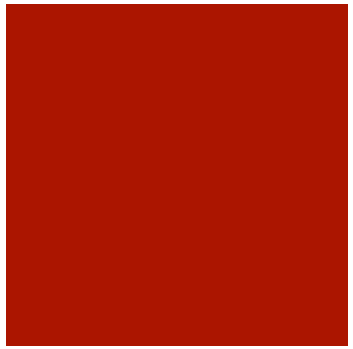
Algorithm discovery by protein folding game players

Firas Khatiba, Seth Cooper, Michael D. Tykaa, Kefan Xub, Ilya Makedonb, Zoran Popovićb, David Baker, and Foldit Players

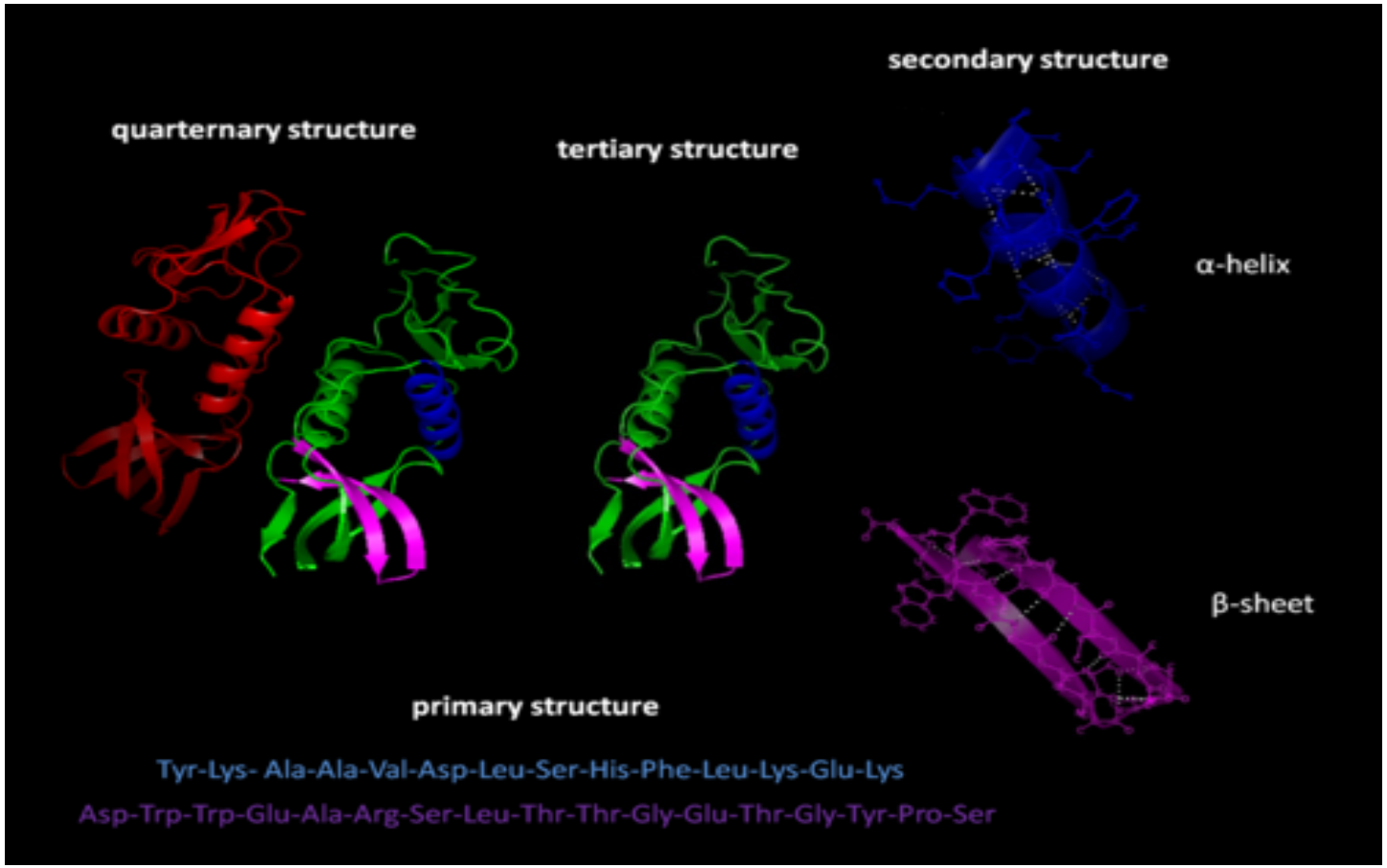
Presented by: J. White Bear

Key Ideas



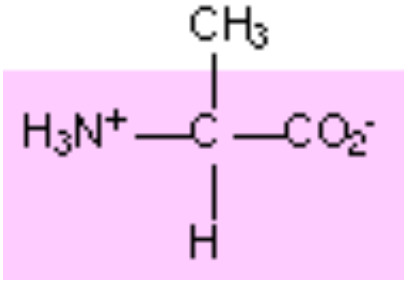
- To determine whether high performing players strategies could be collectively codified to produce effective algorithms for tackling *de novo* protein prediction using a GWAP: FoldIt and crowd-sourcing for human spatial reasoning and strategy.
