1 Constraint Programming (CP)

1.1 Mathematical Model

The **CP** mathematical model, is encoded with Constraint Programming [2]. This model is valid for $m = 1$ only and requires integral $A_{i,j}$ values due to the implementation (described in Subsection 1.2). It is based on introducing variables M_i where $M_i = j$ if genomic bin i interacts with genomic bin j, and $M_i = i$ otherwise. The goal of this model is to solve M_i for all i. The model is given in Mathematical Model 1. Since this model encodes a combinatorial problem, its time complexity is exponential in the worst case.

maximize

$$\sum_{i \in V} A_{i,M_i} \tag{1}$$

subject to:

$$M_i = j \leftrightarrow M_j = i, \ \forall i, j \in V \tag{2}$$

$$M_i \in \{i\} \cup \{j \mid A_{i,j} > 0\}, \ \forall i \in V \tag{3}$$

Mathematical Model 1: The **CP** model, valid for $m = 1$ and integral interaction frequencies ($A_{i,j}$) only. V is the set $\{1, \ldots, N\}$ representing the genomic bins.

1.2 Implementation

The **CP** mathematical model (depicted in Mathematical Model 1) was implemented in MiniZinc [1] with the OR-Tools constraint solver from Google [1]. An example MiniZinc program (Additional File 1) and a corresponding example data file (Additional File 2) with the integral interaction frequencies from the hypothetical whole-genome contact map depicted in Figure 3A of the corresponding manuscript are provided in the supplementary folder. This model leverages the fact that the solution will never contain more than $m \times N$ interactions making it scalable to larger genomes in terms of space complexity. It is worth noting that Equation (2) can be encoded by the `inverse` global constraint [2], whereas Equation (1) is encoded with one `element` constraint per row of A plus one `sum` constraint. These constraints are propagated by efficient algorithms in many constraint programming solvers.

1.3 Results

The MiniZinc program corresponding to the complete fission yeast genome could not be solved to optimality after several days of run time on a server-grade computer. In an attempt to overcome this, the divide-and-conquer approach described above was applied. A MiniZinc

[1] `https://developers.google.com/optimization/`

[2] `http://www.minizinc.org/doc-lib/doc-globals-channeling.html`

program for each *cis*- or *trans*- subproblem was generated and run independently. Similarly to the complete whole-genome contact map, not a single *cis*- or *trans*- problem could be solved to optimality in several days.

References

[1] Nicholas Nethercote, Peter J. Stuckey, Ralph Becket, Sebastian Brand, Gregory J. Duck, and Guido Tack. MiniZinc: Towards a standard CP modelling language. In Christian Bessière, editor, *Principles and Practice of Constraint Programming – CP 2007*, LNCS, pages 529–543, Providence, RI, USA, September 23-27 2007. Springer Berlin Heidelberg.

[2] F. Rossi, P. van Beek, and T. Walsh, editors. *Handbook of Constraint Programming.* Elsevier, New York, NY, USA, 2006.