S16: BP Implementation considerations.

The main manuscript covers all of the theoretical considerations of our BP algorithm. However, there are a few implementation issues that affect the efficiency of the algorithm.

Computational calculation of Equation 14.A for each update requires substantial memory and computational power. The algorithm has to keep track of all M x(N -1) factors, and then perform *M* multiplications, (*N-*1) times in preparation for a single update. We take advantage of the multiplicative relationship between a marginal P(wij ) and its factors , to reduce the number of computations need to evaluate and eliminate precision error associated with many multiplications of small numbers.

The marginal distribution and the global information are related by a simple division of the marginal by the cavity factor (Equations S1A and S1B). In log space, the same calculation is just a simple subtraction (Equation S1C).

**Equation S1**



The calculation in equation S1C is a fast calculation and involves only an inverse log operation, one subtraction and the cost of doing the normalization (K additions and K division operations). The calculation needs only to keep track of all factors and the sum of the natural logarithms of the factors across all M conditions, stored in an array .

**Equation S2**



An additional advantage for performing the calculation in log space is that the operation does not risk precision errors from the multiplication of many small numbers. The calculation is a subtraction of negative numbers followed by a natural antilogarithm calculation, neither of which risk precision error. In non log space, the calculation requires M multiplications of probability distributions, which contain some very small numbers. In fact, an earlier implementation of the BP algorithm experienced detrimental errors for even small M. This error cascaded and produced meaningless results. The log space calculation is robust up to about at least M=1,000, above which has not been tested.

**Analytical approximation of the local factor update**

The integral in Equation 14B is not solvable analytically due to the nonlinearity introduced from the sigmoid function . One can solve the integral numerically with a method such as trapezoidal integration. However, since Equation 14B has to be calculated NxM times per iteration over many iterations, numerical solutions of this kind dramatically slow down the computational time to convergence. To alleviate the cost of this calculation, we use a simple algebraic rearrangement as shown in Equation S4B to make the fitness term  linear with the integration variable .

The goal is to rewrite the error inside the sum of squares term to be linear in . The fitness is maximum when the cost is zero and therefore , which in turns means that . Inverting the nonlinear function on both sides of the equation (Equation S3A) and yields an approximation of the fitness that is linear with .

**Equation S3**



Equation S4 is the analytical solution to the integral in Equation S3B.

**Equation S4**



**The Algorithm**

See Figure-S16 for an algorithm flow-chart.