- Crowd-sourcing: The existing FoldIt game was augmented to allow players to encode their strategies as recipes that could be shared with other players who are able to further modify and redistribute them
- Resulting strategies are then compared to an independently generated algorithm in Rosetta: Fast Relax.

The Protein Structure Prediction Problem

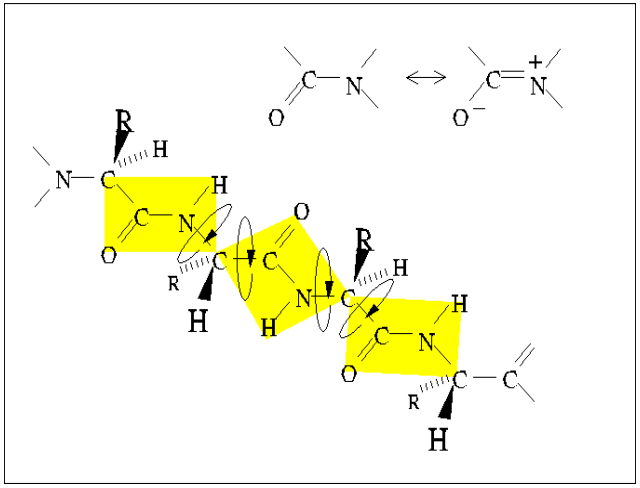


- Sequence (Primary Structure) does not infer abosolutely Secondary, Tertiary, or Quaternary structures.

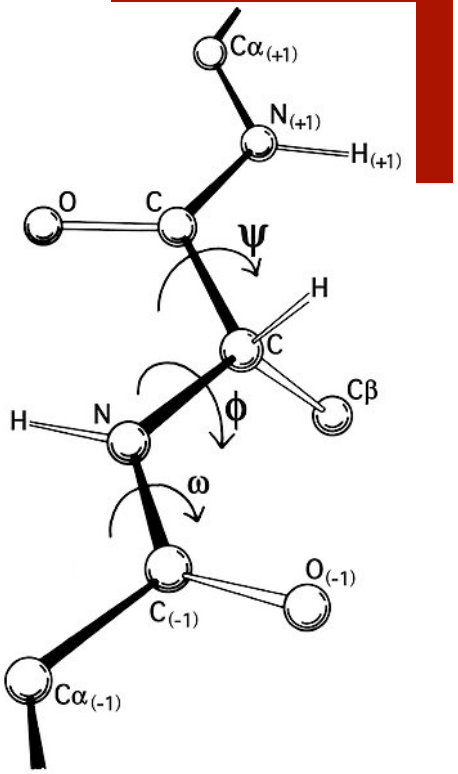
The Protein Structure Prediction Problem



