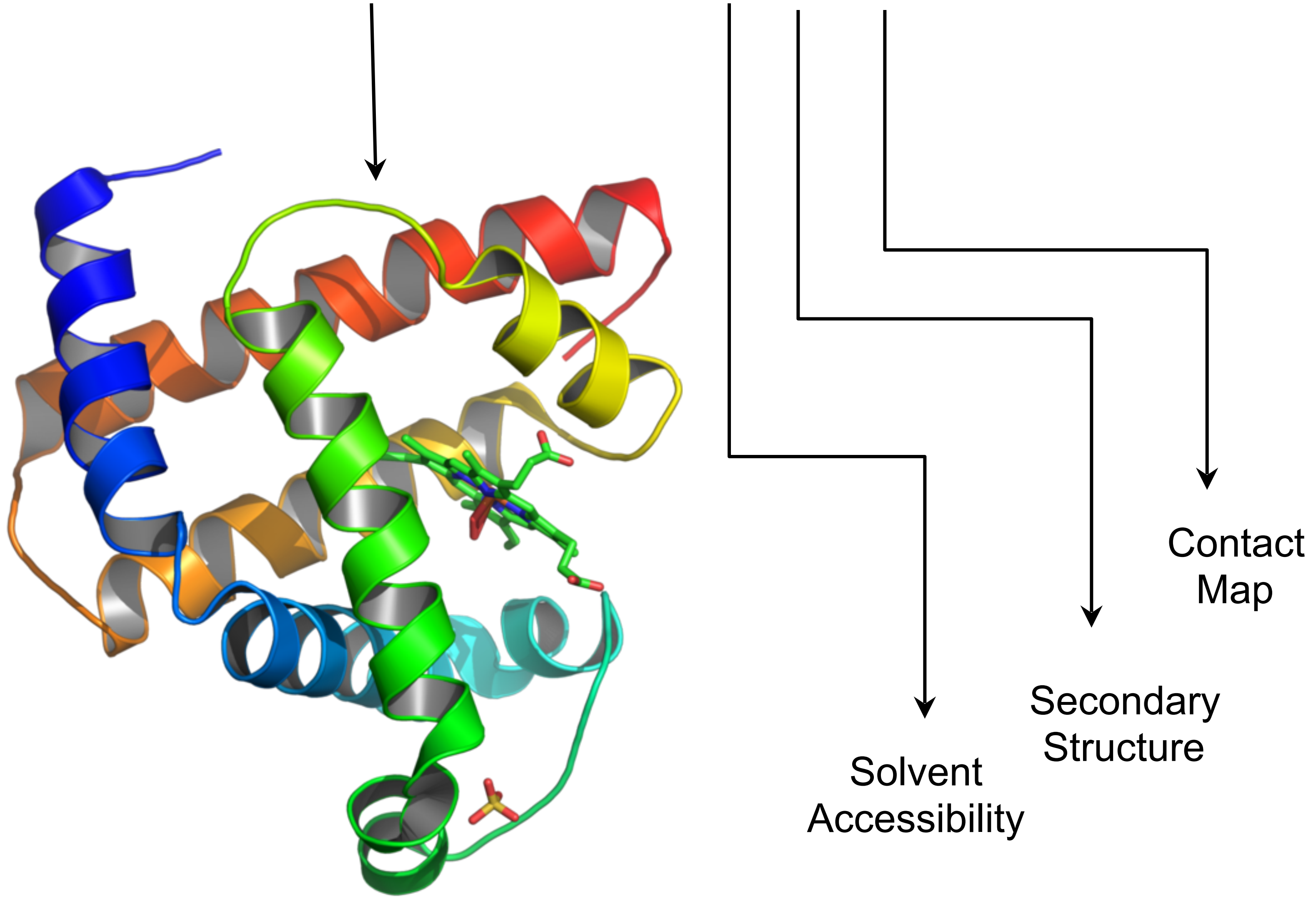
Application of Linear Regression

Protein Structural Similarity Prediction

Protein Structure Prediction (1D ,2D…3D)

EAEASICSEPKKVGRCKGYFPRFYFDSETGKCTPFIYGGCGGNGNNFETLHQCRAICRALG



structure = $\boldsymbol{f}$(sequence)

Comparing protein 3D structure

How to Compute Similarity?

- We have to define a similarity (or distance) measure(s) to assess how different two conformations are.
- Usages:
  - Assessing the success of a folding algorithm.
  - Measure structural similarity between two different proteins which may be related
  - Measure the similarity (or complementarity) of the surfaces two potentially interacting molecules.
  - ...
- No one-size-fits-all quick fix ...

