Protein Folding

2D MJ Lattice model on D-Wave Quantum Annealing Hardware

Argenis Arriojas
UMass Boston - Computational Sciences PhD program
CS612 - Algorithms in Bioinformatics. Spring 2019
June 5th, 2019

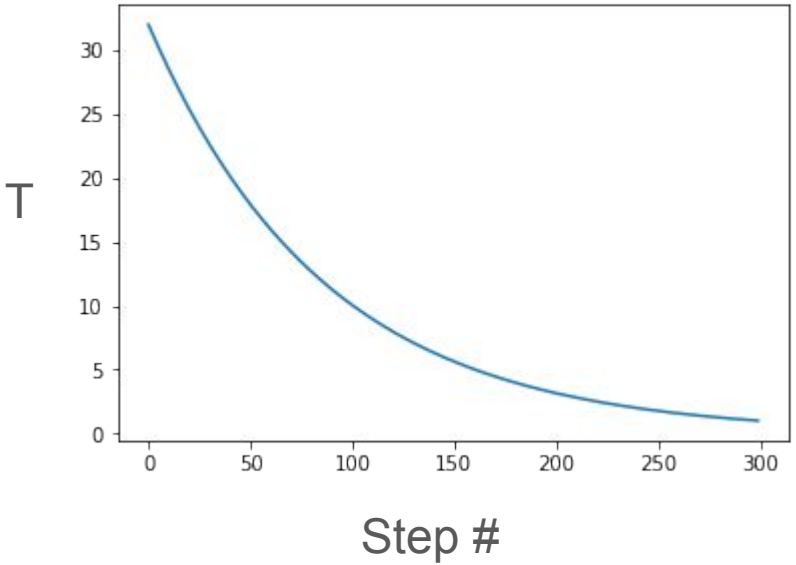
Use Conventional Monte Carlo Simulated Annealing

Methods *Simulated Annealing (Conventional MC)*

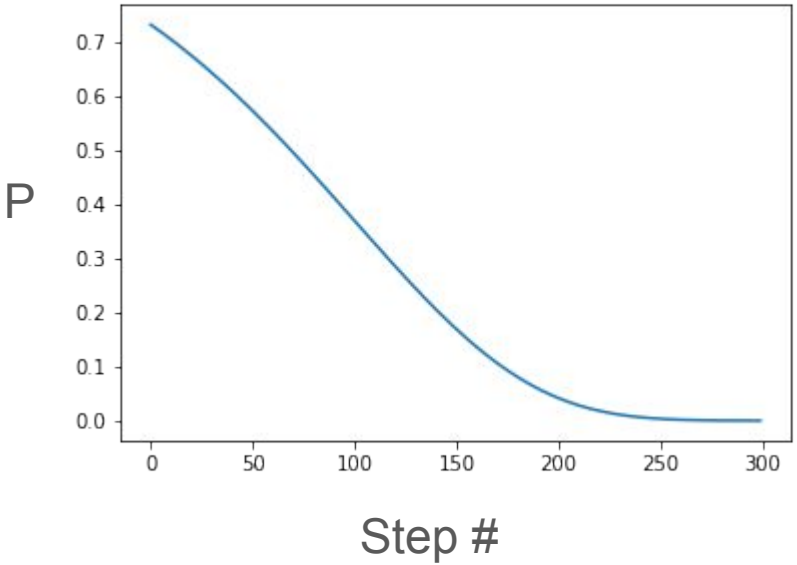
- Define a schedule for annealing temperature T
- Randomly choose i-th residue r_i
- Perform random walk, with respect to $r_{(i - 1)}$
  - Keep residue path directions for r_j with $j > i$
- Compute the change in energy $dE = E - E'$
  - E' and E are the energies before and after the step is taken
- Accept step if $\exp(-dE/T) >$ random.uniform(0, 1)
  - Note that $dE \leq 0 \rightarrow$ accept always