Amino Acid Alanine



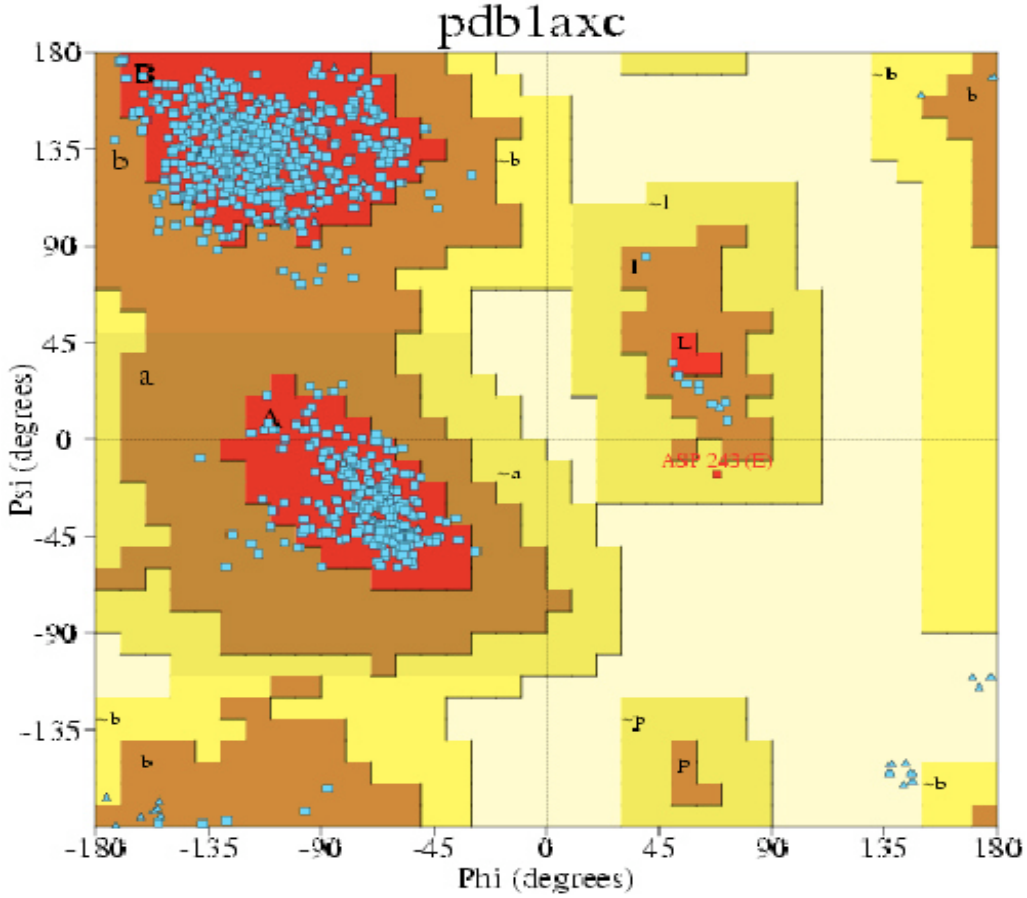
Polypeptide Protein Backbone



Phi, psi, and omega angles

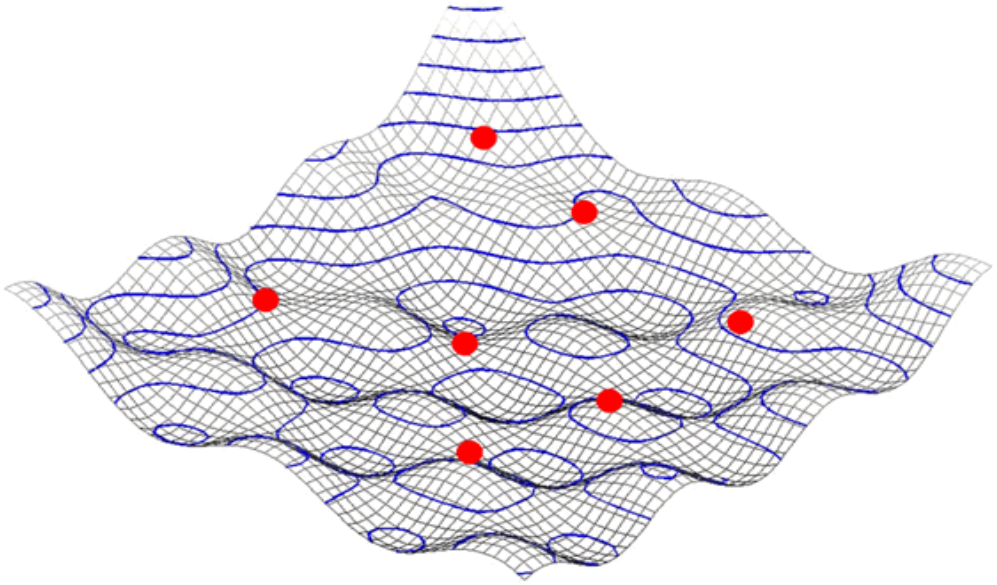
- Proteins are composed of a "backbone" and "side chains". All of these are bonded uses chemical bonds with different associated energies.

The Protein Structure Prediction Problem



- Ramachandran plot visualizing dstribution of phi and psi angles. The red, brown, and yellow regions represent the favored, allowed, and "generously allowed" regions

