interhole alignment) and uses dynamic programming to evaluate the fit for all possible alignments of intervals using a range of possible matching ratios (e.g., two intervals from Hole A aligned to three intervals from Hole B) to simulate differences in sedimentation rate or extension between the holes.

The original version of the algorithm linearly interpolated core properties across gaps (using the mbsf depth scale) and penalized for differences in stratigraphic features between holes based on that interpolation. This technique was robust enough to find an approximate mapping between two holes but produced some distortion around large gaps and coring overlaps [*Lisiecki and Lisiecki,* 2002]. The updated algorithm specifically identifies core breaks in both records, which permits a greater range of gap lengths, including negative gaps in the case of coring overlaps. Gap lengths are now constrained based on core alignments (with no interpolation across gaps) and an adjustable "gap size" penalty.

The structure of the updated algorithm is illustrated in Figure S2 by an array of subgrids for each pair of cores from the two holes. These matrices store "scores" for the best possible alignment path to get from the top of a hole to the pair of depths specified by that grid point in the matrix. Scores are calculated sequentially using dynamic programming and optimize alignments based on the fit between holes and various penalty functions.

As in the LL02 algorithm, a third dimension (not shown in Figure S2) is used to evaluate alignments at different relative sedimentation rates (interval matching ratios) between the two holes, and sedimentation rates are allowed to change (with a speed-change penalty) between each set of matched intervals. As suggested in LL02, we also apply a "speed" penalty proportional to the difference between the current matching ratio and a 1:1 ratio. This helps reduce the amount of extension/compression between records, which is particularly helpful because holes from a single site should have similar sedimentation rates at time-equivalent depths.

As in the LL02 algorithm, the first step of the algorithm is to initialize the top and left edges of the matrix with endpoint/no-match penalties for alignments which produce mismatches in the specified start depths of the two holes. Next, the algorithm sequentially optimizes alignment scores for each core-pair subgrid. Scores for the top and left edges of each interior subgrid are initialized (see below) based on a gap-size penalty and the scores of previously solved subgrids. The cumulative alignment scores within the subgrid are minimized using the same dynamic programming technique employed by LL02. After all subgrids have been filled, endpoint/no-match penalties are added along the bottom and right edges of the matrix for alignments which produce mismatches in the specified end depths of the two holes. The optimal alignment between holes is selected by identifying the alignment which produces the minimum cumulative score at the end of either hole (i.e., the bottom or right side of the grid array).

The scores used to initialize the top and left edges of each core-pair subgrid are calculated by optimizing the sum of the cumulative score from the previous core end and a gap-size penalty, which constrains the length of stratigraphy missing between core breaks. As an example, we describe the alignment of the top of core B2 to some point in core A3 (i.e., initializing the left edge of subgrid A3-B2 in Figure S2). A symmetric procedure applies to aligning Hole A core tops to some depth in Hole B (i.e., initializing the top edge of each subgrid). Thin gray lines in Figure S2 illustrate some of the gaps which are considered for A3-B2 and A2-B3 alignments.

Each point on the left edge of subgrid A3-B2 represents the alignment of the top of core B2 to a different depth along core A3. To find the best possible score for the alignment of the B2 core top to a particular depth in A3, we add the best cumulative score for each possible B1 end depth (stored in the right edge of B1 subgrids) to a gap-size penalty based on the gap length for that particular alignment of the end of B1 and top of B2. The minimum sum is selected as the optimized cumalitive score of that B2 start depth and is stored in the corresponding grid point on the left edge of A3-B2.

The gap-size penalty for each possible alignment is proportional to the square of the difference between the gap length estimated in the Hole B mbsf depth scale and the length of Hole A stratigraphy (in the mbsf scale of Hole A) spanned by the gap. The multiplicative factor for this penalty is a user-defined parameter which can be adjusted based on the accuracy of mbsf gap size estimates (e.g., due to ship motion during drilling).

The alignment algorithm considers all positive core gap lengths, including gaps which extend the entire length of a hole. The occurrence of extremely large gaps in the "optimal" alignment is controlled by the user-defined gap-size and endpoint/no-match penalty weightings. However, the algorithm does constrain the possible lengths of core overlaps. An overlap occurs if, for example, the top of core B2 is aligned to a depth in Hole A above the end of core B1 because stratigraphy appears to be duplicated in cores B1 and B2. To simplify our optimization algorithm, we constrain the length of core overlaps by requiring that the duplicated stratigraphy may not extend across more than one core in Hole A. For example, when the algorithm considers paths which align the top of core B2 to some point in core A3, it will not consider alignments in which core B1 ends in core A4. (Conversely, paths which align the end of B1 to A4 cannot include the alignment of core B2 above core A4.) This allows the algorithm to calculate the optimal cumulative scores for each subgrid sequentially, solving from left to right and top to bottom.