Methods *Simulated Annealing (Conventional MC)*

Annealing Schedule



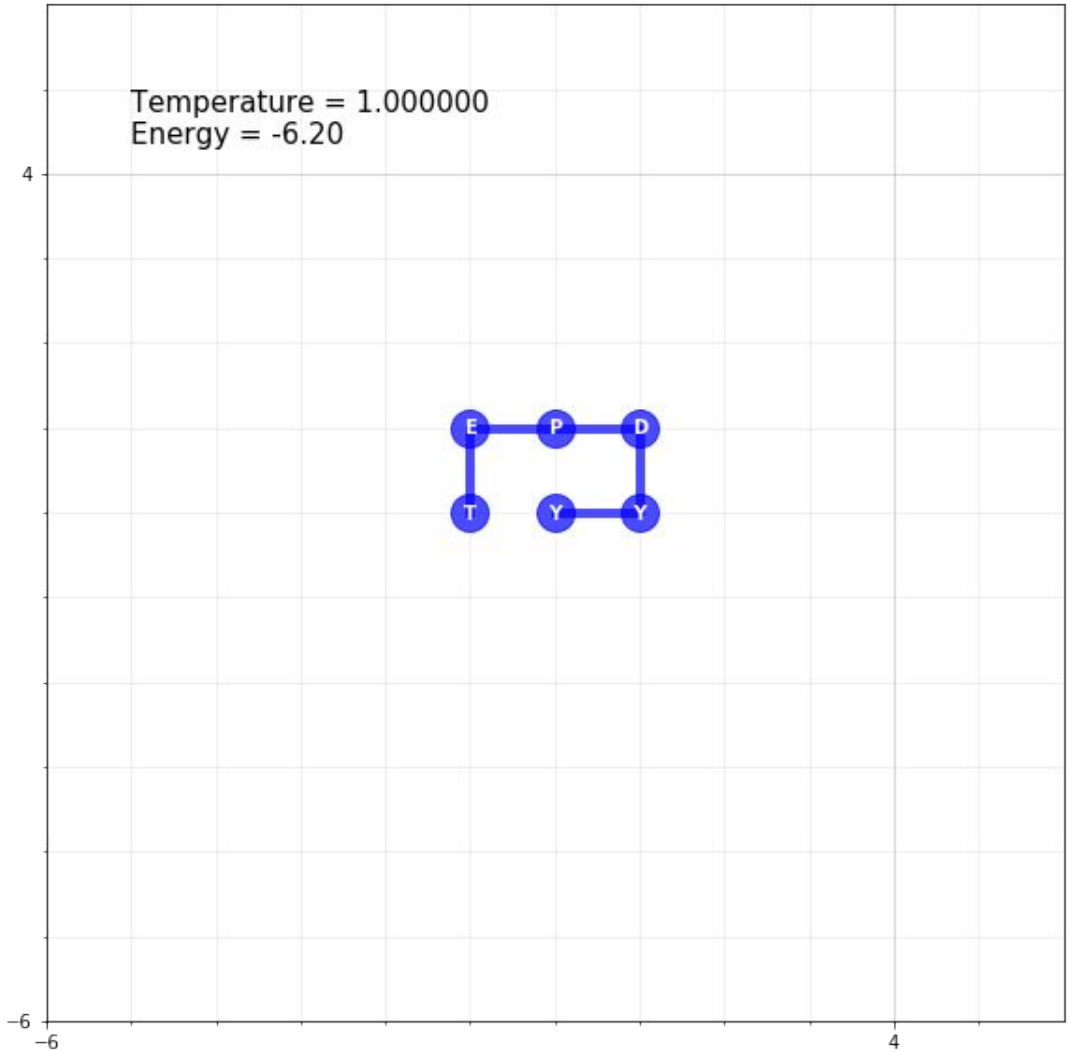
Probability of accepting dE = +10



Results

Simulated Annealing (Conventional MC)

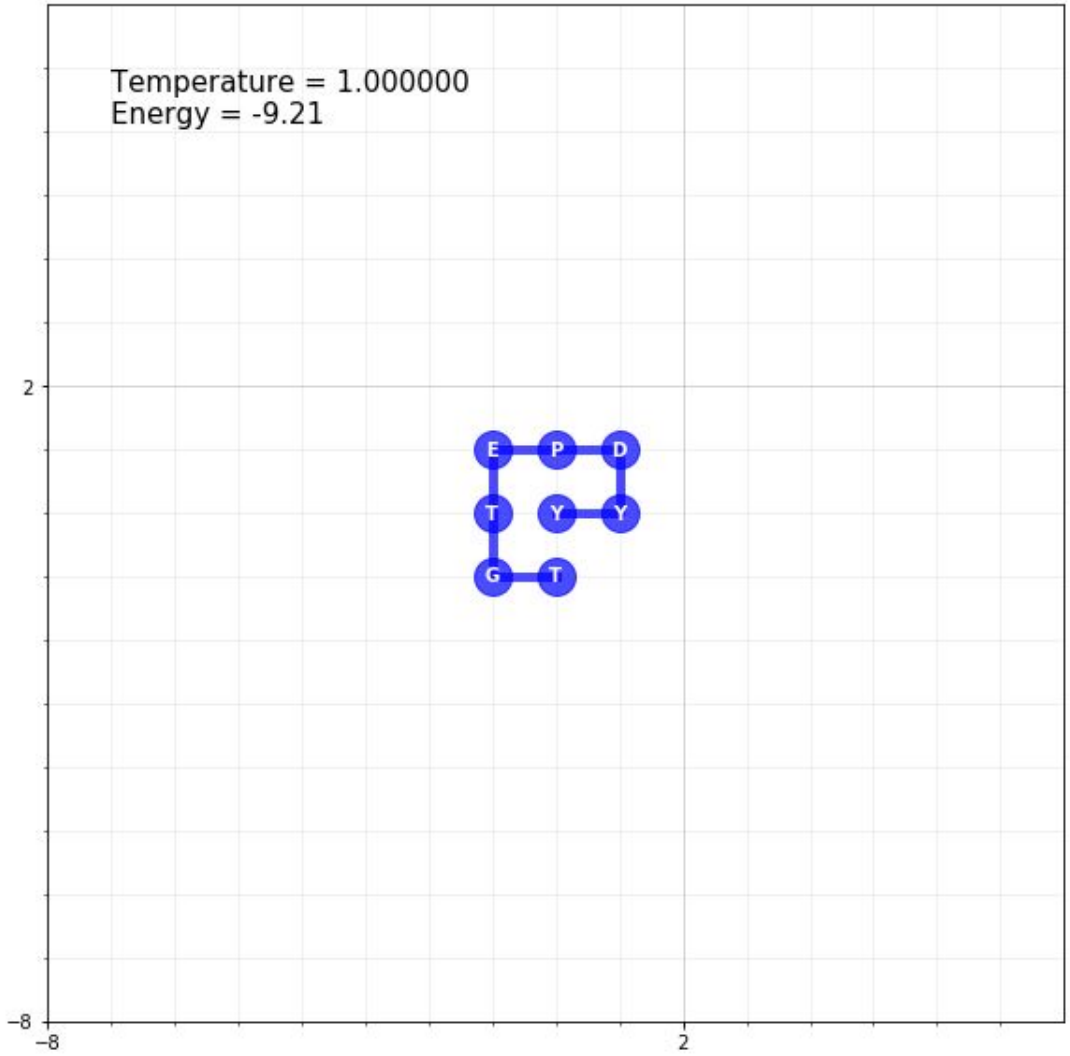
67 out of 360 (18.6%) in lowest energy conformation