The Protein Structure Prediction Problem

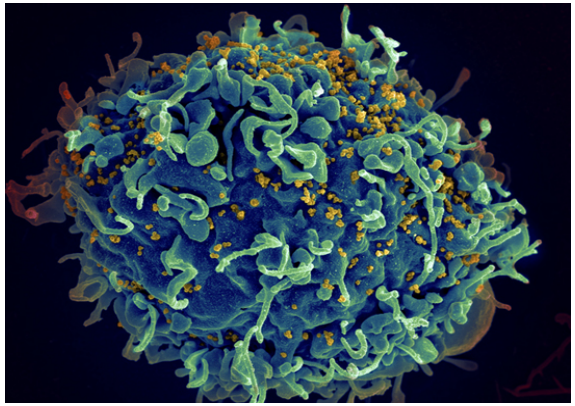


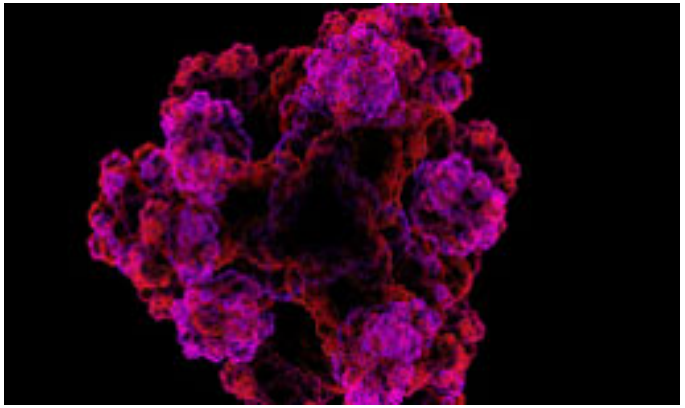
Multi-objective optimization

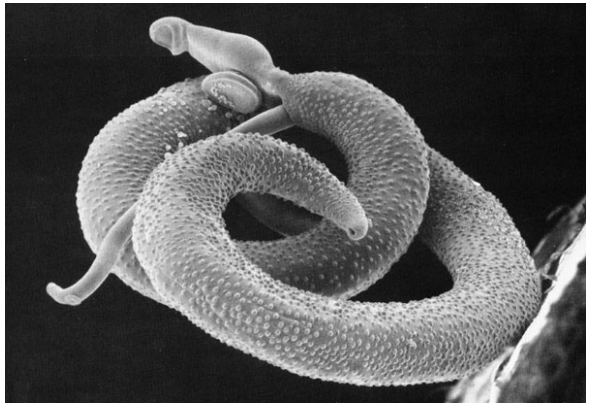
- Finally, proteins have different energies in different solutions which create repulsion/attraction and also help determine structure.
- Most protein predictions algorithms attempt to minimize these energy, because often proteins found in nature have minimum energy conformations – low entropy. (not always)

The Protein Structure Prediction Problem

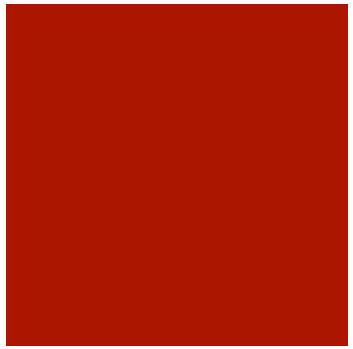
- Impossible and fun problem!
- Why do we care?
  - Proteins control pretty much every function in living organisms.
- Many diseases associated with protein function problems:
  - Cancer: tumor suppression and growth (cellular signaling)
  - HIV, Schistosoma, Malaria: invasion and proliferation (viral and parasitic proteins)
  - Alzheimer's, Mad Cow: proteins gone wrong
  - Medicine: new drugs to target viral proteins or provide deficient proteins (multi-billion dollar pharma industry)





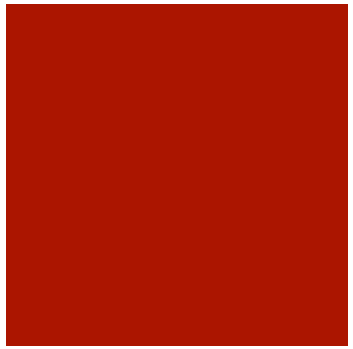


Approaches



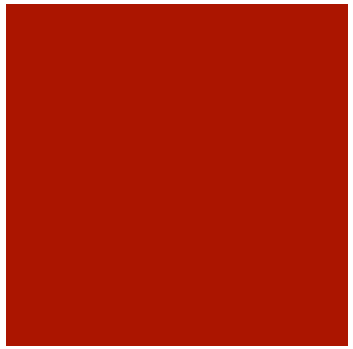
- FoldIt
  - Protein structures are manipulated by players to find the lowest energy
  - A multiplayer online game used to help address the protein structure prediction problem
  - Players download a client that allows them to create scripts for game play and communicate with other players about strategies
  - Strategies are then shared and evolve between players
  - Why crowd-sourcing?
    - Hopefully offers a unique benefit to the protein folding problem because humans can use spatial reasoning that is difficult for computers
    - Humans can create strategies that allow for temporarily low scores for a higher later gain. Computationally it is difficult to teach an algorithm to do this.

