Problem	Solution(s)	Assumption(s)
When applying mutation to BCR (4 Digit Code), it is not clear how to decide where to plus or minus one.	If both are possible, apply either with equal chance of occurring. If only one is possible, apply that change.	Both options equally likely.
We require a probability distribution to determine the turning angle. Distribution not found in the cited references.	Can estimate it using a Random Variable.	Turning angle is equally likely to go "left" or "right"
In algorithm 3: "with probability persistentLengthTime(C.type)". In parameters, all values but one of these parameters have a value less than 1.	Value is given in terms of minutes. We will convert this to minutes so that all cases have a probability less than one. Worth noting this probability will be tiny after the conversion.	-
In algorithm 9: "get new BCR, randomly or from a predefined repertoire".	Uniformally generate BCR values for now.	-
Require amount of CXCL12 and CXCL13 at each position within the germinal center.	Use Guassian or Uniform random variable to assign temporary values.	Assumed values are distributed similar to critical values given in table of values
	Assign value near critical value to all positions in cell.	Assumed the germinal center starts off with a constant amount throughout.
pDif2Out(time) has a formula with a variable that's always zero.	-	-
In algorithm 9 it is not clear whether the FDC branches remain within the Light Zone.	Forced branches to be only within light zone.	As suggested by diagrams at the start of the paper, the FDC branches cannot leave the Light Zone.
Algorithm 10 contains two function that have not been described.		
The initial polarity of each cell is unknown.	Randomly assign the polarity.	
Algorithm 3. The move function has a vector called North. North is a vector pointing towards the light zone and is not defined. This is peculiar as it only influences T cells and they are already in the Light Zone.	Light zone is the bottom of the sphere, could use $(0, 0, -1)$.	
Algorithm 3, in the move function we must not move the cell against the polarity. Hard to determine this.	We find the best possible points the cell can move to, in order and only consider the first 8 points, taking the best possible. If none of these points are free, the cell does not move.	

Not explicitly stated if a cell is able to divide and have new cell a diagonal movement away. Similar for if we want to know if one type of Cell is next to another.	Allow cells to divide into positions diagonally away.	
It is not clear how to determine where a cell is IAmHighAg and it does not state where a cell starts off as True or False.	Initially start all cells if IAmHighAg set to False.	
It is not specified whether the 4 digit BCR value can start with a zero. Algorithm 1 suggests it can while figure 4 suggests it cannot.	We will not let the value start with zero. Python will automatically convert it to a 3 digit number and that will cause issues.	Assume it cannot start with a zero.
Algorithm 4, sampling Gaussian Distribution to find a value for sep, the mean is given but standard deviation is not.	Let standard deviation be one.	
Algorithm 5, C.clock suggest we add one to the clock counter. Next line says to test if C.clock > testDelay, which will always be true since testDelay is 0.02 .	Changed to C.clock increment by dt.	Assumed algorithm has a type. From notes, can somewhat assume C.clock is supposed to be C.tcClock which increments by dt and is not mentioned in the algorithms.
Algorithm 5, lines 8 and 9. What if he have multiple fragments next to the cell?	Found the maximum expression for the formula given on line 9.	Can only bond with one and the antigen amounts from multiple neighbouring fragments would not add. Bonds with the one with most antigen.
Algorithm 5 has two main if statements. The first if statement can change the state of a cell such that it also triggers the second if state. Same problem in algorithm 6.	Using an If and else if statement.	Assumed that only one if statement should be carried out since they both increment a time variable by the time step, dt. Would not make much sense having that increment twice in one time step.
Algorithm 5, line 31: AntigenAmount[f]- -, Antigen amount seems to be a float but this code says to remove one from it, seems strange.	Follow what it says exactly, subtract one from it.	Assume that a cell can take a certain amount of antigen from a fragment and without loss of generality, we can set this quantity to be one.
Algorithm 6, line 13. Refers to B cells having antigens.	Treat this as amount of retained antigen.	
Algorithm 6, line 13 also does not say what to do if a B cell is in contact with multiple T cells.	We will only consider the first T cell in contact with a B cell in the first main if statement of the algorithm. This is not a great solution and will probably need refining.	

When calculating affinity, we need to find hamming distance between B cell BCR and Antigen. Diagram suggests that we use something slightly different to the hamming distance.	We will use the traditional hamming distance.	
Not given BCR values or antigen value.	Randomly generate 1000 four-digit values to act as BCR values. Let the antigen value be 1234 as a placeholder.	
Algorithm 10, lines 18, 19, 20. We calculate transfert then subtract it from NumBCROutCells and NumBCROutCellsProduce. This calculation is almost surely going to end in a float.	Floor the result to return an integer.	
Move function has no specifications forcing certain types of cells in the light or dark zone.	Not a problem, the boundaries are loosely defined.	
Algorithm 10, line 10 refers to a variable d_t, not referenced elsewhere.	Let it be dt.	Assume typo was made.
Does not state whether T cells are responsive to CXCL12 and CXCL13, which is required for move function.	Set both to False	Assume unresponsive.
	T Cells may not be required to pass through part of move function requiring these values (which would correspond to both values being False). Seems possible due to poor documentation.	