

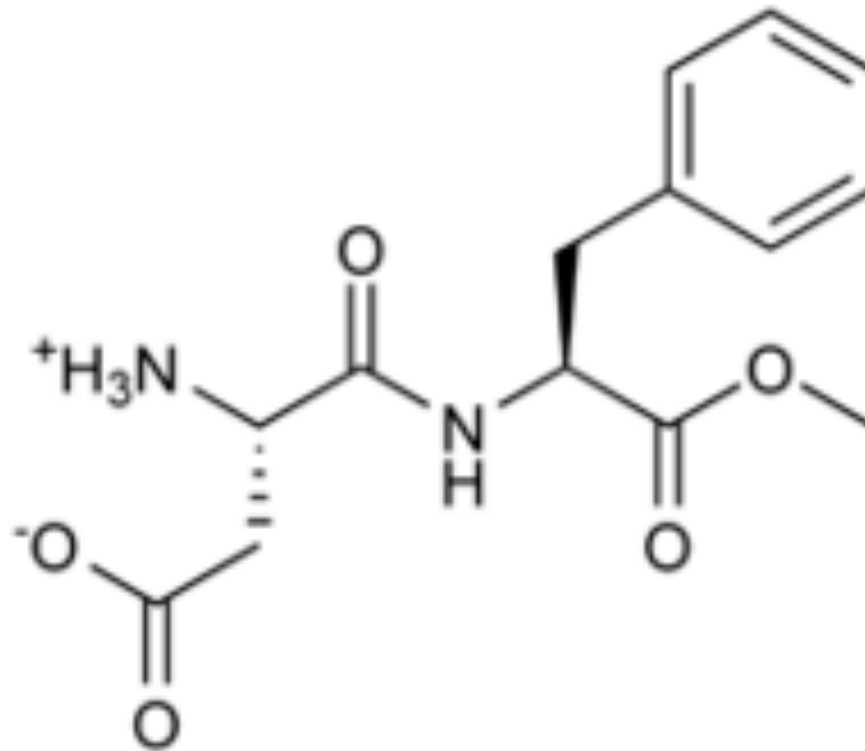
Bioinformatics 2 -- lecture 6

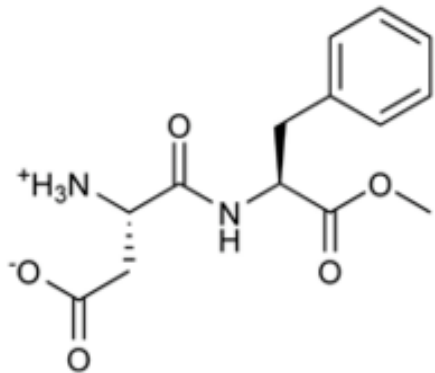
Building a small molecule

Secondary structure prediction

MOE Exercise 1

Building aspartame

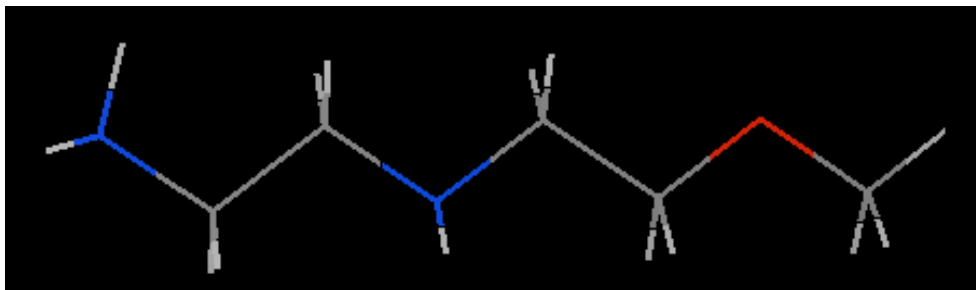




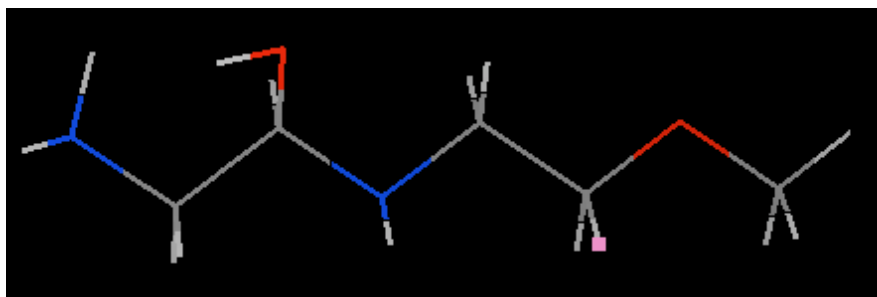
Building aspartame

Starting with an empty Moe window:

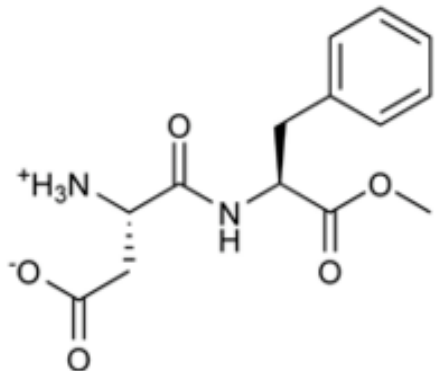