Approaches



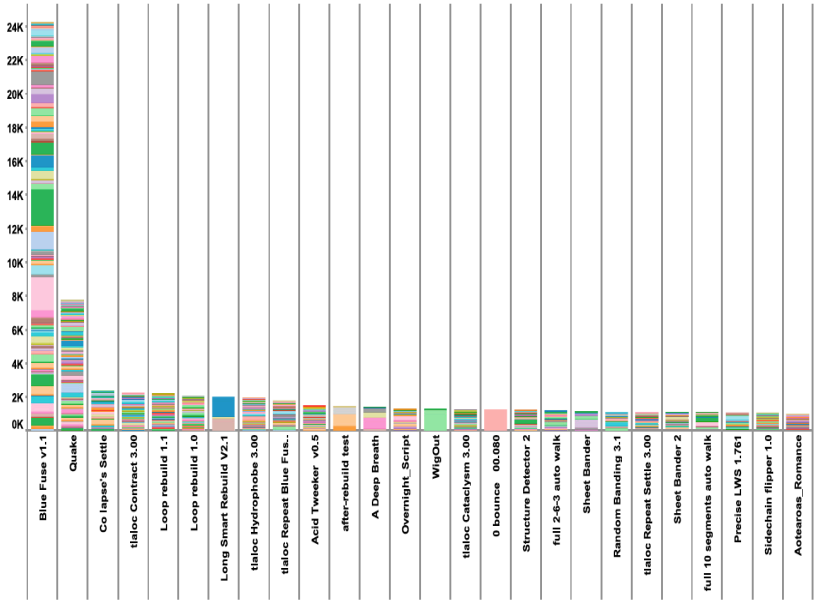
- Rosetta/Rosetta@Home
  - Protein conformations are explored algorithmically to find the lowest energy
  - Large collaborative software used for protein prediction
  - Algorithms are developed by software team
  - Rosetta@Home follows distributed computing model

Approaches

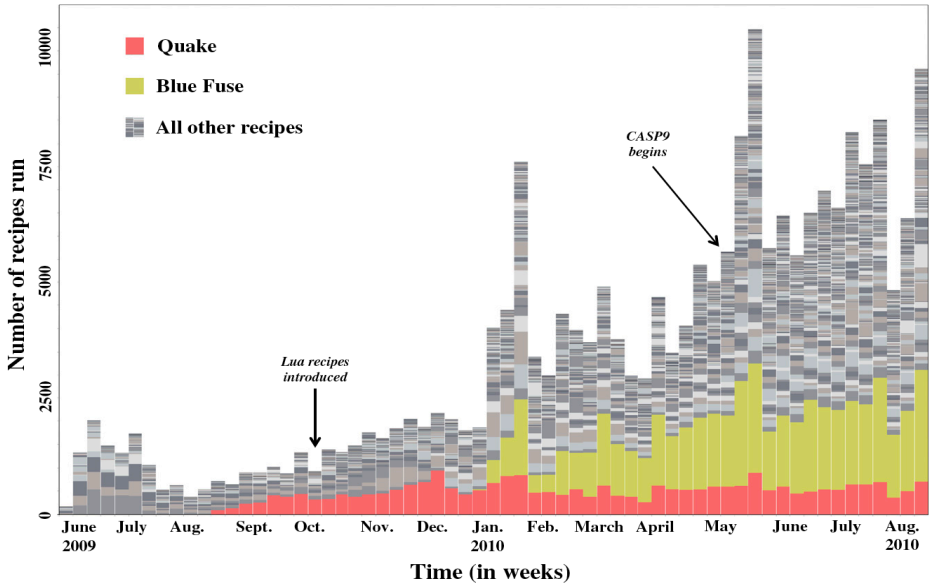


- FoldIt Tools for building "Recipes"
  - Freeze
  - Rebuild
  - Rubber Bands
  - Shake Sidechains
  - Tweak
  - Wiggle
  - The Alignment Tool
- Rosetta
  - Automated versions of tools
  - Homology modeling and other methodologies

