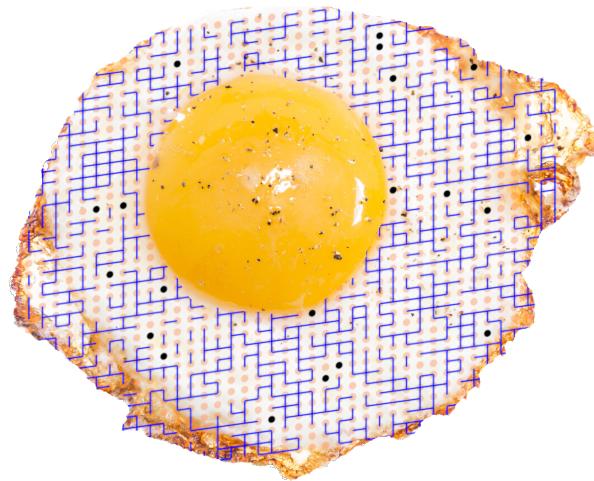


Applying a 2D Bond Percolation Model with Defects to Determine Egg Cooking Temperatures



Authors: Alex Cohen, Jonathan Zheng

Date: December 9, 2020

1 Introduction: Protein denaturing and crosslinking

1.1 Background

Protein denaturing and crosslinking are thermodynamic processes that strongly determine protein properties. At physiological temperatures, proteins are often folded in a low-energy state. At higher temperatures, thermal energy can cause proteins to denature, unfolding into random coil configurations. Crosslinking involves interactions between denatured proteins and can result in networks of proteins binding together. The process of cooking an egg can be described by studying protein denaturation and crosslinking. As an egg cooks, its proteins denature and crosslink. Once these linkages extend throughout the bulk of the system, the egg white turns into a hardened material that can be eaten. A schematic of this process is shown in Figure 1.

1.2 Overview of model

In this project, we model how an egg cooks by examining how proteins in an egg white denature and crosslink. Egg white is composed mainly of water (90% by mass) and protein, 54% of which is *ovalbumin* [1]. Hence, we model the egg white as a solution of ovalbumin protein in water, in which ovalbumin denaturing is described by a statistical mechanical model of a two-state material with a degeneracy associated with the high energy state. Then, we model protein crosslinking using a 2D bond percolation model with both theory and simulation. Finally, we connect these two processes and analyze how the competition between cooking temperature, denatured proteins, and crosslinking influences the critical transition to a fully percolated protein network, or an edible egg.

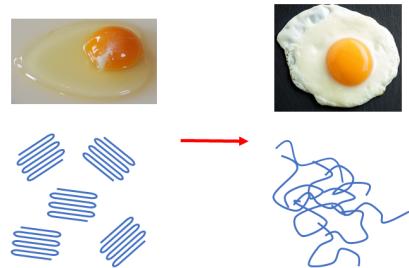


Figure 1: Diagram of protein denaturing and crosslinking during cooking.

2 Molecular model

2.1 Protein denaturing

Protein denaturing is the process through which a protein unfolds from a low-energy folded structure to a random coil. Egg proteins are typically denatured with heat, as increasing thermal energy enables the thermal motion of the atoms to overcome forces which maintain the folded shape. We consider each protein as existing in one of two states: folded or denatured. The folded state has a favorable, low energy which we define as 0, while the denatured state has higher energy ϵ . A microstate of a system of N proteins is fully defined by N occupation numbers n_i such that

$$E_\nu = \sum_{i=0}^N \epsilon n_i \quad \Bigg| \quad n_i = \begin{cases} 0 & \text{if folded} \\ 1 & \text{if denatured} \end{cases} \quad (1)$$

We must also consider the degeneracy of the two states. It is a remarkable fact of biology that specific proteins always fold into unique native states; thus, we consider the degeneracy of the folded state to be 1. For the higher-energy state, denatured proteins can be modeled as a 3D random walk

on a cubic lattice, of which there are many degenerate states. We estimate this degeneracy by considering a lattice model and enumerating the number of ways to arrange a protein. The result from this enumeration is shown in Equation (2), where M is the length of the protein in the number of persistence length units, and $z = 6$ in a 3D cubic lattice.

$$g = z(z - 1)^{M-2} \quad (2)$$

We can derive the canonical partition function for the system as (see **Appendix A**):

$$Q = (ge^{-\beta\epsilon} + 1)^N \quad (3)$$

One quantity of interest when cooking an egg is the fraction of proteins that are folded at a given temperature. Proteins need to be denatured to crosslink, so more folded proteins hinder the ability for the egg to cook. The average fraction of folded proteins is given in Equation (4).

$$\langle f_F \rangle = 1 - \left(\frac{6 \cdot 5^{M-2} e^{-\beta\epsilon}}{1 + 6 \cdot 5^{M-2} e^{-\beta\epsilon}} \right) \quad (4)$$

2.2 Percolation theory for protein-protein bonding

We can consider protein-protein bonding using percolation theory [2]. Percolation theory describes how networks form between sites or bonds. When a continuous cluster extends throughout an entire system, the cluster is said to be *percolating*. This model predicts second-order phase transitions, which we are principally interested in estimating. The theory also enables estimation of scaling behavior near the critical point (see **Appendix A** for more details on scaling).

