Algorithm 1 BAPSO algorithm

Input: The number of genes in the GRN (N); the maximum number of iterations of BAPSO ($maxit$); and the size of the swarm in BAPSO (ps)

Output: The weight matrix representing the structure of the inferred network (F)

for gene $g = 1$ to N **do**
  Initialise position vector, $\mathcal{P}^g = [p_i^g]_{1 \times ps}$ randomly and velocity vector, $\mathcal{V}^g = [v_i^g]_{1 \times ps}$ to $\emptyset$.
  Each element, p_i^g (and v_i^g) is defined as: $p_i^g = [f_{i1}^g, f_{i2}^g, \cdots, f_{iN}^g, \delta_i^g, \epsilon_i^g, \mu_i^g]$, where N is the number of genes.
  Calculate the fitness, er_i^g, of each of the particles in the swarm using Algorithm 2 and store them in the fitness vector, $\mathcal{E}^g \leftarrow [er_i^g]_{1 \times ps}$.
  Store the minimum (best) fitness, $fit_{best} \leftarrow minimum(\mathcal{E}^g)$ and its index in min.
  Set *personal best solutions*, $\mathcal{PB}^g = [pb_i^g]_{1 \times ps} \leftarrow \mathcal{P}^g$ and their corresponding fitness, $\mathcal{PE}^g = [per_i^g]_{1 \times ps} \leftarrow \mathcal{E}^g$.
  Calculate the *global best solution*, $gb^g \leftarrow pb_{min}^g$.
  for $iter = 2$ to $maxit$ **do**
    Update all particle velocities, $\mathcal{V}^g$ using (9).
    Update all particle positions, $\mathcal{P}^g$ using (10).
    Update the fitness of the swarm, $\mathcal{E}^g$, using Algorithm 2.
    Update $\mathcal{PB}^g$, $\mathcal{PE}^g$, gb^g, fit_{best}, and min using Algorithm 3.
  end for
  Store gb^g at the end of $maxit$ iterations.
end for
Combine the stored gb^g (for $1 \le g \le N$) to get an $N \times (N+3)$ matrix.
Extract the first N elements from each row to get an $N \times N$ matrix, $[f_{ij}]_{N \times N}$.
Return $F \leftarrow [f_{ij}]_{N \times N}$.

the received waves. In BA, the virtual bats are assumed to possess random flight, and in each generation, some of the bats are randomly designated to perform local search

Algorithm 2 Fitness calculation of particles, i.e. obtaining the predicted time-series using Half-systems.

Input: The time-series gene expression dataset (X); the gene being considered (g); and the particle positions ($\mathcal{P}^g$)

Output: Fitness of the swarm ($\mathcal{E}^g$)

Extract number of genes, N, from X.
Extract number of timepoints, tp, from X.
Extract population size, ps, from $\mathcal{P}^g$.
for $i = 1$ to ps **do**
  Extract $[f_{ij}^g]_{1 \times N}, \delta_i^g, \epsilon_i^g, \mu_i^g$ from p_i^g.
  for $t = 2$ to tp **do**
    Calculate the predicted expression level, $\hat{x}_i^g(t)$ of gene g from the original expression level at the previous timepoint, i.e. $x_i^g(t-1)$ using (6).
  end for
  Calculate the fitness of particle p_i^g and store it in er_i^g using (11).
end for
Return $\mathcal{E}^g \leftarrow [er_i^g]_{1 \times ps}$.

Algorithm 3 Updating personal and global best solutions in BAPSO.

Input: The particle positions ($\mathcal{P}^g$); the *personal best solutions* ($\mathcal{PB}^g$), and their fitness ($\mathcal{PE}^g$); the *global best solution* (gb^g), its fitness value (fit_{best}), and index (min); and the fitness vector ($\mathcal{E}^g$)

Output: Updated $\mathcal{PB}^g$; $\mathcal{PE}^g$; gb^g; fit_{best}; and min

Extract population size, ps from $\mathcal{P}^g$.
for $i = 1$ to ps **do**
  if $er_i^g < per_i^g$ **then** ▷ *update the personal best solutions*
    $per_i^g \leftarrow er_i^g$ ▷ where $er_i^g \in \mathcal{E}^g$ and $per_i^g \in \mathcal{PE}^g$
    $pb_i^g \leftarrow p_i^g$ ▷ where $pb_i^g \in \mathcal{PB}^g$ and $p_i^g \in \mathcal{P}^g$
  end if
  if $er_i^g < fit_{best}$ **then** ▷ *update the global best solution*
    $fit_{best} \leftarrow er_i^g$ ▷ where $er_i^g \in \mathcal{E}^g$
    $gb^g \leftarrow p_i^g$ ▷ where $p_i^g \in \mathcal{P}^g$
    $min \leftarrow i$
  end if
end for
Return updated $\mathcal{PB}^g$, $\mathcal{PE}^g$, gb^g, fit_{best}, and min.