Results

Simulated Annealing (Conventional MC)

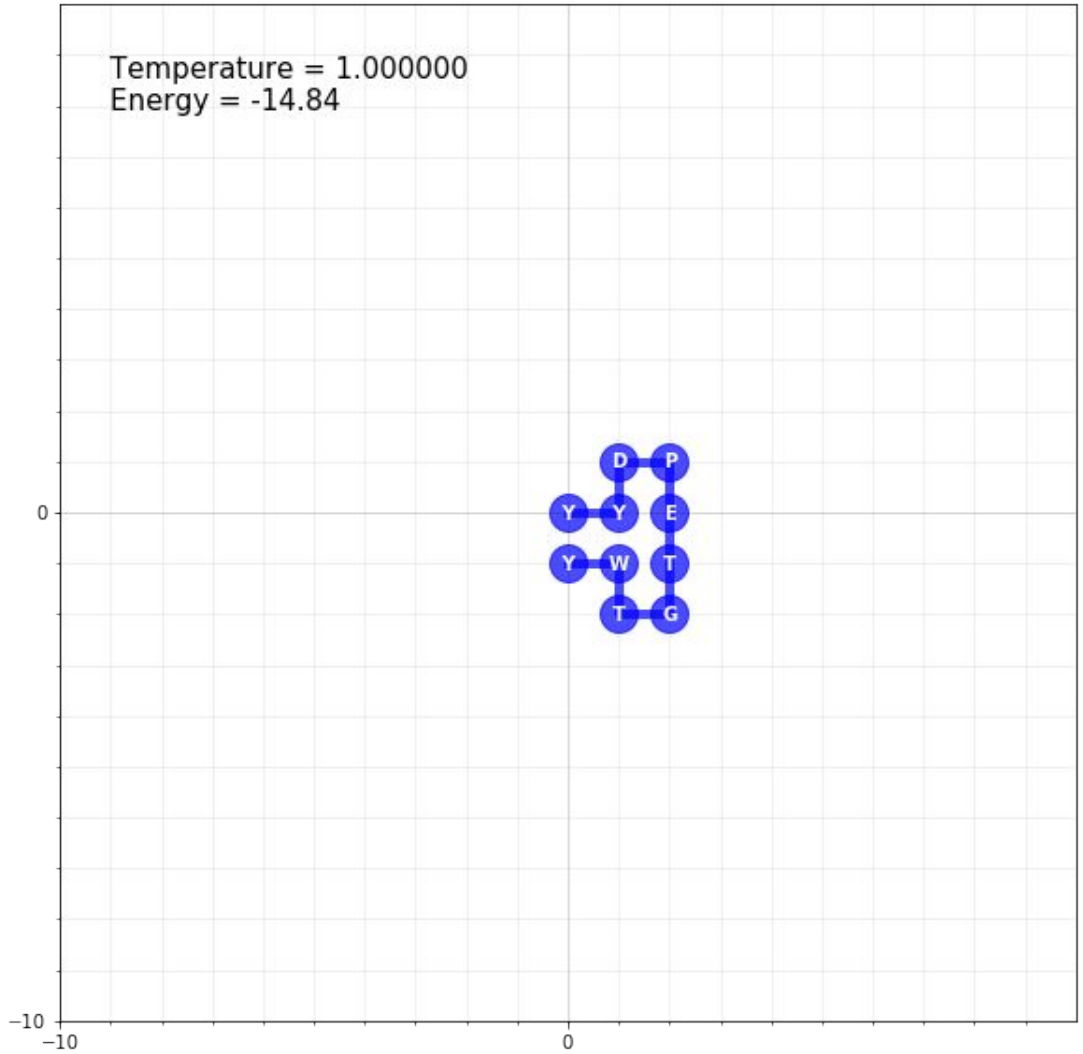
22 out of 360 (6.1%) in lowest energy conformation



Results

Simulated Annealing (Conventional MC)

5 out of 360 (1.4%) in lowest energy conformation



Simulated annealing with qbit model: Turn Ancilla Encoding

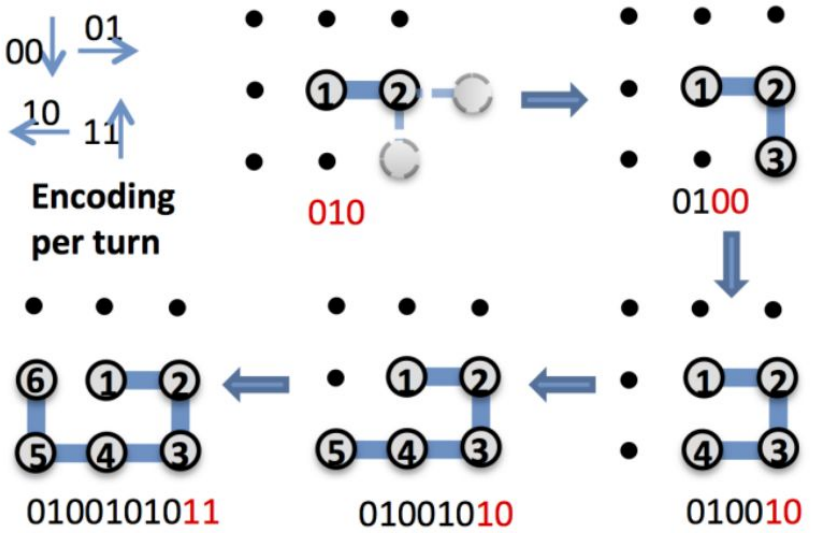
Babbush, R., Perdomo-Ortiz, A., O'Gorman, B., Macready, W., and Aspuru-Guzik, A. (2013). Construction of energy functions for lattice heteropolymer models: a case study in constraint satisfaction programming and adiabatic quantum optimization. arXiv preprint arXiv:1211.3422v2