- Edit-->Build-->Molecule (or Builder)
- Create backbone using atoms buttons: N,C,C,N,C,C,O,C
(Notice the chain is made in the fully reduced state.)



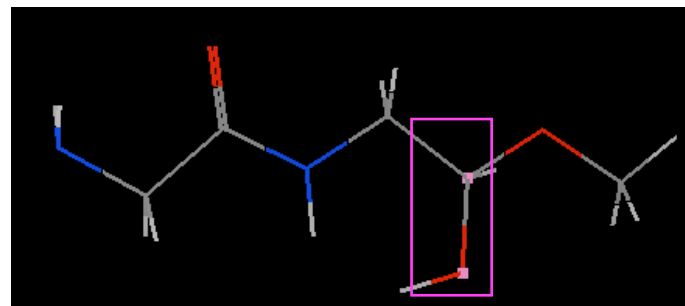
- Add carbonyl oxygens: Select an H, hit O in Builder. the H becomes an O.



Building aspartame

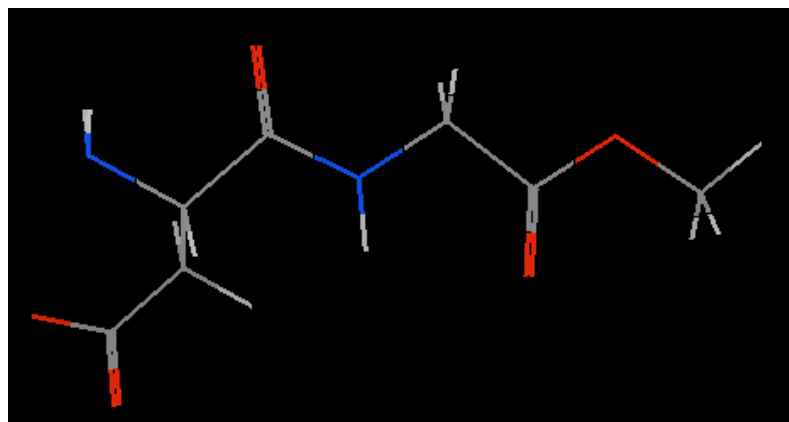


- Select carbonyl groups. Click double bonds (=)