RMSD

- RMSD: Root Mean Squared Deviation
- The most popular distance measure between two conformations
- Average atomic distance
- given two conformations of a chain of N atoms, represent the conformations as two 3N vectors a and b
- RMSD(a,b) is the euclidean distance between a and b, averaged over the N atoms

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}|a_i - b_i|^2} \text{ where } a_i = (x_i^a, y_i^a, z_i^a), b_i = (x_i^b, y_i^b, z_i^b)$$

Structural Superposition: Translation

- Optimal alignment of two chains after removal changes due to rigid body transformations
- Removing translation:
  - Simply align the centroids of the two conformations (centroid = average of all the coordinates).
  - Obtain $c_a = centroid(a)$ and $c_b = centroid(b)$
  - Then "drag" a to centroid of b through $a - [c_a - c_b]$
  - Check that the new c_a is now at c_y
  - Alternatively, can drag both a and b to have (0,0,0) as centroid

Structural Superposition: Rotation

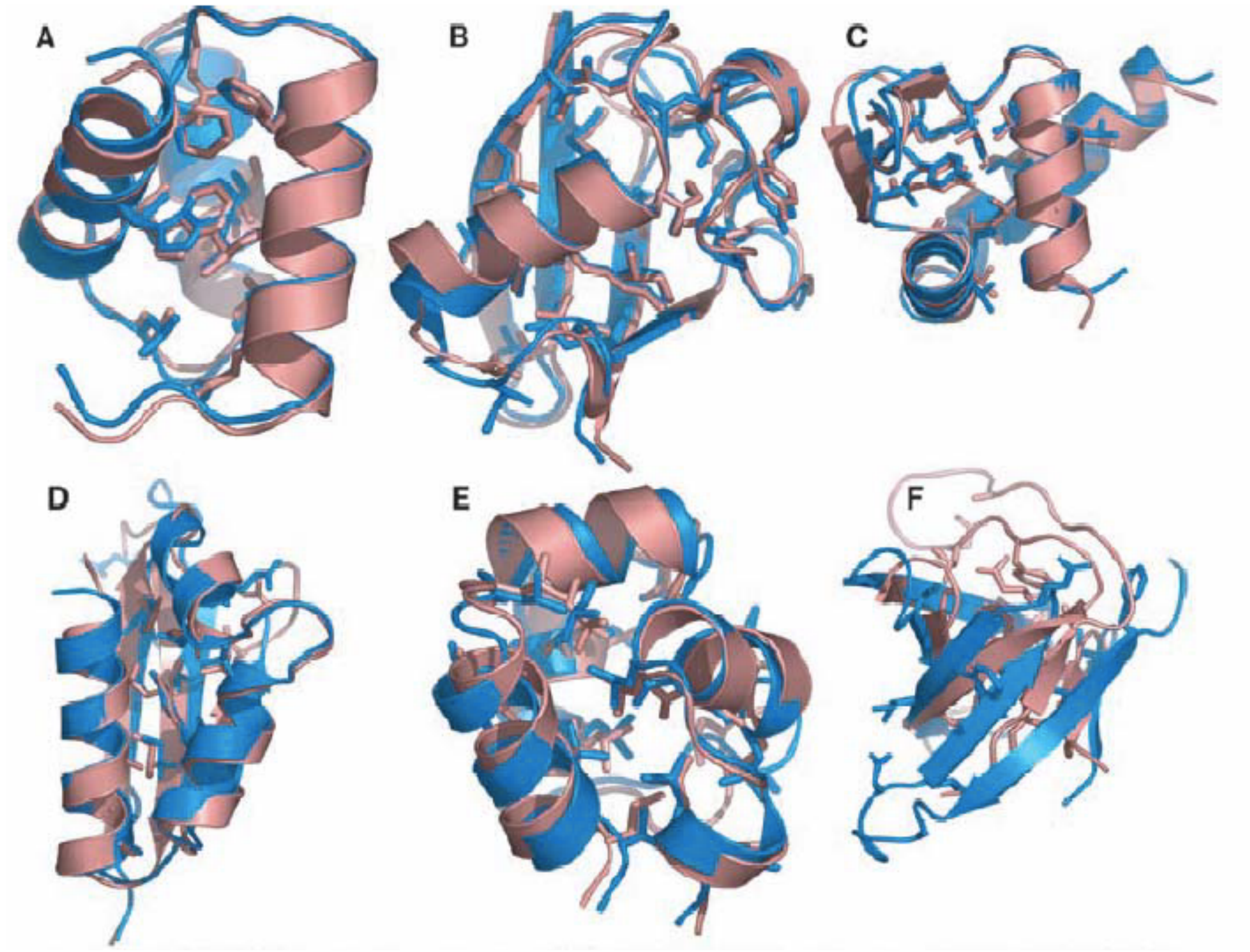
- Generally, we need to find optimal transformation U that minimizes the distance E between b and the transformed a