Methods *Turn Ancilla Encoding*

Babbush, R. et al. (2013). arXiv preprint arXiv:1211.3422v2

The model

$$\boldsymbol{q} = 01\ \underbrace{0q_1}_{turn2}\ \underbrace{q_2q_3}_{turn3}\cdots\underbrace{q_{2(N-1)-4}q_{2(N-1)-3}}_{turn(N-1)}$$



Methods *Turn Ancilla Encoding*

Babbush, R. et al. (2013). arXiv preprint arXiv:1211.3422v2

The model

$$E(\boldsymbol{q}) = E_{back}(\boldsymbol{q}) + E_{overlap}(\boldsymbol{q}) + E_{pair}(\boldsymbol{q})$$

- Construct energy functions
  - Penalize bad conformations
  - Favor (or penalize) contact between residues
    - Use Miyazawa-Jernigan interaction potentials
- Optimization problem: minimize objective function E(q)

Methods *Turn Ancilla Encoding*

Babbush, R. et al. (2013). arXiv preprint arXiv:1211.3422v2

The model

$$\begin{aligned} E_{back}(\boldsymbol{q}) &= \lambda_{overlap}\left(q_1q_2 + q_2q_3 - 2q_1q_2q_3\right) \\ &+ \lambda_{overlap}\sum_{i=2}^{2N-8} E_{back}\left(q_iq_{i+1}q_{i+2}q_{i+3}\right)\left[(i+1)\mathrm{mod}2\right] \end{aligned}$$

$$E_{back}\left(q_iq_{i+1}q_{i+2}q_{i+3}\right) = \lambda_{overlap}\left(2q_iq_{i+2} - q_i - q_{i+2}\right)\left(2q_{i+1}q_{i+3} - q_{i+1} - q_{i+3}\right)$$

Methods *Turn Ancilla Encoding*

Babbush, R. et al. (2013). arXiv preprint arXiv:1211.3422v2

The model

$$E_{overlap}(\boldsymbol{q}) = \sum_{i=1}^{N-4} \sum_{j=i+4}^{N} \left[(1+i-j) \bmod 2\right] \gamma_{ij} \qquad \text{with} \qquad \gamma_{ij} = \lambda_{overlap} \left[2^{\mu_{ij}} - g_{ij} - \alpha_{ij}\right]^2$$

$$\forall\, g_{ij} \geq 1 \,\exists\, \alpha_{ij} : 2^{\mu_{ij}} - g_{ij} - \alpha_{ij} = 0$$

hence $2^{\mu_{ij}} - g_{ij} - \alpha_{ij} \geq 1 \,\forall\, \alpha_{ij}$ if and only if $g_{ij} = 0$

- g_{ij} ➔ Manhattan distance between i-th and j-th residues
- α_{ij} ➔ Some number in decimal representation, with $0 \leq \alpha_{ij} \leq 2^{\mu_{ij}} - 1$
- μ_{ij} ➔ Number of qbits used for α_{ij} `s binary representation

Methods *Turn Ancilla Encoding*

Babbush, R. et al. (2013). arXiv preprint arXiv:1211.3422v2

The model

$$E_{pair}(\boldsymbol{q}) = \sum_{i=1}^{N-3} \sum_{j=i+3}^{N} \omega_{ij} J_{ij} \left(2 - g_{ij}\right)$$

$$\varphi_{ij} = \omega_{ij} J_{ij} \left(2 - g_{ij}\right)$$

$$\varphi_{ij} = J_{ij} \text{ if and only if } g_{ij} = 1 \wedge \omega_{ij} = 1$$

J_{ij} ➔ Miyazawa-Jernigan interaction potential between i-th and j-th residues

- Note that $J_{ij} < 0$ for all possible pairwise interactions, hence if gij > 2 then φ_{ij} becomes positive, penalization that is easily removed by setting $\omega_{ij} = 0$
- Only pairs with odd relative positions along the backbone can be in contact, we set $J_{ij} = 0$ for the rest

ω_{ij} ➔ Ancillary bit: a switch which is only “on” without incurring an energy penalty if two amino acids are interacting ($g_{ij} = 1$)

Methods *Simulated Annealing (Ancilla Encoding MC)*

- Define a schedule for annealing temperature T
- Randomly choose i-th “qbit” q_i
- Perform qbit flip
- Compute the change in energy dE = E - E’
  - E’ and E are the energies before and after the step is taken
- Accept flip if exp(-dE/T) > random.uniform(0, 1)
  - dE ≤ 0 → accept always