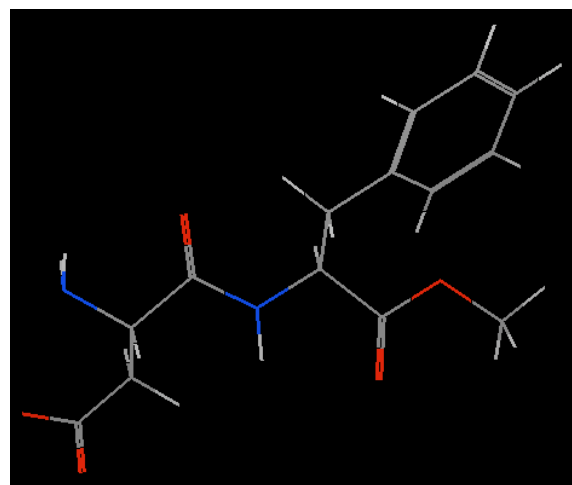


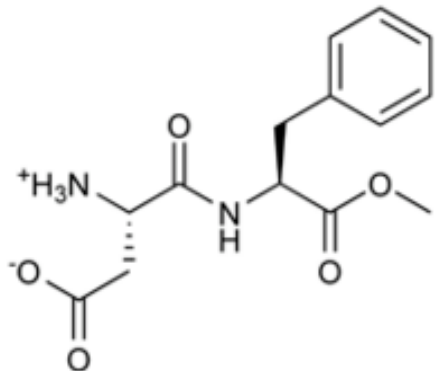
- Add sidechains:

- Select the *back* H on the first alpha-carbon. Click C, then -COO-



- Select the *front* H on the second alpha-carbon. Click C, then benzene.



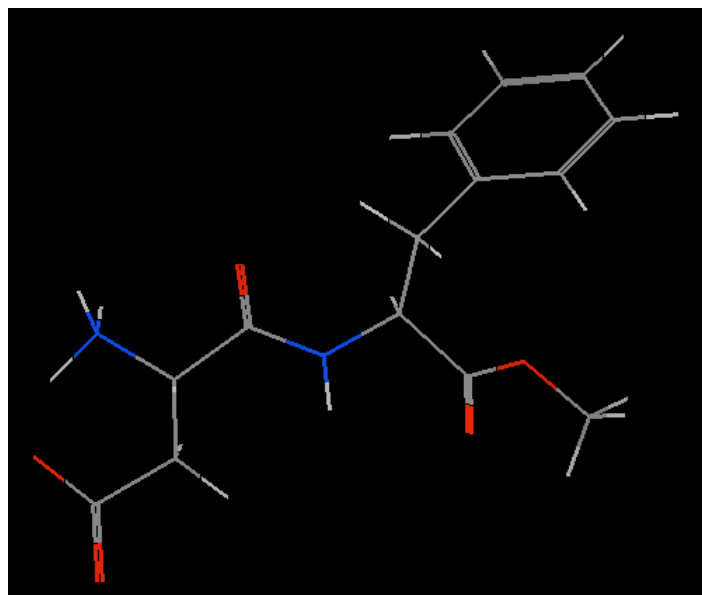


Building aspartame

- Fix ionization of NH_3

Select N. In Builder, click "+1" (a proton is added)

- Fix hybridization of NH.
- Double-click second N. Choose Geometry: "sp2". Click "Apply"
- Click "Minimize".



What is energy minimization?

- Energy minimization is a **molecular simulation** that leads the system to a **lower potential energy**.
- This is similar to the problem of finding the parameters that minimize a function, but there are generally too many parameters. No **optimal** solution is possible.
- Energy minimization is a **heuristic** method.

How is the energy of a molecular model calculated?

- **Energy** is a function of :
 - (1) The coordinates of the atoms.
 - (2) Their names.
 - (3) Their numbers.

The “names” and “numbers” *tell the program* what **element** the atoms are, how they are **bonded**, and what **oxidation state** they have.