We model crosslinking proteins using percolation theory. To simplify our analysis, we examine percolation on a 2D square lattice where each bond has an independent probability to be occupied. Each edge connecting adjacent nodes in Figure 2 represents a potential bond, while each node represents a protein. This representation simplifies the system by assuming each protein can bond to at most four other proteins.

When a crosslinked network forms, the egg white hardens as proteins lock to their neighbors. We assume that the second-order transition to a percolated network serves as a proxy to estimating when a crosslinked network forms, or when an egg is cooked. The probability that a bond site is occupied by a bond, p , determines whether a system percolates. When $p > p_c$ (the critical occupation probability), the system is in an equilibrium percolated phase [3].

Renormalization group (RG) theory can be used to derive estimates for the critical point p_c . RG theory involves coarse-graining (CG) the lattice at a specified length scale in order to ignore fluctuations at smaller length scales. As the system repeatedly undergoes CG, the coupling constants (in this system, analogous to the bond probability) change. Based on the CG length scale, we can define a Renormalization Operation (RO), R_l , according to Equation (5).

$$p' = R_l(p) \quad (5)$$

R_l specifies how the bond probability changes when we CG at scale l . Equation (5) has fixed points that satisfy $p^* = R_l(p^*)$ where the system does not change with CG. Since CG with length l changes the correlation length so $\xi' = \xi/l$, and because ξ must remain constant at fixed points, ξ must equal 0 or ∞ at fixed points. $\xi = 0$ corresponds to trivial fixed points: phases with complete disorder or order. $\xi = \infty$ corresponds to the critical point, when fluctuations diverge. Therefore, solutions to $p^* = R_l(p^*)$ give the bond occupation probabilities at trivial fixed points (when all or no bonds are occupied) and critical fixed points (the transition to percolating).

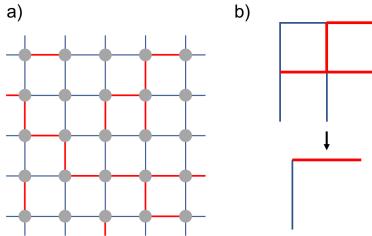


Figure 2: a) Bond percolation on a 2D lattice. Gray dots represent proteins and edges represent bond sites; red bonds are occupied and blue lines are not occupied. b) RO; top has a horizontal spanning cluster but no vertical spanning cluster, so the CG view (bottom) only has a horizontal bond.

We can solve for p_c by creating an RO that we call the Spanning Cluster Rule (SCR) [4] (*note*: we compare this with another RO in **Appendix A**). SCR involves CG the lattice such that regions with 2×2 horizontal bonds and 2×2 vertical bonds become regions with 1 horizontal bond and vertical bond ($l = 2$). In this 2×2 region, we determine if there is a horizontal spanning cluster. If there is a horizontal spanning cluster, the CG representation will have an occupied horizontal bond. The same process is repeated for the vertical bonds. An example of this procedure is outlined in Figure 2b.

To find $R_l(p)$ for the RO, we calculate the probability of finding a horizontal spanning cluster in the 2×2 region, as this is the probability of finding a single horizontal bond in the CG representation. The details of this calculation are included in **Appendix A** and the result is cited in Equation (6).

$$p' = 2p^5 - 5p^4 + 2p^3 + 2p^2 \quad (6)$$

Solutions of Equation (6) indicate that the fixed points are $p_c = 0, \frac{1}{2}, 1$. The critical fixed point is $\frac{1}{2}$, in agreement with exact theoretical results [4].

To further model this application, we also developed a probabilistic simulation described in **Appendix B**. The simulation randomly generates a lattice according to a specified probability of bond formation and calculates properties of the network averaged over several samples, including ratio of percolation (see **Appendix A** for visualizations). As shown in Figure 3, we observed a p_c of $\frac{1}{2}$ for this square lattice model, which corresponds precisely to the results described above.

2.3 Competition between denaturation and crosslinking with temperature

The model discussed above assumed that all sites are equally likely to bond. However, only denatured proteins can form crosslinks with other proteins; at any finite temperature, there will be a fraction of non-denatured proteins. Therefore, our model must account for sites that cannot bond, sites we will call *defects* [5, 6]. The temperature-dependent probability that any site will be a defect was previously derived in Equation (4) as the fraction of proteins which do not denature. We first modify the RG theory to extend the SCR to include defects.

The CG region that we considered previously consisted of a 2×2 region of horizontal and vertical bonds and sites. In order to derive a new RO, we consider the probability of finding 0 to 4 defects in the CG region, then account for each number of defects in the probability of finding a spanning cluster. For example, the chance that one specific site in a CG region is defective is $p(1 \text{ defect}) = P_D(1 - P_D)^3$. The derivations are shown in **Appendix A** and the result for the new RO is given in Equation (7), where P_D represents the probability of finding a defect at an individual