Methods *Simulated Annealing (Ancilla Encoding MC)*

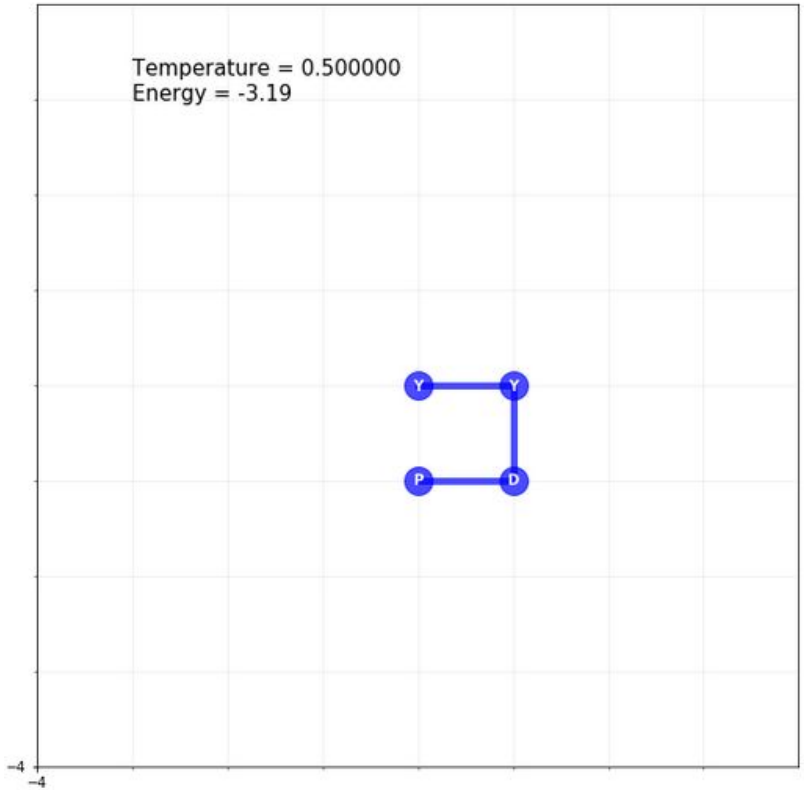
The energy (4 residue protein)

$$
\begin{aligned}
E = {} & 20.0q_0q_1q_2q_3 - 10.0q_0q_1q_2 - 10.0q_0q_1q_3 + 5.0q_0q_1 - 10.0q_0q_2q_3 \\
& + 12.76q_0q_2q_6 + 5.0q_0q_3 + 12.76q_0q_4q_6 - 12.76q_0q_6 - 10.0q_1q_2q_3 \\
& + 5.0q_1q_2 + 12.76q_1q_3q_6 + 12.76q_1q_5q_6 - 12.76q_1q_6 + 20.0q_2q_3q_4q_5 \\
& - 10.0q_2q_3q_4 - 10.0q_2q_3q_5 + 10.0q_2q_3 - 10.0q_2q_4q_5 + 12.76q_2q_4q_6 \\
& + 5.0q_2q_5 - 12.76q_2q_6 - 10.0q_3q_4q_5 + 5.0q_3q_4 + 12.76q_3q_5q_6 \\
& - 12.76q_3q_6 + 5.0q_4q_5 - 12.76q_4q_6 - 12.76q_5q_6 + 22.33q_6
\end{aligned}
$$

Results *Simulated Annealing (Ancilla Encoding MC)*

4 residue protein

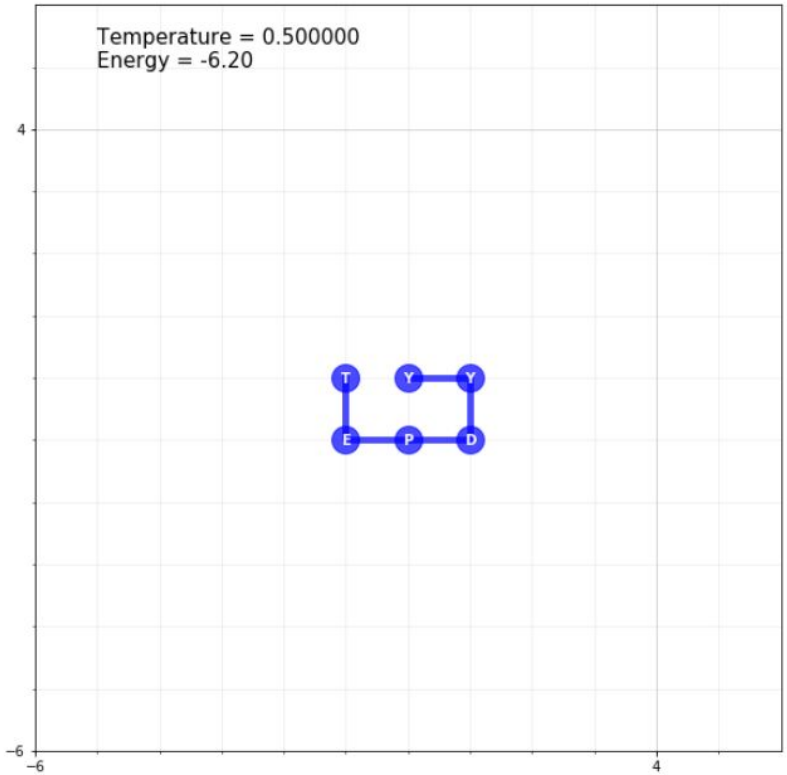
625 out of 720 (86.8%)
in lowest energy
conformation