Molecular mechanics energy

An *energy function* is a sum over a set of simple functions. This sum is the so-called “energy” of the system.

$$E = f(a_1, a_2) + f(a_1, a_3) + f(a_2, a_3) + f(a_1, a_2, a_3) + \text{etc.}$$

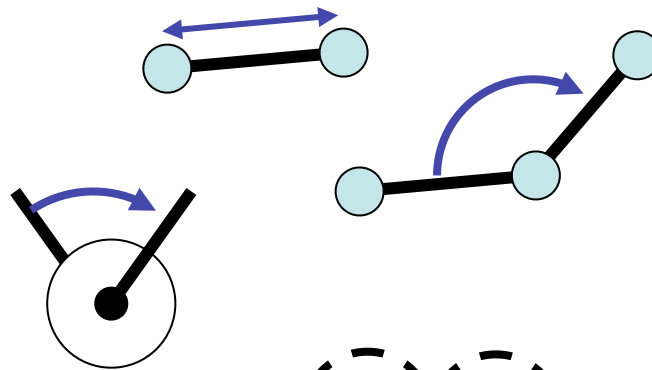
Each simple energy function (f) may have 2,3 or more atoms as parameters: coordinates, names and numbers. Each function uses stored information about each atom name to choose constants within each function. Together the entire set of functions and constants is called a “force field.”

Molecular mechanics

A molecular mechanics energy function includes the following components (and others):

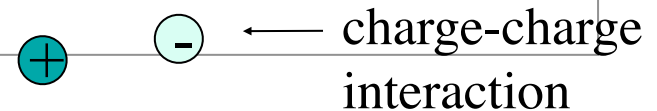
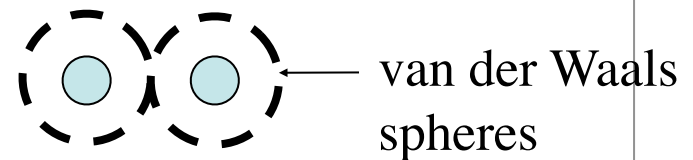
- bonded

- bond lengths
- bond angles
- torsion angles



- non-bonded

- Lennard-Jones or Vander Waals
- Coulomb, or electrostatic



constraint/restraint

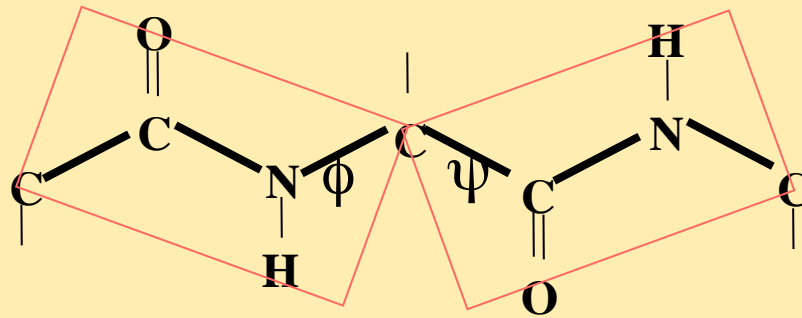
restraint = a function that approaches a minimum as the parameters approach ideal values.

For example, the distance A-B is restrained to 3.8Å using the restraint $E(A,B) = (D_{AB} - 3.8)^2$

constraint = a function that reduces the number of variable parameters in the system.

