**TAD boundaries definition**

TAD boundaries were defined using a custom algorithm based on the assumption that there are few interactions between elements located before and those after the boundaries. These interactions are measured by sliding a diamond of width *w* (Figure 7A) on every position *d* (between *1+w* and *n-w*) along diagonal of the square matrix *M* of *n* dimensions and computing the sum in the diamond at each position. The limits of *d* were set to prevent the diamond from exceeding the limits of the matrix. At each position *d*, the sum of all values in the diamond are stored in a vector *V*. This can be rewritten as:

\vspace{0.2in}

$\left\{\begin{matrix}1\leq w\leq \frac{n}{2} \\ \forall d\in\left \{ 1+w , ... , n-w \right \}\end{matrix}\right. V\_{d}=\sum\_{i=d-w}^{d}\sum\_{j=d}^{d+w}M\_{i,j}$

\vspace{0.2in}

\noindent The size of the diamond (*w*) has been set to an arbitrary threshold of 100kb, considered reasonable as the median length of filtered TADs is 140kb.

The sums from the diamond are then used to find boundaries. For all TADs, boundaries are extended inwards from the borders as long as the value of *V* does not exceed an arbitrary threshold defined as the starting value (at the TAD border) plus 10\% of the maximum value in the TAD (Figure 7B).The boundaries were filtered afterwards to remove all those extending beyond their TAD. This happened in cases where the border was already among the highest values in the vector *V*. This shows the algorithm is not optimal and would need be improved to properly process the boundaries in these special cases. Lowering the threshold would reduce the number of boundaries that fail to stop inside their TAD, but it would also reduce the length of all boundaries. To accurately define boundaries, one would need a more complex algorithm that takes additional factors into account.

**Figure 7 :** Visual representations of the algorithm used to compute TAD boundaries in Hi-C matrices. **A.** A diamond (blue) of width w set to 100kb is slid on all position (*d*) along the diagonal. For each position *d*, the sum of all values in the blue diamond (*Vd*) is computed, representing a measure of all interactions across position *d* (i.e. elements upstream of *d* contacting those downstream of *d*) at a maximum range of *w* (blue loops at the top). In other words, the diamond measures contacts between regions [d-w;d] (upstream) and [d;d+w] (downstream) (blue), excluding interactions happening inside those regions (orange). **B.** Example of the calculated sums of interactions in vector *V* through a TAD. *Vd* represents the sum in the diamond at every position along the matrix diagonal. Boundaries were extended inwards from TAD borders until *Vd* reached an arbitrary threshold defined as the value of *V* at the border, plus 10\% of the maximum value of *V* inside the TAD. The solid vertical lines represent the TAD borders, the horizontal dashed lines represent the thresholds required to stop extending boundaries and the transparent areas represent the final boundaries. All blue elements relate to the left side, while all green elements relate to the right side.