Approaches



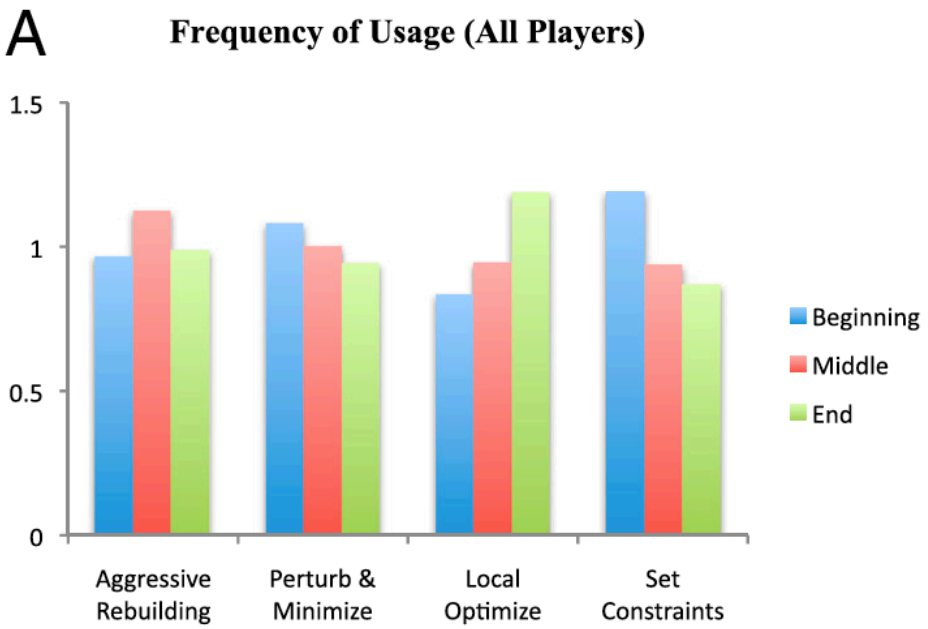
Frequency of BlueFuse Usage

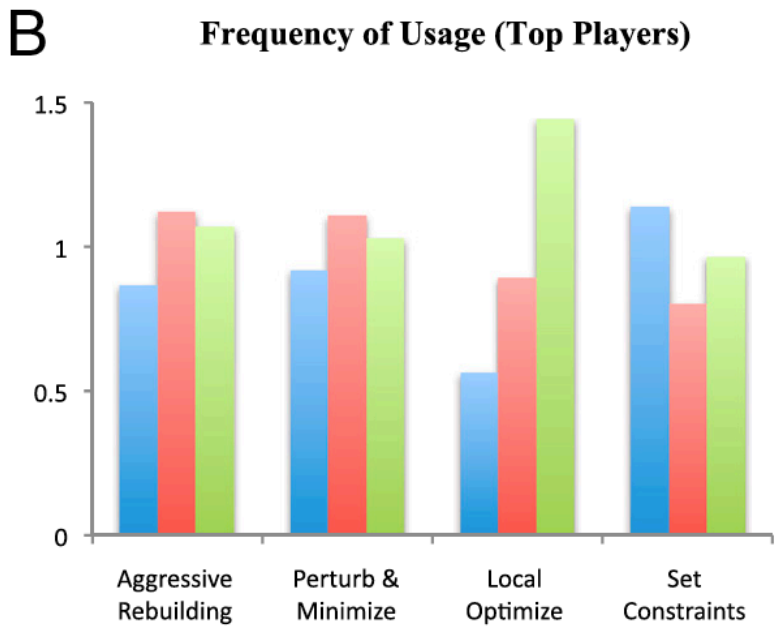


Proliferation of BlueFuse over time

- The "Recipe" that emerged from the crowd was BlueFuse based on overall frequency and proliferation of usage by players.
- From June 2009-August 2010 increased usage of BlueFuse/Quake
- Rosetta developed FastRelax during this same time period