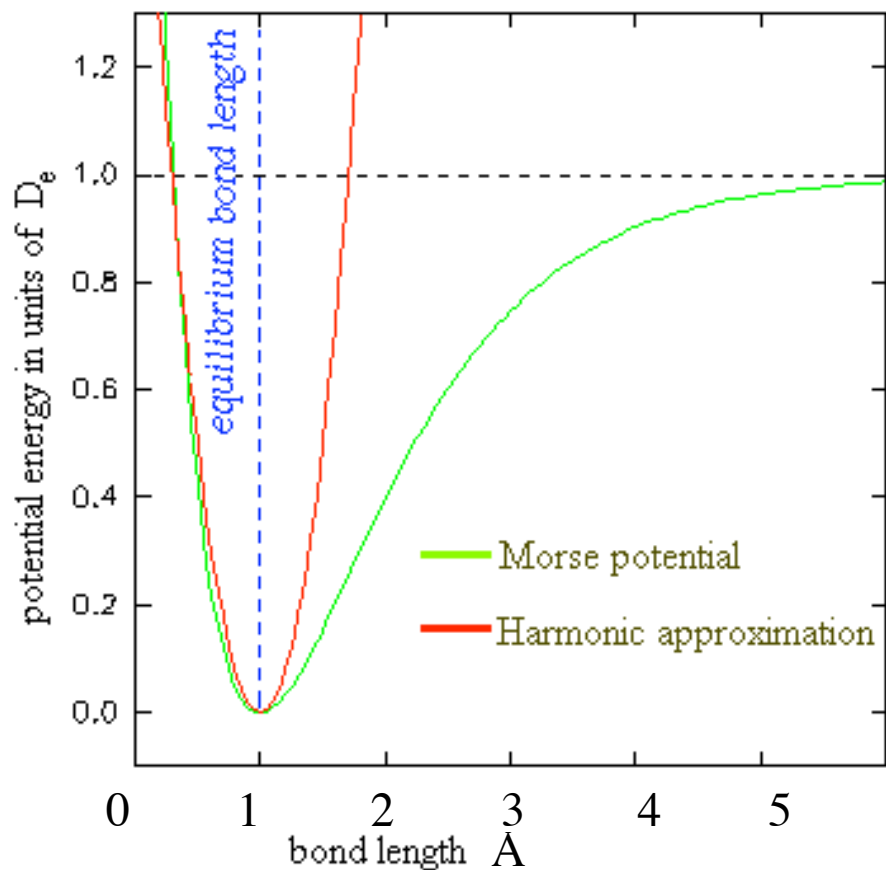
For example, atoms A,B,C and D are constrained to be in the same plane.

Planar groups may be constrained



Distance restraints

Harmonic and Morse potentials are **restraint** functions.



© O. S. Smart, 1995

Restraint forces are applied to move the atoms to their ideal distances/angles.

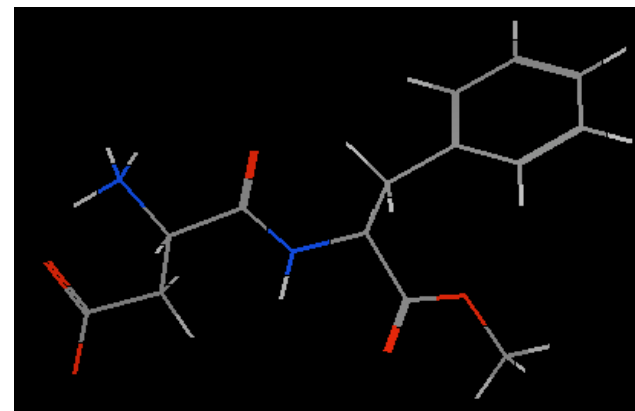
Harmonic potential:

$$E(i, j) = \omega(x_{ij} - T)^2$$

where x_{ij} is the distance between i and j , and T is the ideal distance between i and j .

Building aspartame the easy way

- Close current system
- Edit-->Build-->Protein
- Click ASP, PHE
- Unselect by clicking in empty space.
- Click "C". a methane appears.
- Select it and use the *meta-middle mouse* to move it close to the -COO group
- Select methane C and one O from the Phe-COO. In Builder, click - (single bond)
- Minimize.



Representing protein structure

Secondary structure-- 1D three states

Local structure -- motifs, backbone angles.

Supersecondary structure -- beta-alpha-beta motif, etc.

Inter-residue distances -- 2D contact maps

Tertiary structure, backbone only -- 3D coordinates

Sidechain conformation -- rotamers

Domain-domain interactions -- interface

Quaternary structure -- protein/protein interactions

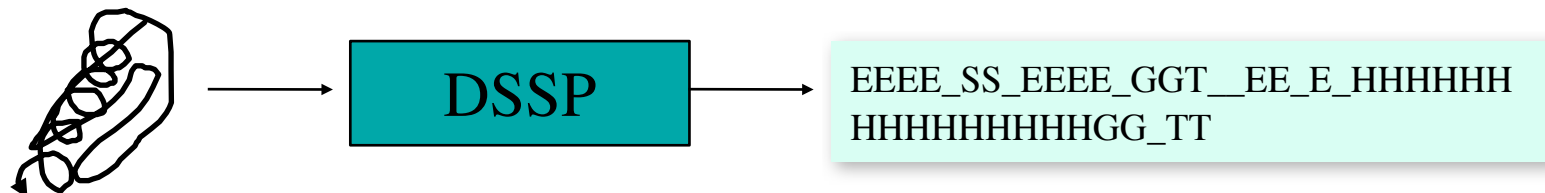
Predicting protein structure

Representation -- Algorithms used.

