Appendix A: Tiling selection code

In order to track the highest scoring tiling along with its score, let us associate a set of tiles and a score in an ordered pair \( s = (a, T) \). Here, \( a \) is the score and \( T \) is a (possibly partial) tiling, meaning that it is a set of non-overlapping tiles. When the tiling is partial, the tiles do not tile the entire space. The notation \( s_{\text{score}} \) and \( s_{\text{tiling}} \) will be used to refer to the elements of the ordered pair. The notation \( T(x, R_L, R_H, C_L, C_H) \) will be used to represent a tile with outbreak state \( x \) (0 or 1) and boundaries \( R_L, R_H, C_L, C_H \). \( \emptyset \) denotes the empty set.

Tiling selection algorithm:

1. For a grid of size \( R \times C \)
2. Track best hypotheses: best\( R \), best\( C \)
3. Keep caches \( rCache \), \( cCache \)
4. Init:
   - \( rCache[0] \leftarrow (1, \emptyset) \)
   - \( cCache[0] \leftarrow (1, \emptyset) \)
5. for \( R_H \) from 1 to \( R \)
6.   best\( R \) \leftarrow (0, \emptyset)
7.   for \( C_H \) from 1 to \( C \)
8.     for \( C_L \) from 1 to \( C_H \)
9.       \( x \leftarrow \arg \max_{x \in \{0, 1\}} \text{score}(x, R_L, R_H, C_L, C_H) \)
10.      \( s \leftarrow (\text{score}(x, R_L, R_H, C_L, C_H), \{T(x, R_L, R_H, C_L, C_H)\}) \)
11.     if \( \text{bestC}_{\text{score}} < cCache[C_L - 1]_{\text{score}} \cdot s_{\text{score}} \) then
12.       \( \text{bestC} \leftarrow (cCache[C_L - 1]_{\text{score}} \cdot s_{\text{score}}, cCache[C_L - 1]_{\text{tiling}} \cup s_{\text{tiling}}) \)
13.     end if
14.   end for
15. end for
16. if \( \text{bestR}_{\text{score}} < rCache[R_L - 1]_{\text{score}} \cdot \text{bestC}_{\text{score}} \) then
17.   \( \text{bestR} \leftarrow (rCache[R_L - 1]_{\text{score}} \cdot \text{bestC}_{\text{score}}, rCache[R_L - 1]_{\text{tiling}} \cup \text{bestC}_{\text{tiling}}) \)
18. end if
19. \( rCache[R_H] \leftarrow \text{bestR} \)
20. end for
21. return best\( R \)