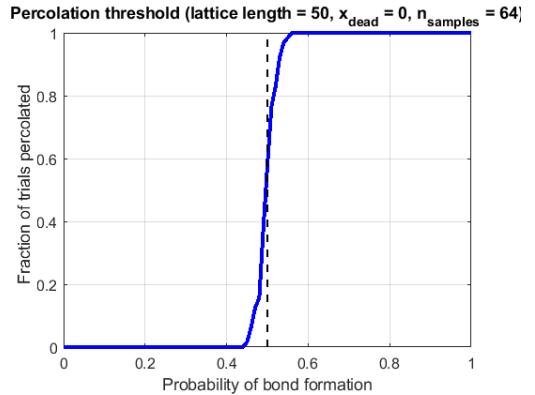


Figure 3: Simulated probability of percolation given bond probability p , with $p_c = 0.5$.

site. To our knowledge, this RO for percolation with defects does not exist in the literature.

$$\begin{aligned} p' = & (1 - P_D)^4 [2p^5 - 5p^4 + 2p^3 + 2p^2] + 2P_D(1 - P_D)^3 [p^2] \\ & + 2P_D(1 - P_D)^3 [p^4 + 3p^3(1 - p) + p^2(1 - p)^2] + 2P_D^2(1 - P_D)^2 [p^2] \end{aligned} \quad (7)$$

The fixed points to Equation 7 now depend on P_D . We solved for these points, plotted as the black line in Figure 4b. We observed that increasing P_D also increased p_c up to a point above which percolation was not expected for any p . We describe this "critical point of critical points" as the *transition point*, P_D^\dagger . Beyond P_D^\dagger , a percolated network cannot exist as an equilibrium phase due to the larger fraction of defects. We calculated this to be $P_D^\dagger = 0.11$.

We modified the simulations to account for denatured proteins by allowing formation of nodes that are unable to form bonds. The simulations revealed similar qualitative behavior as the theory, with p_c increasing with P_D up to $P_D^\dagger = 0.38$. Figures showing this behavior are shown in **Appendix A**. The difference between P_D^\dagger is likely due to approximations from the RG theory as coarse-graining necessarily averages out many details, as well as small simulation lattice length.

3 Calculations: Relating model to physical parameters

3.1 Denaturation

Before applying these models, we need estimates for the parameters in the equations. Our denaturation model needs (1) the persistence length of the proteins and (2) the energy difference between folded and denatured states.

Persistence length is the length scale over which local alignment of the polymer chain becomes uncorrelated. At length scales much larger than the persistence length, the polymer behaves as a random walk. Persistence lengths of proteins are on the order of one peptide [7], and peptides are typically 25 to 30 amino acids. Ovalbumin has 385 amino acids [8] and therefore 15 persistence length units. We use this to calculate the degeneracy of protein unfolding using a 3D random walk on a cubic lattice according to Equation (2).

The energy difference between folded and denatured ovalbumin is difficult to approximate. Therefore, we back-calculated a denaturing energy based on the observations (1) eggs do not cook inside chickens, so nearly no proteins should be denatured by 40°C (the body temperature of chickens); (2) eggs can start cooking at around 50°C, so most proteins should be denatured by then. We found that an energy of $\epsilon = 1.05 \cdot 10^{-19} \frac{\text{J}}{\text{protein}}$ (or $63 \frac{\text{kJ}}{\text{mol}}$) satisfied these constraints with our determined persistence length. Solving Equation (4) with these parameters results in Figure 4a.

3.2 Crosslinking

To estimate the probability that a bond will form, we modeled each bond site using a two-state model combined with a reversible reaction model of forming the protein-protein bond.

$$p = 1 - e^{\frac{\Delta S}{k_B} e^{\frac{-\Delta H}{k_B T}}} / \left(1 + e^{\frac{\Delta S}{k_B} e^{\frac{-\Delta H}{k_B T}}} \right) \quad (8)$$

A thorough derivation is described in **Appendix A**. Data of ΔH and ΔS for ovalbumin self-bonding is not readily available, so estimates for these thermodynamic properties were required. Liu, Peng, and Zhang (2019) calculated protein-protein binding enthalpies and entropies for 20 different systems [9]. Using values near the averages of their data, we arrived at the following:

$$\Delta H \approx 47 \frac{\text{kJ}}{\text{mol}}, \quad \Delta S \approx 36 \frac{\text{kJ}}{\text{mol K}} \quad (9)$$

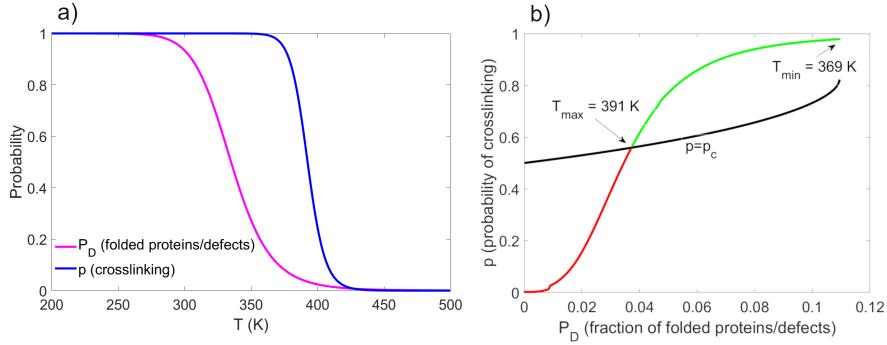


Figure 4: a) Magenta: Probability of finding a defect at a given site. Blue: Probability of bond formation at a given edge. b) Temperature range to cook an egg. Black: p_c as a function of P_D . Red/Green: p and P_D at equal temperatures. In the red section, $p < p_c$; in the green section, $p > p_c$, so the egg can cook.