Secondary structure--	neural nets, HMMs,
Local structure --	sequence profiles
Supersecondary structure --	simulations, rules.
Inter-residue distances --	neural nets, rules, covariance
Tertiary structure--	simulations, homology
Sidechain conformation --	energy calculations
Domain-domain interactions --	energy calculations
Quaternary structure --	energy calculations

secondary structure alphabet

3D protein coordinates may be converted to a 1D secondary structure representation using DSSP or STRIDE



DSSP= Database of Secondary Structure in Proteins

Both programs use hydrogen bonding patterns (see next slide)

DSSP symbols

H = helix backbone angles (-50,-60) and H-bonding pattern (i-> i+4)

E = extended strand backbone angles (-120,+120) with beta-sheet H-bonds (*parallel/anti-parallel are not distinguished*)

S = beta-bridge (isolated backbone H-bonds)

T = beta-turn (specific sets of angles and 1 i->i+3 H-bond)

G = 3-10 helix or turn (i,i+3 H-bonds)

I = Pi-helix (i,i+5 Hbonds) (rare!)

_ = unclassified. None-of-the-above. Generic loop, or beta-strand with no regular H-bonding.

collectively
called

L

for Loop

Accuracy of 3-state predictions

True SS: EEEE_SS_EEEE_GGT__EE_E_HHHHHHHHHHHHHHHGG_TT
Prediction: EEEELLLLHHHHHLLLLEEEEHHHHHHHHHHHHHHHHLL

Q3-score = % of 3-state symbols that are correct

Measured on a "test set"

Test set == An independent set of cases (protein) that were not used to train, or in any way derive, the method being tested.

Best methods:

PHD (Burkhard Rost) -- 72-74% Q3

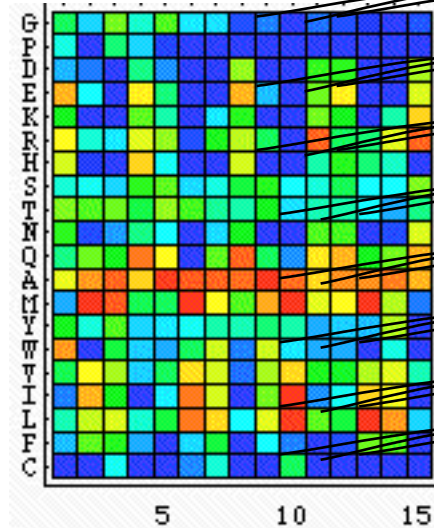
HMMSTR (Bystroff) -- 74-75% Q3

Psi-pred (David T. Jones) -- 76-78% Q3

Psi-Pred: A neural network

input to hidden units weights

hidden units to SS state weights



Sequence profile
(input units)

Hidden units

output units

Prediction (each position) is the state with the greatest sum of weights.

Psi-pred : a neural net

(Step 1) Run PSI-Blast --> output sequence profile

(Step 2) 15-residue sliding window = 315 *weights*, multiplied by *hidden* weights in 1st neural net. Output is 3 weights (1 weight for each state H, E or L) per position.

(Step 3) 45 input weights, multiplied by weights in 2nd neural network, summed. Output is final 3-state prediction.