We additionally require that gaps from the two holes cannot overlap because this configuration permits no constraint on the amount of stratigraphy which may be missing within the overlapping gaps. The algorithm can approximate overlapping gaps by aligning the end of the core from one hole to the start of the next core in the other hole (e.g., cores A3 and B4 in Figure S2, which generate gaps e and f). Because such alignments may actually correspond to missing stratigraphy, every attempt should be made to avoid including them in the composite section. If no hole spans the potentially overlapping gaps, the continuity of the composite section cannot be verified. If the length of missing stratigraphy can be estimated (e.g., from downhole log data), a manual correction should be made to the composite depth scale.

References

Lisiecki, L. E., and P. A. Lisiecki (2002), Application of dynamic programming to the correlation of paleoclimate records, *Paleoceanography, 17,* doi:10.1029/2001PA000733.

Lorraine E. Lisiecki, Department of Earth Sciences, Boston University, 675 Commonwealth Avenue, Boston, Massachusetts 02215, USA. (lisiecki@bu.edu)

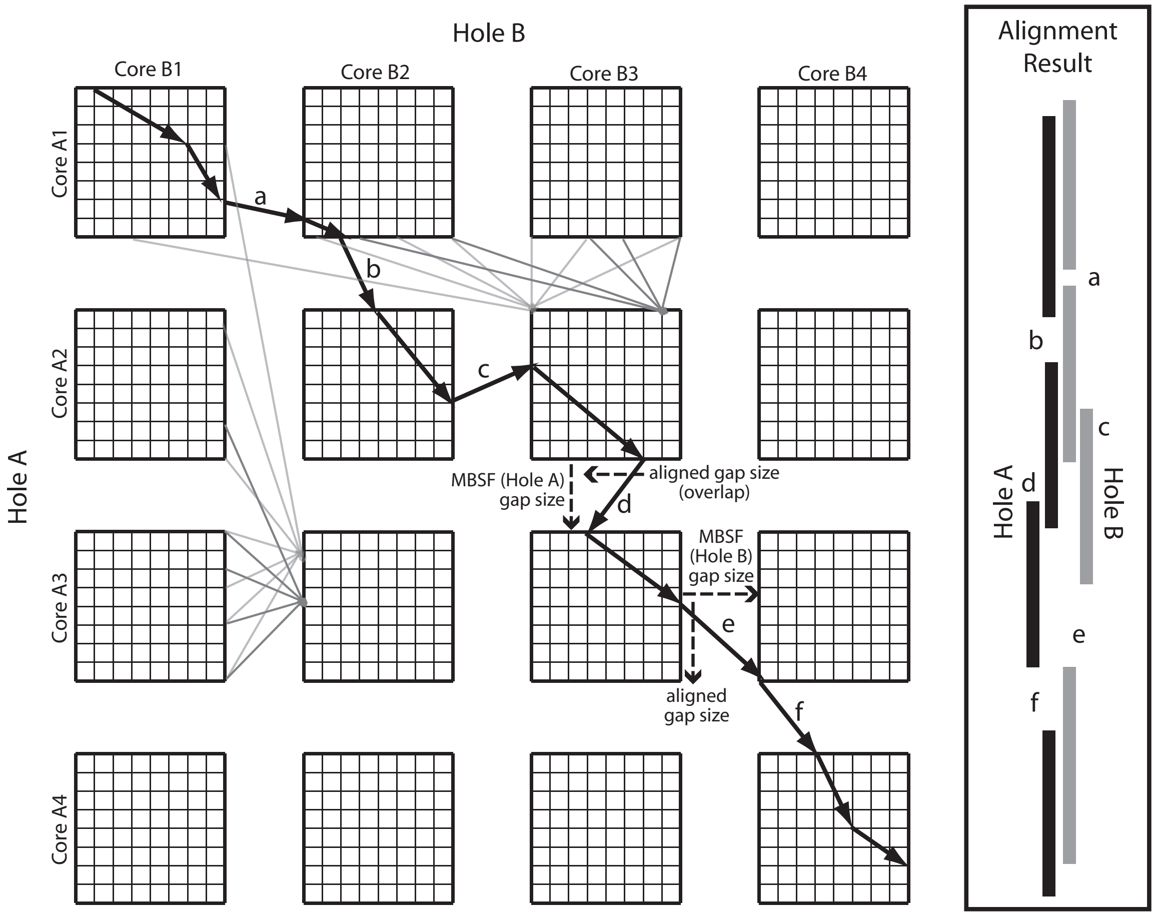

Figure 1. Dynamic programming algorithm for determining optimal interhole core alignment and gap size (see Appendix B). The inset on the right shows the relative alignment of cores with gaps labelled to indicate their position in the alignment matrix.