Using these parameters with Equation (8), we graphed p as a function of temperature in Figure 4a.

4 Discussion: Applying full model to describe egg cooking

Our denaturing model predicts P_D , the percent of sites that are defects as a function of temperature. Our percolation analysis predicts p_c , the percolation threshold for a given temperature, as a function of P_D . For a percolation network to form (a hardened egg), the probability of bond formation given in Equation (8) must be above p_c for that temperature. If $P_D > P_D^\dagger$, an equilibrium percolated network cannot form, and P_D increases with decreasing temperature.

Combining these observations leads to a range of acceptable cooking values: if temperatures are too low, then $P_D > P_D^\dagger$ so not enough proteins will be available to form bonds. If temperatures are too high, then $p < p_c$ so the bond probability won't be high enough to lead to percolation. We solved for this range of temperatures by iteratively fixing P_D , finding the temperature that P_D corresponds to, calculating p_c as a function of P_D , and determining whether $p(T)$ is above or below $p_c(T)$. The results are shown in Figure 4b.

Figure 4b predicts a temperature range of 369 K to 391 K for cooking an egg, which qualitatively agrees with our experience cooking breakfast. The predicted minimum cooking temperature is close to the actual minimum of 345 K [10]. When cooking an egg, we never see a hardened egg white begin to melt above the maximum temperature. This is because the burning temperature of the egg white is less than 391 K, so the egg white will burn before reaching melting temperature, and thus never melt. Finally, we note that we do not see large fluctuations in egg white near the critical point when cooking an egg (analogous to critical opalescence in fluids). The reason for this is observed in Figure 4b (note that temperature increases from right to left in the diagram). For most points in the valid cooking range, the green curve, p , is far enough from the black curve, p_c , such that large fluctuations, which only occur very close to the critical point, do not occur. The only region we could expect to see large fluctuations is near where the green curve intersects p_c at 391 K. However, the egg starts to burn before p gets close enough to p_c to observe large fluctuations.

By combining a statistical mechanical model for denaturation with a percolation model for crosslinking and by applying reasonable estimates for physical parameters, we have predicted a reasonable temperature range for cooking. We observed similar qualitative results between our simulated percolation networks and renormalization operations.

References

- [1] Harold McGee. *On food and cooking: the science and lore of the kitchen*. Simon and Schuster, 2007.
- [2] Manoj Gopalakrishnan, Beate Schmittmann, and RKP Zia. “Bond percolation of polymers”. In: *Journal of Physics A: Mathematical and General* 37.29 (2004), p. L337.
- [3] Gary S Grest and Kurt Kremer. “Critical properties of crosslinked polymer melts”. In: *Journal de physique* 51.24 (1990), pp. 2829–2842.
- [4] Peter J Reynolds, HE Stanley, and W Klein. “A real-space renormalization group for site and bond percolation”. In: *Journal of Physics C: Solid State Physics* 10.8 (1977), p. L167.
- [5] GK Iliev, EJ Janse van Rensburg, and N Madras. “Phase diagram of inhomogeneous percolation with a defect plane”. In: *Journal of statistical physics* 158.2 (2015), pp. 255–299.
- [6] A Brooks Harris. “Effect of random defects on the critical behaviour of Ising models”. In: *Journal of Physics C: Solid State Physics* 7.9 (1974), p. 1671.
- [7] Guillaume Stirnemann et al. “Elasticity, structure, and relaxation of extended proteins under force”. In: *Proceedings of the National Academy of Sciences* 110.10 (2013), pp. 3847–3852.
- [8] UniProt Consortium. “UniProt: a worldwide hub of protein knowledge”. In: *Nucleic acids research* 47.D1 (2019), pp. D506–D515.
- [9] Xiao Liu, Long Peng, and John ZH Zhang. “Accurate and Efficient Calculation of Protein–Protein Binding Free Energy-Interaction Entropy with Residue Type-Specific Dielectric Constants”. In: *Journal of chemical information and modeling* 59.1 (2018), pp. 272–281.
- [10] Carl M Schroeder et al. “Overview and summary of the Food Safety and Inspection Service risk assessment for *Salmonella Enteritidis* in shell eggs, October 2005”. In: *Foodborne Pathogens & Disease* 3.4 (2006), pp. 403–412.

5 Appendix A

5.1 Equation derivations

5.1.1 Denaturation partition function

The following lines give the derivation of Equation (3):

$$Q = \sum_{\nu} g^{\sum_{i=1}^N n_i} e^{-\beta E_{\nu}} \quad (10)$$

$$= \sum_{\nu} g^{\sum_{i=1}^N n_i} e^{-\beta \sum_{i=0}^N n_i \epsilon} \quad (11)$$

$$= \sum_{\nu} \prod_{i=1}^N (ge^{-\beta\epsilon})^{n_i} \quad (12)$$

$$= \prod_{i=1}^N \sum_{n=0}^1 (ge^{-\beta\epsilon})^n \quad (13)$$

$$= \prod_{i=1}^N (ge^{-\beta\epsilon} + 1) \quad (14)$$

$$= (ge^{-\beta\epsilon} + 1)^N \quad (15)$$

The degeneracy given in Equation (2) can be derived by assuming a fixed point for the first unit (as we are finding the degeneracy of a stationary protein). The next unit has z placement options on the cubic lattice and then the remaining $M - 2$ units have $z - 1$ placement options.