Making a sequence profile

1. Multiple sequence alignment

VIVAAANRSA
 VIASAVRTA
 VIVDAGRSA
 VIASGVRTA
 VIVAAKRRTA
 VIVSAVRTP
 VIVSAARTA
 VIVSAVRTP
 VIVDAGRRTA
 VIVDAGRRTA
 ... VIVSGARTP ...
 VIVDFGRTP
 VIVSATRTP
 VIVSATRTP
 VIVGALRTP
 VIVSATRTP
 VIVSATRTP
 VIASAARTA
 VIVDAIRTP
 VIVAAYRTA
 VIVSAARTP
 VIVDAIRTP
 VIVSAVRTP
 VIVAAHRTP

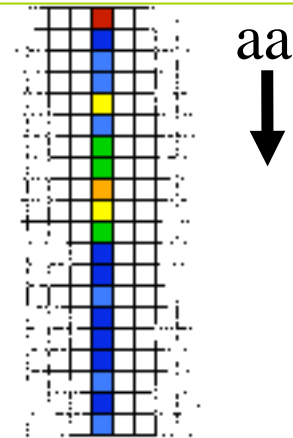
3. sum weights of each amino acid.

$$P_{ij} = \frac{\sum_{k=seqs} w_k \delta(s_{kj} = aa_i)}{\sum_{k=seqs} w_k}$$

2. sequence weights from phylo.tree



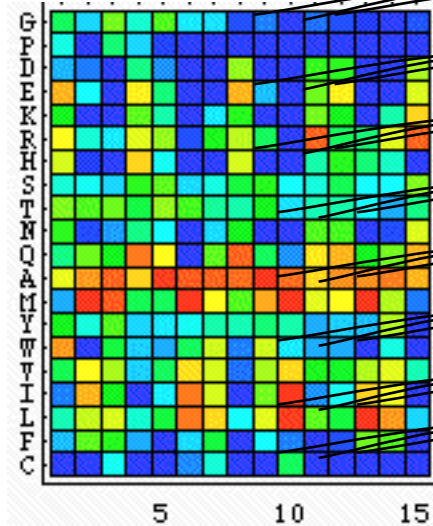
4. Sequence profile, probabilities of 20 amino acids



Red = high prob ratio (LLR>1)
 Green = background prob ratio (LLR≈0)
 Blue = low prob ratio (LLR<-1)

Psi-Pred: Training the neural network (NN)

weights are found that *minimize errors*



Sequence profile

Hidden units

output units

NN output is compared with the true SS. Weights are "back propagated."

Protein database provides both input and output

```

True SS:  EEEE_SS_EEEE_GGT_EE_E_HHHHHHHHHHHHHHHHGG_TT
Prediction: EEEE_LLLL_HHHHHH_LLLL_EEEEE_HHHHHHHHHHHHHHHHLL
Errors:   0000000011111100000010100000000000000000100
    
```

What can you do with a secondary structure prediction?