$$E = \frac{1}{N}\sum_{i=1}^{N} |Ua_i - b_i|^2$$

- Finding the optimal transformation U:
- After some linear algebra:
- Some more linear algebra uses eigenvector decomposition to find U:

$$NE = \sum_{i=1}^{N}(a_i^2 + b_i^2) - 2Tr(B^T A')$$

Structural Superposition - Examples



Structural alignment demo

101M vs. 1MBA

TM-align: Global Quality Score

$$\text{TM-score} = \text{Max}\left[\frac{1}{L_{\text{Target}}} \sum_{i}^{L_{\text{ali}}} \frac{1}{1 + \left(\frac{d_i}{d_0\left(L_{\text{Target}}\right)}\right)^2}\right]$$

TM-align: Global Quality Score

https://zhanglab.ccmb.med.umich.edu/TM-align/

TM-align

Quick & Accurate Structural Alignment

TM-align demo

Protein Similarity Prediction

using Linear Regression…

Protein Structural Similarity Prediction

- We have real-values Y: TM-score
  - Continuous (Y)

- What about features (X)?
  - Overall PSSM for each proteins (20 + 20 = 40)
  - Overall secondary structure content (3 + 3 = 6)
  - Overall solvent accessibility content (2 + 2 = 4)

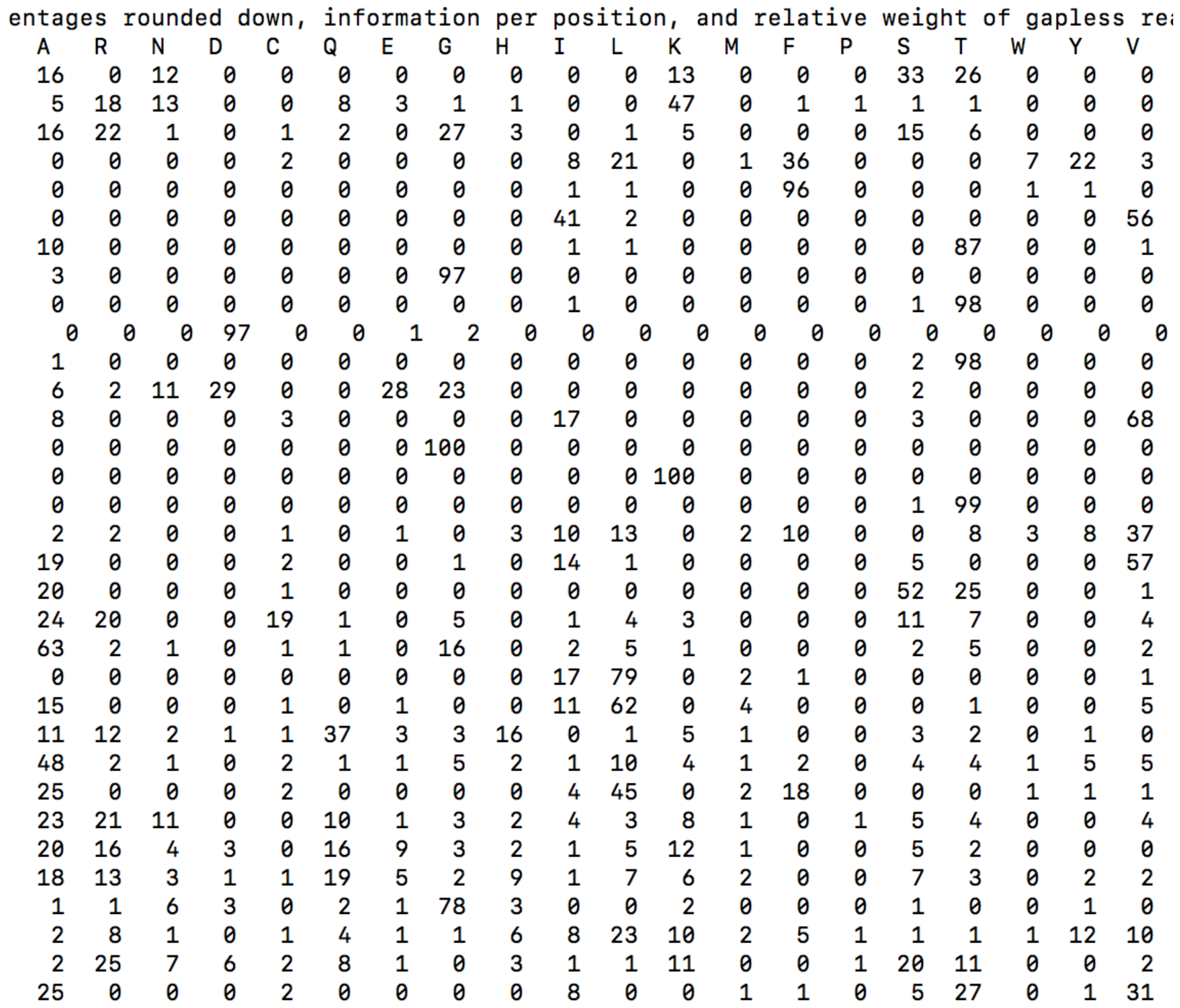
Position Specific Scoring Matrix (PSSM)