Results *Simulated Annealing (Ancilla Encoding MC)*

6 residue protein

3 out of 720 (0.4%)
in lowest energy
conformation

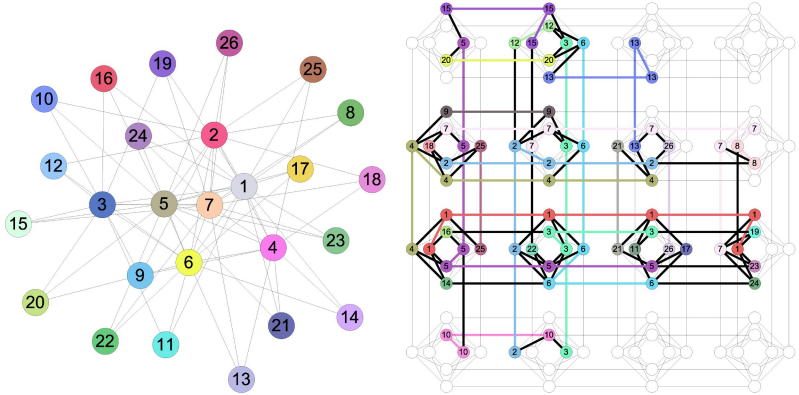


Adapting the problem

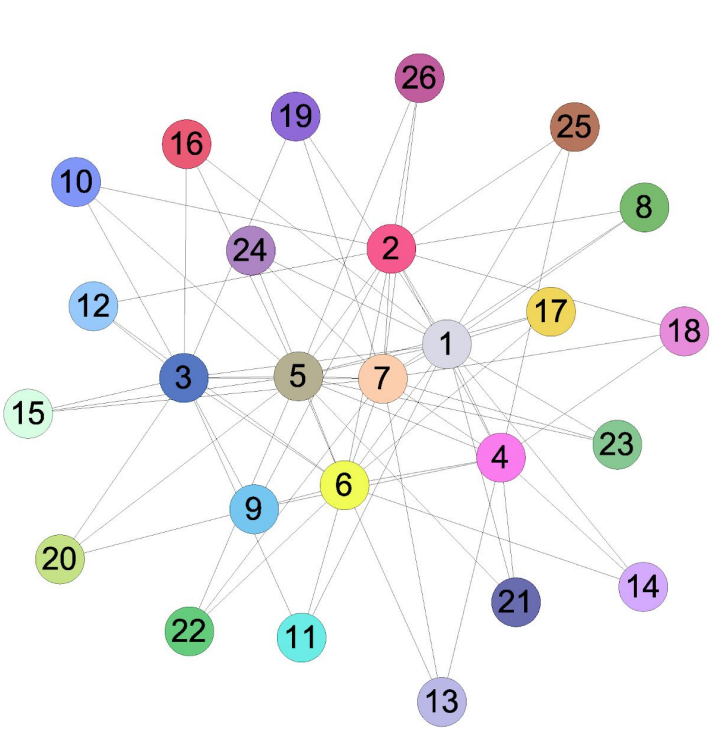
So it fits in D-Wave's Hardware
with Chimera Architecture

Methods *Order reduction and D-Wave Minor-Embedding*

- Our energy expression has terms with up to degree 4
  - We need to introduce ancillary qbits to build equivalent expression of order 2
- A qbit in our model may need be coupled with many other qbits
  - Qbit connectivity is limited
  - Qbit chains need to be introduced to build equivalent interaction graph



Methods *D-Wave Minor-Embedding*



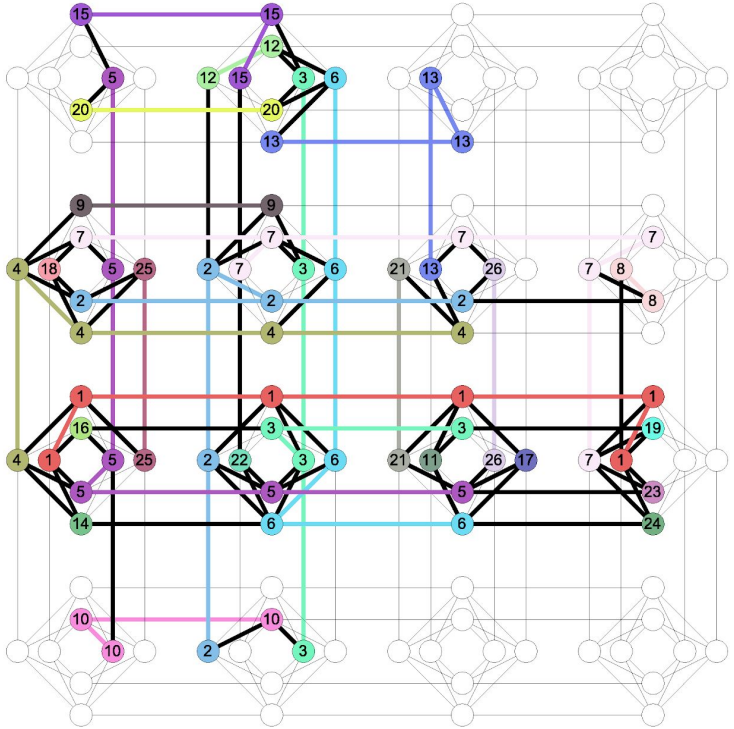


Image is not related to our problem at hand. Shown here just for illustration of embedding

Methods *Quantum Annealing (Ancilla Encoding on D-Wave QPU)*

```
# Now make sure an embedding can be found
problem = nx.Graph()
for k, v in dict(bqm.linear).items():
    problem.add_node(k)
for k, v in dict(bqm.quadratic).items():
    problem.add_edge(*k)
```

```
num_cores = 12
embedding_list = Parallel(n_jobs=num_cores)(delayed(find_embedding)(problem, solver_G) for jobid in range(10*num_cores))
```

```
assigned_qbits = np.array([len(sum(emb.values(), [])) if len(emb) > 0 else np.nan for emb in embedding_list])
assert bool(sum(assigned_qbits[~np.isnan(assigned_qbits)])), "No embedding has been found"
print(np.sort(assigned_qbits[~np.isnan(assigned_qbits)]))

embedding = embedding_list[np.nanargmin(assigned_qbits)]

all_assigned_qbits = sum(embedding.values(), [])
print(f"Using {len(all_assigned_qbits)}/{solver.properties['num_qubits']} qbits")
```

```
[1330. 1342. 1354. 1355. 1356. 1362. 1364. 1365. 1367. 1375. 1376. 1380.
 1381. 1387. 1388. 1391. 1394. 1399. 1400. 1402. 1403. 1405. 1405. 1405.
 1407. 1409. 1412. 1413. 1416. 1417. 1418. 1418. 1419. 1420. 1422. 1423.
 1424. 1425. 1425. 1426. 1426. 1426. 1426. 1426. 1427. 1432. 1433. 1434.
 1435. 1437. 1437. 1438. 1439. 1444. 1446. 1450. 1454. 1455. 1459. 1459.
 1460. 1461. 1463. 1464. 1464. 1466. 1472. 1476. 1479. 1481. 1494. 1496.
 1499. 1500. 1500. 1509. 1518. 1521. 1523. 1550. 1571.]
Using 1330/2048 qbits
```