5.1.2 Average Fraction of Folded Proteins

Equation (4) gives the average fraction of folded proteins at a given temperature. We can derive this using a relationship between the number of denatured proteins (N_D) to the energy E given in Equation (16).

$$\langle N_D \rangle = \frac{\langle E \rangle}{\epsilon} \quad (16)$$

The average energy is derived below:

$$\langle E \rangle = -\frac{\partial \ln Q}{\partial \beta} \quad (17)$$

$$= N\epsilon \left(\frac{ge^{\beta\epsilon}}{1 + ge^{\beta\epsilon}} \right) \quad (18)$$

The fraction of folded proteins, $\langle f_F \rangle$ can be found by subtracting the fraction of denatured proteins ($\frac{\langle N_D \rangle}{N}$) from 1. This is done below:

$$\langle f_F \rangle = 1 - \frac{\langle N_D \rangle}{N} \quad (19)$$

$$= 1 - \left(\frac{ge^{\beta\epsilon}}{1 + ge^{\beta\epsilon}} \right) \quad (20)$$

$$= 1 - \left(\frac{6 \cdot 5^{M-2} e^{-\beta\epsilon}}{1 + 6 \cdot 5^{M-2} e^{-\beta\epsilon}} \right) \quad (21)$$

where in the final line we plug in for the degeneracy.

5.1.3 Crosslinking bond formation probability

Starting from the canonical ensemble partition function and degeneracy g_b , we derived Q and subsequently p as follows:

$$Q = \sum_{\nu} g_b^{\sum_{i=1}^N n_i} e^{-\beta E_{\nu}} \quad (22)$$

$$= \sum_{\nu} g_b^{\sum_{i=1}^N n_i} e^{-\beta \sum_{i=0}^N n_i \epsilon} \quad (23)$$

$$= \sum_{\nu} \prod_{i=1}^N (g_b e^{-\beta \epsilon})^{n_i} \quad (24)$$

$$= \prod_{i=1}^N \sum_{n=0}^1 (g_b e^{-\beta \epsilon})^n \quad (25)$$

$$= (g_b e^{-\beta \epsilon} + 1)^N \quad (26)$$

$$q = 1 + g_b e^{-\beta \epsilon} \quad (27)$$

$$p_{\nu} = \frac{g_b e^{\frac{-E_v}{k_B T}}}{q} = \frac{g_b e^{\frac{-E_v}{k_B T}}}{1 + g_b e^{\frac{-E_v}{k_B T}}} \quad (28)$$

To connect this model to real-world properties, we modeled the probability of bond formation based on its thermodynamic properties driven by free energy. With this formulation, $E_v = \Delta H$. To estimate the degeneracy g_b , we noted that from the definition of free energy,

$$\Delta G = \Delta H - T \Delta S \quad (29)$$

$$e^{\frac{-\Delta G}{k_B T}} = e^{\frac{\Delta S}{k_B}} \cdot e^{\frac{-\Delta H}{k_B T}} \quad (30)$$

Since we are modeling bond formation based on free energy, this formulation indicates that the degeneracy stems from entropy. We thus formally calculate g_b via:

$$\Delta S = k_B \log \Delta \Omega \quad (31)$$

$$\therefore g_b = \Delta \Omega = e^{\frac{\Delta S}{k_B}} \quad (32)$$

Finally, we can write out our final form for p , noting that $p = 1 - p_{\nu}$, and arrive at the result in equation 8.

5.2 Majority Rule Renormalization Operation

Another method for renormalization, which is common in site percolation but has not been applied to bond percolation, is using the majority rule. This rule is simpler to find a renormalization operation rule so we can use a larger length scale. We will use a length scale of $l = 3$. In this renormalization, we will assign a horizontal/vertical bond to the CG representation if the over half of the horizontal/vertical bounds are occupied in the larger scale representation. Using this rule, we arrive at Equation (33), where the first term represents the probability of finding 9 horizontal bonds in a 3×3 region, the second term represents the probability of finding 8 horizontal bonds in a 3×3 region, etc., all the way down to finding 5 horizontal bonds.

$$p' = p^9 + 9p^8(1-p) + 36p^7(1-p)^2 + 84p^6(1-p)^3 + 126p^5(1-p)^4 \quad (33)$$

The fixed points of Equation (33) are $p_c = 0, \frac{1}{2}, 1$, which are the same as using the SCR.

There are advantages and disadvantages to using SCR versus the majority rule. An advantage to SCR is that it is more connected to the actual quantity of interest, the formation of spanning clusters. However, one disadvantage is that there are some cases where spanning clusters could form between coarse-grained regions when they did not exist before, and vice-versa. In addition, it is more complicated so we must use a small length scale to coarse-grain. The majority rule has the advantage that we can more easily coarse-grain over large length scales. However, because the majority rule does not involve finding spanning clusters, it is not too related to the actual topological property of interest.