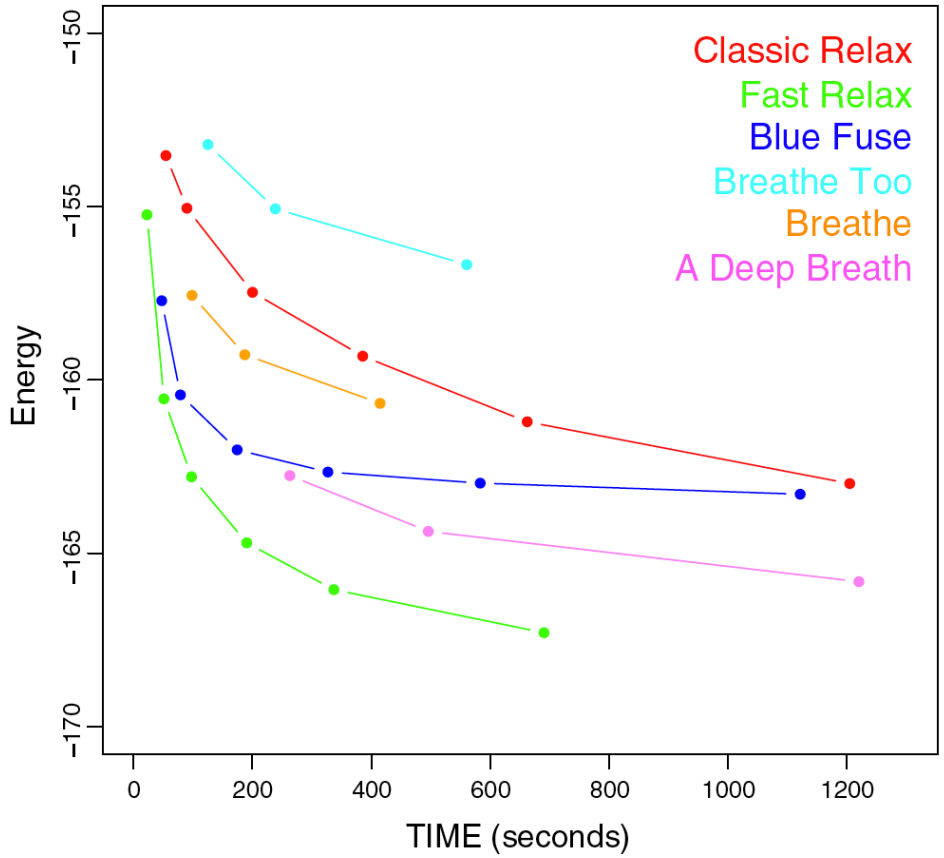
Approaches





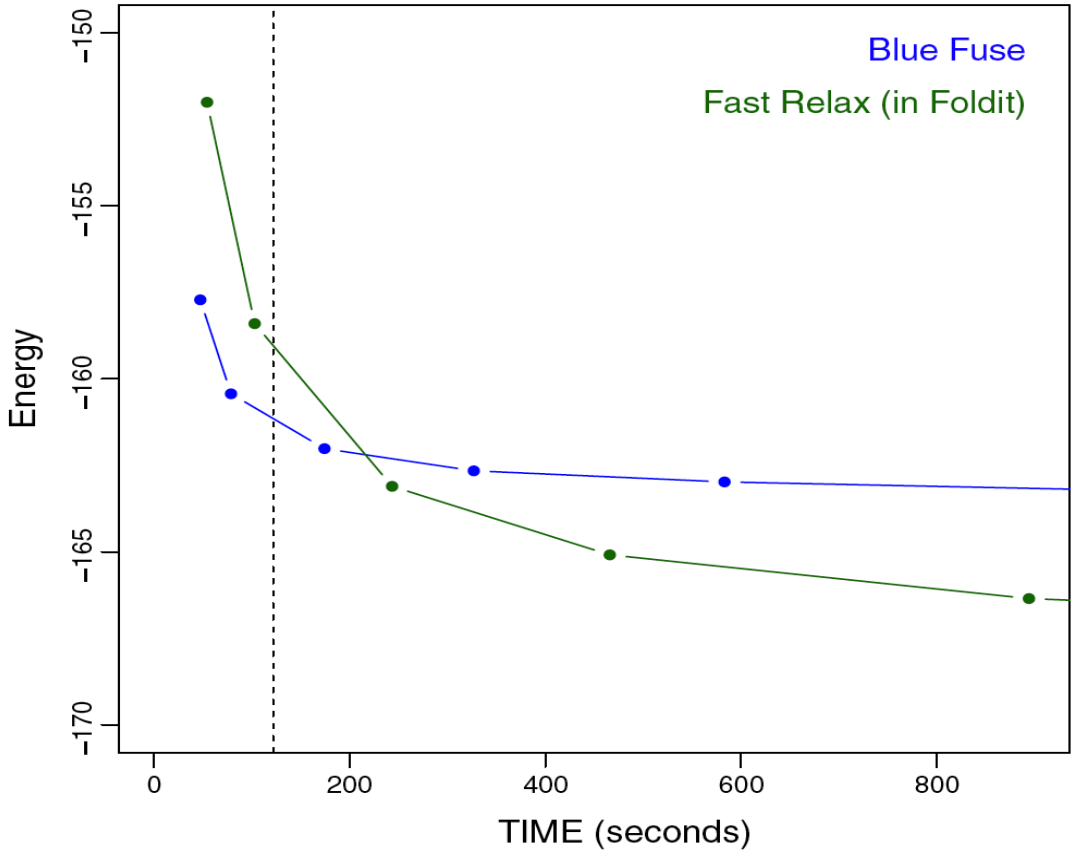
- Beginning
  - Both groups rely on set constraints
  - Default game start behavior
- Middle
  - Perturb and Minimize are the dominant strategies in both groups
  - A stronger preference for them amongst top players
- End
  - Local optimizes are favored by top players, but still the dominant strategy in both groups
  - Logical choice because it optimizes already packed chains along the backbone