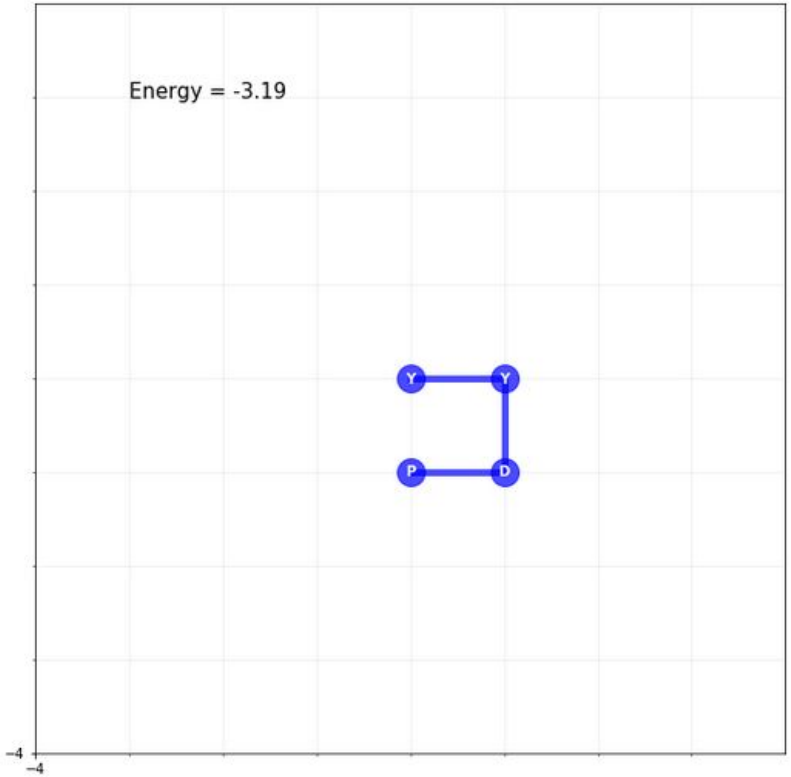
Methods *Quantum Annealing (Ancilla Encoding on D-Wave QPU)*

```
solver = DWaveSampler(solver="DW_2000Q_5")
solver_G = nx.Graph(solver.edgelist)


sampler = FixedEmbeddingComposite(solver, embedding=embedding)


response = sampler.sample(bqm_spin, num_reads=num_reads,
                          annealing_time=20,
                          num_spin_reversal_transforms=2,
                          postprocess='optimization',
                          answer_mode='raw')
```

Results *Quantum Annealing (Ancilla Encoding on D-Wave QPU)*

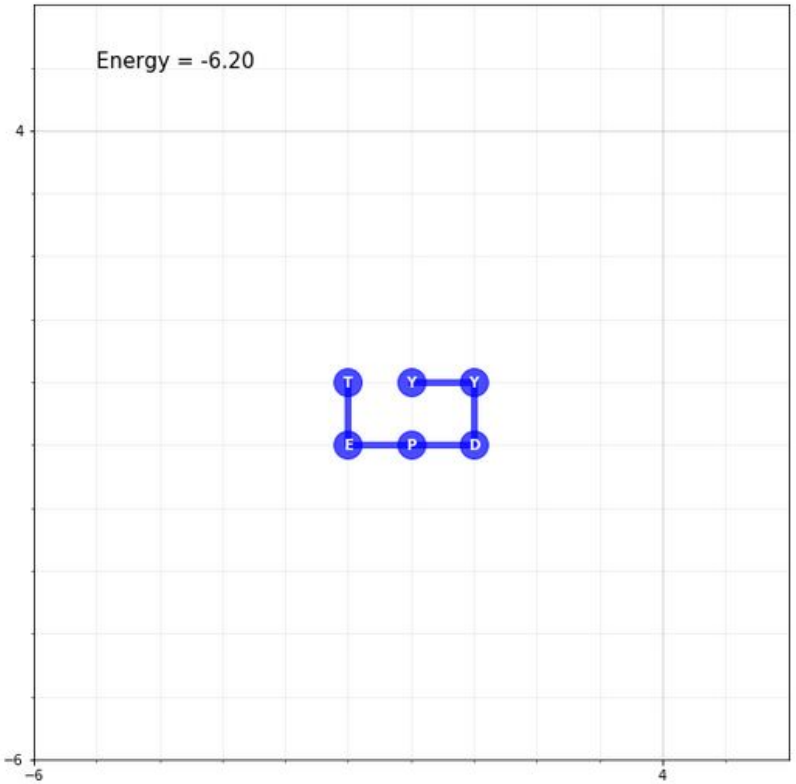


	energy	count
0	-0.075758	1700

- 170 instances in single embedding
  - 1190 qbits used
- 10 reads
- QPU time used: 24 ms