5.3 Derivation of Spanning Cluster Rule

5.3.1 Without Defects

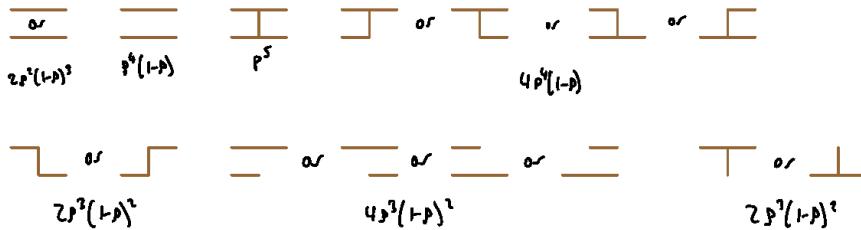
Coarse-Graining Region



Bonds that contribute to horizontal spanning cluster



Ways to form spanning cluster with probability



Summing the probabilities results in the renormalization rule:

$$g^1 = 2p^5 - 5p^4 + 2p^3 + 2p^2$$

5.3.2 With Defects

Coarse-graining region



Case with zero defects:

This is equivalent to the normal spanning cluster rule, multiplied by the probability of not having a defect

$$(1-p_0)^4 [2p^5 - 5p^4 + 2p^3 + 2p^2]$$

Case with one defect: $(1-p_0)^3 p_0$

Bottom Left:

p^4	$p^3(1-p)$	$p^2(1-p)$	$p^2(1-p)^2$	$p^2(1-p)^3$

Bottom Right:

$$\overline{p^2}$$

Top Left:

p^4	$p^3(1-p)$	$p^2(1-p)$	$p^2(1-p)^2$	$p^2(1-p)^3$

Top Right:

$$\overline{p^2}$$

Case with 2 defects: $(1-p_0)^2 p_0^2$

Bottom 2:

$$\overline{\overline{p^2}}$$

Top 2:

$$\overline{\overline{p^2}}$$

Summing the probabilities results in the renormalization rule:

$$\begin{aligned} p' &= (1-p_0)^4 [2p^5 - 5p^4 + 2p^3 + 2p^2] + 2p_0(1-p_0)^3 [p^2] \\ &\quad + 2p_0(1-p_0)^3 [p^4 + 3p^3(1-p) + p^2(1-p)^2] \\ &\quad + 2p_0^2(1-p_0)^2 [p^2] \end{aligned}$$

5.4 Order parameter scaling

5.4.1 Theory

First, we will derive general relationships for the scaling. Near the critical point, there is a universal scaling exponent ν that is shown in Equation (34).

$$\xi = |p - p_c|^{-\nu} \quad (34)$$

We can use the formula for ν derived in the literature [4], given in Equation (35).

$$\nu = \frac{\ln l}{\ln \frac{dR_l(p)}{dp}|_{p=p_c}} \quad (35)$$

By taking the derivative of Equation (33), we find that the ν predicted by the majority renormalization operation is 1.22. By taking the derivative of Equation (6), we find that the ν predicted by the SCR is 1.43. The analytical value of ν for 2D bond percolation on a square lattice is $\nu = \frac{4}{3}$. Both predictions are around 0.1, or 7.5% off.

We can now find the scaling exponent for the SCR with defects. By taking the derivative of Equation (7), we get Equation (36).

$$\begin{aligned} \frac{dR_l(p)}{dp} &= (1 - P_d)^4[10p^4 - 20p^3 + 6p^2 + 4p] \\ &\quad + 2P_D(1 - P_D)^3[2p] + 2P_D^2(1 - P_D)^2[2p] \\ &\quad + 2P_D(1 - P_D)^3[4p^3 + 3(3p^2 - 4p^3) + 2p(1 - p)^2 - 2p^2(1 - p)] \end{aligned} \quad (36)$$

We can use Equation (36) to plot ν as a function of P_D . This plot is shown in Figure 5.

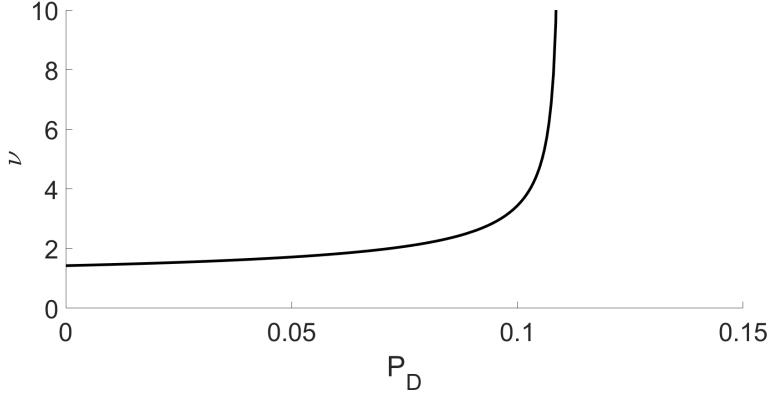


Figure 5: Scaling exponent predicted using SCR with defects.

Figure 5 shows that ν diverges as P_D approaches P_D^\dagger .