Run psiblast against non redundant (nr) sequence database

blastpgp -d <nr_db> -j 3 -b 1 -a 80 –i <protein.seq> -Q <protein.pssm>

```
Last position-specific scoring matrix computed, weighted observed percentages rounded down, information per position, and relative weight of gapless real matches to pseudocounts
           A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V   A   R   N   D   C   Q   E   G   H   I   L   K   M   F   P   S   T   W   Y   V
    1 S    2 -2  2 -3 -3 -2 -2 -3 -3 -4 -4  2 -3 -5 -3  4  4 -5 -4 -3   16   0  12   0   0   0   0   0   0   0   0  13   0   0   0  33  26   0   0   0  0.70 0.31
    2 K   -1  4  3 -3 -6  2 -1 -4 -1 -6 -6  6 -4 -4 -3 -2 -2 -6 -5 -5    5  18  13   0   0   8   3   1   1   0   0  47   0   1   1   1   1   0   0   0  1.04 0.44
    3 R    2  4 -2 -4 -2 -1 -3  4  1 -5 -4  0 -4 -5 -4  2  0 -5 -5 -4   16  22   1   0   1   2   0  27   3   0   1   5   0   0   0  15   6   0   0   0  0.62 0.34
    4 Y   -5 -6 -7 -7  1 -6 -7 -7 -4  1  2 -6 -1  6 -7 -6 -5  5  5 -1    0   0   0   0   2   0   0   0   0   8  21   0   1  36   0   0   0   7  22   3  1.40 0.52
    5 F   -7 -8 -8 -9 -8 -8 -8 -8 -6 -4 -3 -8 -5 10 -9 -8 -7 -1  0 -6    0   0   0   0   0   0   0   0   0   1   1   0   0  96   0   0   0   1   1   0  3.33 0.99
    6 V   -5 -7 -7 -8 -5 -7 -7 -8 -8  6 -2 -7 -3 -5 -7 -6 -5 -7 -6  6    0   0   0   0   0   0   0   0   0  41   2   0   0   0   0   0   0   0   0  56  1.95 0.69
    7 T    1 -6 -5 -6 -6 -6 -6 -6 -7 -3 -5 -6 -5 -7 -6 -3  8 -7 -7 -3   10   0   0   0   0   0   0   0   0   1   1   0   0   0   0   0  87   0   0   1  2.35 0.90
    8 G   -2 -7 -5 -6 -7 -7 -7  8 -7 -9 -9 -6 -8 -8 -7 -5 -6 -8 -8 -8    3   0   0   0   0   0   0  97   0   0   0   0   0   0   0   0   0   0   0   0  2.75 0.90
    9 T   -5 -6 -5 -6 -6 -6 -6 -7 -7 -4 -6 -6 -6 -7 -6 -3  8 -8 -7 -5    0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1  98   0   0   0  2.89 1.03
   10 D   -7 -7 -3  9 -9 -5 -3 -3 -6 -8 -9 -6 -8 -9 -7 -5 -6 -10 -8 -8    0   0   0  97   0   0   1   2   0   0   0   0   0   0   0   0   0   0   0   0  3.05 0.99
   11 T   -4 -6 -5 -6 -6 -6 -6 -7 -7 -6 -6 -6 -6 -7 -6 -2  8 -8 -7 -5    1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2  98   0   0   0  2.85 1.02
   12 E   -1 -2  2  5 -7 -3  4  3 -3 -7 -7 -4 -6 -7 -5 -2 -4 -7 -6 -6    6   2  11  29   0   0  28  23   0   0   0   0   0   0   0   2   0   0   0   0  1.08 0.49