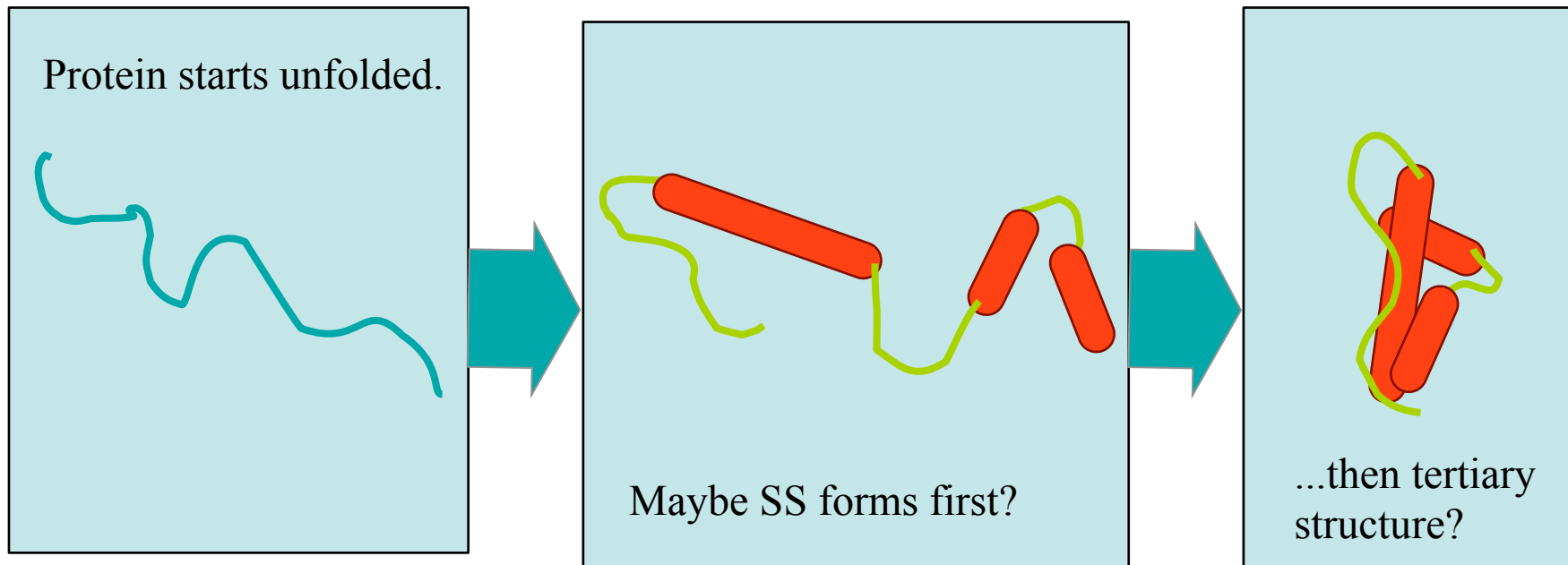
- (1) Find out if a homolog of unknown structure is missing any of the SS (secondary structure) units, i.e. a helix or a strand.
- (2) Find out whether a helix or strand is extended/shortened in the homolog.
- (3) Model a large insertion or terminal domain
- ~~(4) Test remote homology (compare 3-state pred to known SS when sequence homology is very low, i.e. $< 20\%$)~~

Why does it work?

Proteins fold via a 2-state model: folded or unfolded

Usually, no intermediates are observed.

If secondary structure depends on the entire sequence,
then why is a 15-residue window enough to predict SS?



Simple homology-based model -- a MOE exercise (Ex 1)

- ✓ Open course web page and link the “Sequence 1”
- ✓ Open MOE, go to Sequence Editor (ctrl-q)
- ✓ SE: Edit-->Create Sequence (Peptide sequence only)
Paste Sequence 1 from the web page. Call it Strepto.
- ✓ SE:Measure-->Predict secondary structure (look at it)
- ✓ SE:Homology-->SearchPDB (Load chain 1, Search)
- ✓ Load alignment, then Load all (don't load query).
Display-->compound name
- ✓ Delete all sequence except query and 1EM7.A

Ex 1, continued.

- ✓ Align the sequences: SE: Homology--> Align
- ✓ Color the residues: SE: Display-->Color residues (Function)

✓ Change 1EM7.A into your query sequence one residue at a time using: SE: Edit-->Protein...-->Mutate

Select an amino acid, then click on a residue in 1EM7.A to mutate it. **Do not mutate the query!!** Watch the MOE window swap sidechains as you mutate.

- ✓ When done, look at it. Are there any bad contacts?

Ex 1, continued.

✓ When done mutating (the new 1EM7.A is the same sequence as the query), **energy minimize**:

MOE: Selection-->Protein-->backbone

MOE:Edit-->Potential-->Fix

This prevents the backbone atoms from moving during energy minimization.

✓MOE: Selection-->Invert

MOE: Edit -->Potential--> Unfix (this allows sidechains to move)

✓Compute-->energy minimize-->Forcefield-->AMBER94
(Look at this window. Do you know what the weights do?)

Ex 1, continued.

✓ Set *Selection*--> *Synchronize* to the *on* state.

✓ Select and *Delete residue 10* then connect the two ends together (*residue 9 to 11*). Check the atoms you connected for correct hybridization state.

✓ Select and *Unfix* (*Edit* --> *Potential*--> *Unfix*) residues 8-12, then energy minimize.

Save the file in moe format. Upload the file to the website specified. You can upload again if you want to make changes.