5.4.2 Simulation

The computational algorithm we developed can also estimate scaling behavior of the *strength*, defined as the chance of any randomly selected node appearing in a percolating cluster. The strength is expected to diverge near the critical point, which is what we roughly observe in Figure 6. If desired, one could sample points near the critical point and estimate the power law exponent describing scaling behavior. However, to achieve acceptable error, this requires a very large N_{sample} and large N which would need very long simulation times. Hence, we did not calculate a scaling exponent, though the capacity to do so was written into our script.

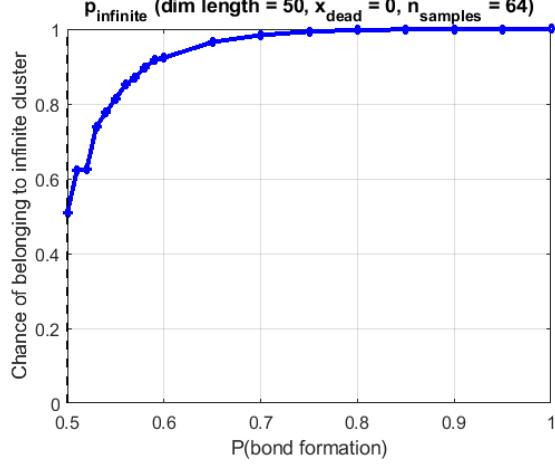


Figure 6: Chance of randomly selecting a node belonging to a percolating cluster. This order parameter roughly diverges near the critical point, and can be used to estimate scaling.

5.5 Effect of defects on simulated p_c

Figure 7 shows how the simulated p_c increases as P_D increases.

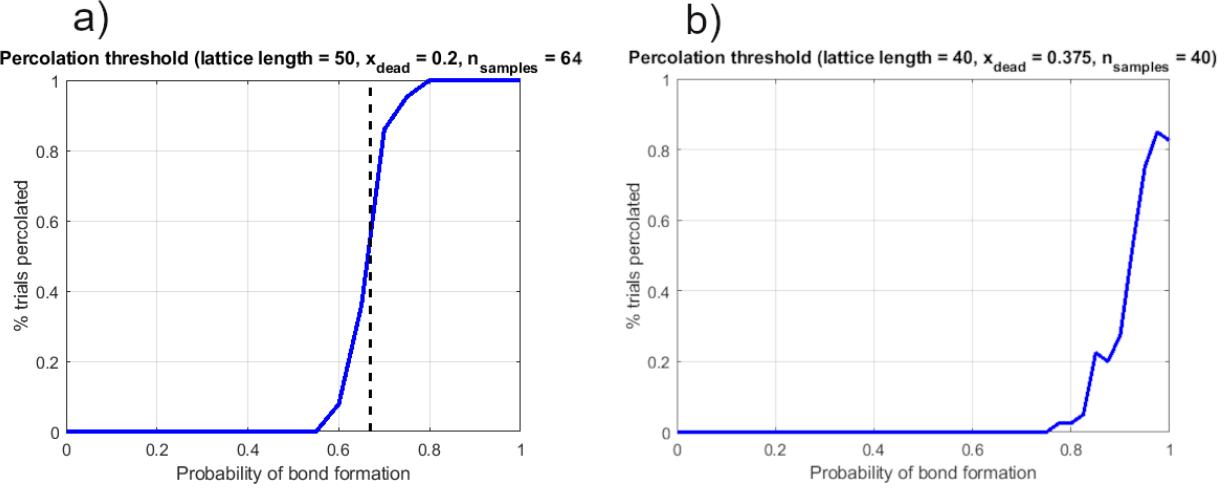


Figure 7: Example of how percolation probability vs. p changes as P_D increases. We observe in (a) that the p_c increases with increasing P_D ; (b) upon reaching P_D^\dagger , there is no bonding probability for which percolation is expected.

5.6 Percolation simulation results

The lattices produced by the percolation simulations were shown in the main body of this paper. These lattices can also be reconstructed as undirected graphs, which enables use of efficient algorithms for determining cluster sizes and identifying percolation pathways. Figure 8 depicts these isomorphic representations for both a non-percolated and percolated system.