100% in lowest energy conformation

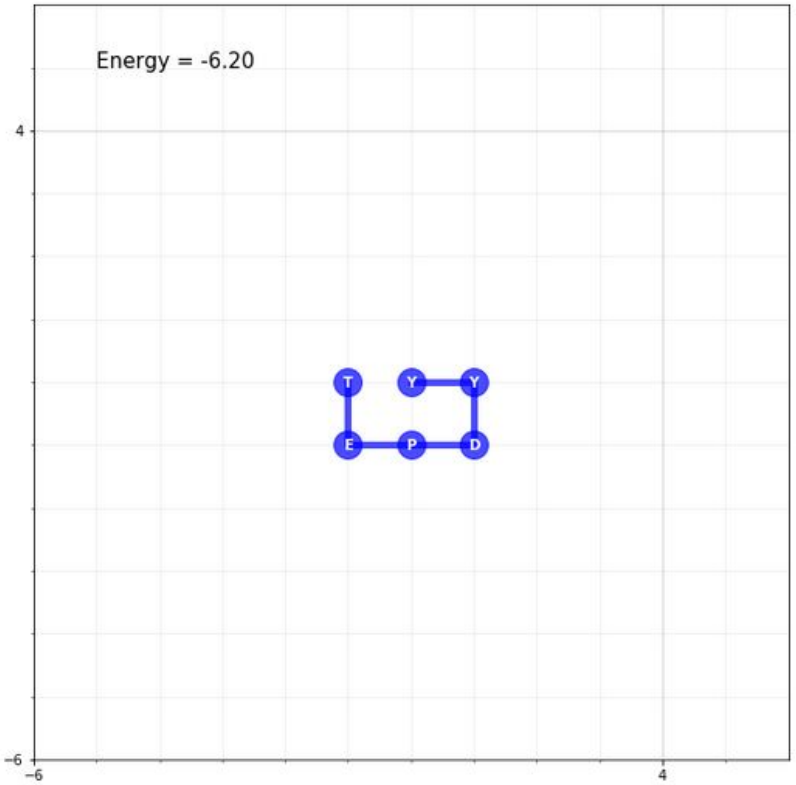
Results *Quantum Annealing (Ancilla Encoding on D-Wave QPU)*



- 7 instances in single embedding
  - 1330 qbits used
- 10000 reads
- QPU time used: 3170 ms

6 out of 70000 (0.00009%) in lowest energy conformation

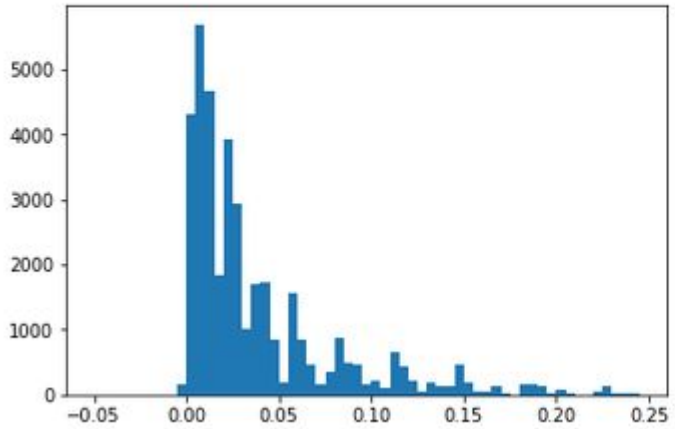
Results *Quantum Annealing (Ancilla Encoding on D-Wave QPU)*



	state	isvalid	energy	Hb	Ho	Hi	count
0	0101011011100111100	True	-1.391382e-03	0	0	-6.20	6
1	1111010011101110110	True	-1.301616e-03	0	0	-5.80	3
2	1001010011101110110	True	-1.301616e-03	0	0	-5.80	6
3	0100001001100111001	True	-1.119838e-03	0	0	-4.99	15
4	0001011001101110101	True	-1.079443e-03	0	0	-4.81	10
5	0100000001101101000	True	-7.158887e-04	0	0	-3.19	8
6	0100001001100111000	True	-7.158887e-04	0	0	-3.19	10
7	0101010011101101000	True	-7.158887e-04	0	0	-3.19	1
8	0101011011100111000	True	-7.158887e-04	0	0	-3.19	6
9	0100010001100011000	True	-7.158887e-04	0	0	-3.19	8
10	0101000011100011000	True	-7.158887e-04	0	0	-3.19	9
11	1111010011101110100	True	-6.754937e-04	0	0	-3.01	2
12	0001011001101110100	True	-6.754937e-04	0	0	-3.01	9

Results *Quantum Annealing (Ancilla Encoding on D-Wave QPU)*

```
plt.hist(all_sbsener[all_sbsvald], [-0.05 + 0.005*i for i in range(60)])
plt.show()
```



Code available

https://github.com/argearriojas/2dlattice-protfold

References

- Miyazawa Jernigan (1985) - Estimation of Effective Interresidue Contact Energies from Protein Crystal Structures: Quasi-Chemical Approximation. https://doi.org/10.1021/ma00145a039
- Miyazawa Jernigan (1996) - Residue-Residue Potentials with a Favorable Contact Pair Term and an Unfavorable High Packing Density Term, for Simulation and Threading. https://doi.org/10.1006/jmbi.1996.0114
- Perdomo-Ortiz (2012) Finding low-energy conformations of lattice protein models by quantum annealing. https://doi.org/10.1038/srep00571
- Babbush (2013) - Construction of Energy Functions for Lattice Heteropolymer Models: A Case Study in Constraint Satisfaction Programming and Adiabatic Quantum Optimization. https://arxiv.org/abs/1211.3422, https://doi.org/10.1002/9781118755815.ch05
- Babej (2018) - Coarse-grained lattice protein folding on a quantum annealer. https://arxiv.org/abs/1811.00713