Results: Performance



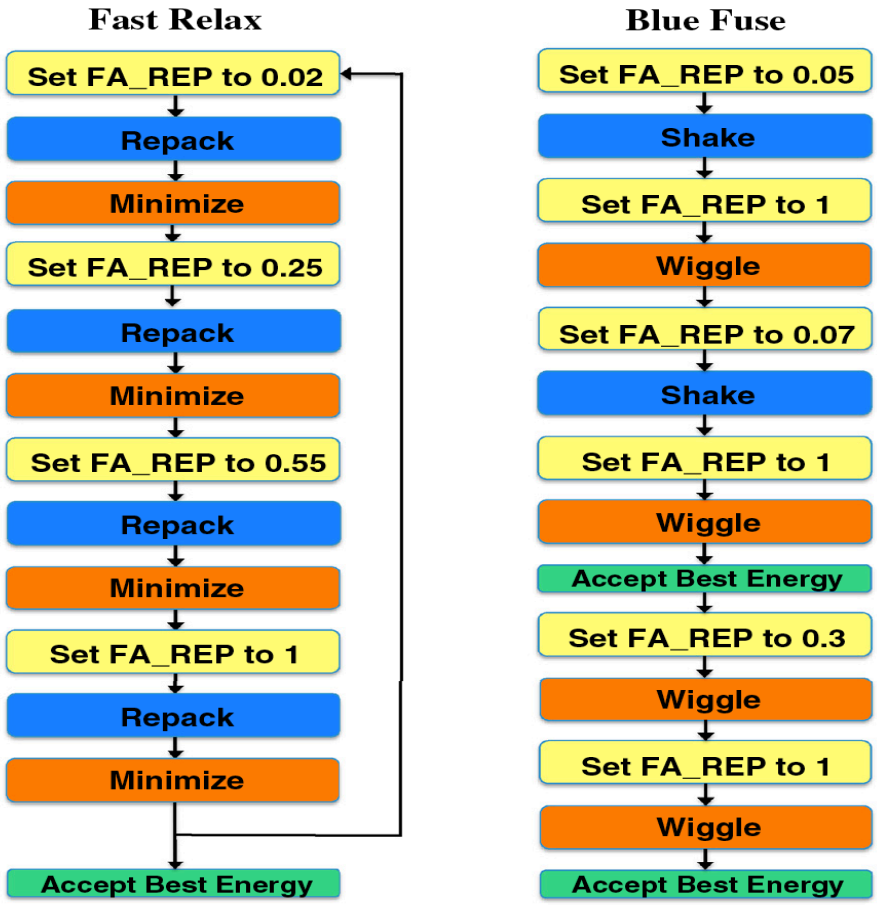
- FoldIt Recipes: Breathe, Breath Too, BlueFuse, A Deep Breath
- Rosetta: Classic Relax, Fast Relax

Results: Performance



- Performance
  - Fast Relax is faster because each cycle doesn't need to be independently evaluated by humans
  - Blue Fuse outperforms Fast Relax at times less than 200s. FoldIt players run Blue Fuse on average 122s. It is unclear from the paper whether this was a found optimum or if longe/shorterr run times might change performance

Results: Comparison



- Key Similarities
  - Both perform a key function of softening repulsion between atoms to find alternate energy minimums (Set FA)
  - Performs a similar cycle of local minimizations and alternating energies to discover minimums (Repack/Minimize | Shake/Wiggle)

Conclusion

- Both methodologies created convergent solutions under similar time frames
- Crowd sourcing using a GWAP was able to reproduce software development efforts pretty accurately and validate the crowd approach.
- Progress was made on determining new methodologies for the protein prediction problem
- FoldIt has yielded viable results:
  - Khatib, F.; Dimaio, F.; Cooper, S.; Kazmierczyk, M.; Gilski, M.; Krzywda, S.; Zabranska, H.; Pichova, I. et al. (2011). "Crystal structure of a monomeric retroviral protease solved by protein folding game players". Nature Structural & Molecular Biology 18 (10): 1175. doi: 10.1038/nsmb.2119

Future Work/Perspectives

- Comparison with Rosetta could be expanded to comparison with machine learning approaches.
- Co-invention in similar time frames implies that it might be more efficient to develop the algorithm in-house
  - The efficiency of crowd-sourcing for this application would need to be validated
- Defining "Top Players"
  - What level of high scores determines a top player and optimizes the search for a solution?
  - How many top players are needed to converge on a solution?
  - Can top players be eliminated and we still converge on a solution?
- Demographics of Players
  - Vital for properly leveraging the crowd sourcing perspective.
  - What is the background of the players? Total number of active players
    - generally low FoldIt has a high learning curve.
- Determining "popularity"
  - How is popularity determined amongst players?
  - What causes a recipe to proliferate amongst players? (high score, simple praise, number of current users)
  - Formal attempts of quantifying this property should be made

Questions?

Highly Optimized ☺