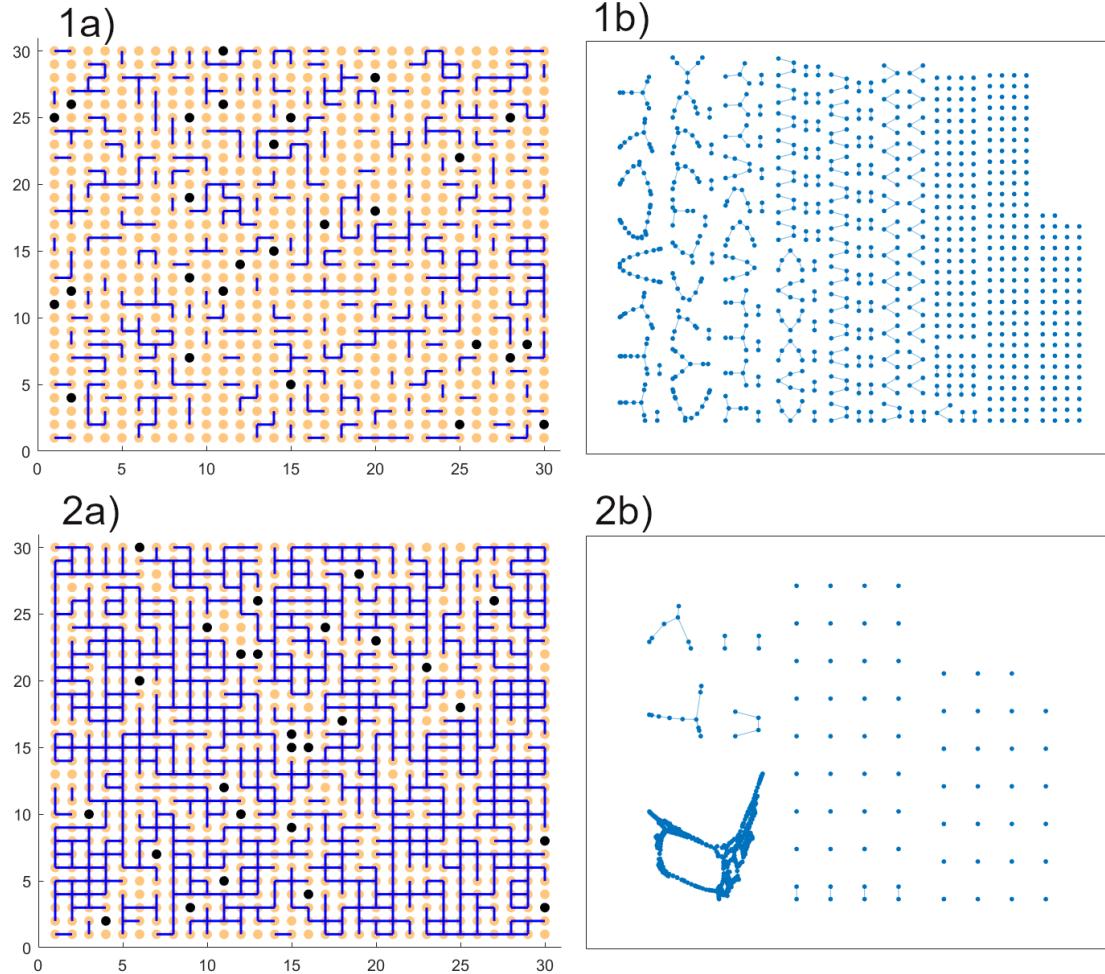


Figure 8: (1a) Lattice view and (1b) graph representation for $p = 0.27$; (2a) Lattice view and (2b) graph representation for $p = 0.60$. In both simulations, $x_{\text{dead}} = 0.03$ and $N = 30$. Orange lattice sites represent denatured proteins / "open" sites, black sites represent dead proteins / defects, and blue lines represent bonds. In the graph, each dot represents a node corresponding to a unique lattice position, and each edge is isomorphic to a lattice bond.

6 Appendix B

6.1 Percolation simulation and analysis script details

The MATLAB script used to simulate percolation is attached to this report as a supplementary file. The user can control the size of the lattice, the number of samples to take (e.g. the number of lattices to randomly generate and average over when measuring bulk quantities), and the site fraction of defects. The general algorithm & features are as follows:

1. User defines input parameters, flags, and sets up experiments to utilize simulation runs (e.g. sweep over energy/temperature range, visualize a lattice, etc).
2. In each simulation run, a lattice is created randomly by creating an $N \times N$ matrix that represents the atomic sites. The algorithm then randomly, exhaustively places dead proteins as specified by the input (if `xdead` > 0).
3. Next, an $N^2 \times N^2$ adjacency matrix `bonds` is created representing the system's bonds. Each row of `bonds` corresponds to an atom being linked *from*, and the column represents the atom being linked *to*, using 1-D indexing (e.g. top-left element of the lattice is indexed 1, the element to its right is 2, the element below it is $2 + N$, etc).
4. Each row of this adjacency matrix is traversed once. At each row, the program calculates the indices of neighboring, non-defect sites. To avoid double counting, only sites to the right and below are considered. If a bond is allowed, the system generates a random number between 0 and 1. If that number is below the acceptance threshold, then the bond is accepted.
5. After traversing all rows, initialization is done. This process is repeated $N_{samples}$ times.
6. The bonds adjacency matrix is then post-processed so that it is an upper diagonal matrix. This matrix is then converted to a graph for the percolation, strength, and cluster size analyses.
7. Percolation is detected for each lattice by checking if any element on the leftmost edge has a path to the rightmost edge or any element on the topmost edge has a path to the bottommost one. If so, then the system is determined as percolated.
8. Cluster sizes are determined by exhaustively performing depth-first search on all nodes and storing info on connected nodes.
9. Strengths are calculated with a slightly more complex algorithm. Each point on an edge is checked for a path against an opposite edge. The unique infinite paths are detected via depth-first search. To improve performance, the algorithm ensures no search is performed on nodes that already belong to an infinite network.
10. A lattice is visualized by plotting dead and alive proteins using the $N \times N$ `sites` matrix, then finding the index of nonzero elements in all rows of the `bonds` matrix and drawing a line between the row position and indexed position.
11. If one set up an experiment to sample near the critical point, one can automatically attempt to calculate the scaling exponent with any of the above order parameters.

Some notes for potential future users:

1. Sparse algorithms for `bonds` and `sites` may greatly improve performance.
2. PBC are not recommended as they make the graph algorithms difficult to write.