   13 V    0 -7 -7 -7  1 -6 -6 -7 -7  4 -3 -6 -2 -5 -6 -2 -4 -7 -5  7    8   0   0   0   3   0   0   0   0  17   0   0   0   0   0   3   0   0   0  68  1.62 0.63
   14 G   -5 -8 -6 -7 -8 -7 -7  8 -7 -9 -9 -7 -8 -8 -7 -5 -7 -8 -8 -9    0   0   0   0   0   0   0 100   0   0   0   0   0   0   0   0   0   0   0   0  2.96 0.99
   15 K   -6 -3 -5 -6 -9 -4 -4 -7 -6 -8 -8  8 -7 -9 -6 -6 -6 -9 -7 -8    0   0   0   0   0   0   0   0   0   0   0 100   0   0   0   0   0   0   0   0  3.03 1.07
   16 T   -5 -7 -5 -6 -6 -6 -6 -7 -7 -6 -7 -6 -6 -8 -7 -2  8 -8 -7 -5    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1  99   0   0   0  2.96 1.09
   17 V   -2 -3 -5 -6 -1 -5 -4 -6  1  2  1 -5  0  3 -6 -4  1  2  2  5    2   2   0   0   1   0   1   0   3  10  13   0   2  10   0   0   8   3   8  37  0.69 0.37
   18 A    2 -6 -6 -7  1 -6 -6 -5 -7  3 -3 -6 -4 -4 -6 -1 -4 -7 -4  6   19   0   0   0   2   0   0   1   0  14   1   0   0   0   0   5   0   0   0  57  1.32 0.60
   19 S    3 -5 -4 -5 -1 -5 -5 -4 -6 -6 -6 -5 -5 -7 -5  6  4 -7 -6 -4   20   0   0   0   1   0   0   0   0   0   0   0   0   0   0  52  25   0   0   1  1.44 0.71
   20 C    3  4 -3 -5  6 -2 -4 -1 -5 -3 -2 -1 -3 -5 -5  1  1 -6 -5 -1   24  20   0   0  19   1   0   5   0   1   4   3   0   0   0  11   7   0   0   4  0.72 0.41
   21 A    6 -2 -4 -6 -1 -3 -5  2 -4 -3 -2 -3 -3 -4 -5 -2  0 -6 -6 -3   63   2   1   0   1   1   0  16   0   2   5   1   0   0   0   2   5   0   0   2  1.12 0.57
   22 L   -6 -7 -8 -8 -6 -7 -8 -8 -7  3  6 -7  1 -3 -7 -7 -6 -6 -6 -2    0   0   0   0   0   0   0   0   0  17  79   0   2   1   0   0   0   0   0   1  1.89 0.74
   23 L    2 -6 -7 -7 -1 -6 -4 -6 -6  2  5 -6  2 -4 -6 -5 -3 -6 -5  0   15   0   0   0   1   0   1   0   0  11  62   0   4   0   0   0   1   0   0   5  1.19 0.55
   24 Q    1  2 -1 -2 -3  6 -1 -2  6 -5 -4  0 -2 -6 -5 -2 -2 -6 -2 -5   11  12   2   1   1  37   3   3  16   0   1   5   1   0   0   3   2   0   1   0  0.96 0.52
   25 A    5 -2 -3 -5  0 -3 -3  0  0 -2  0 -1 -2 -1 -4 -1 -1 -1  1  0   48   2   1   0   2   1   1   5   2   1  10   4   1   2   0   4   4   1   5   5  0.55 0.36
   26 A    3 -6 -6 -7  0 -5 -6 -6 -4  0  4 -6  1  4 -6 -5 -5 -2 -1 -2   25   0   0   0   2   0   0   0   0   4  45   0   2  18   0   0   0   1   1   1  1.02 0.50
   27 K    3  4  2 -4 -4  2 -2 -2  0 -1 -2  1 -2 -5 -3  0  0 -5 -4 -1   23  21  11   0   0  10   1   3   2   4   3   8   1   0   1   5   4   0   0   4  0.42 0.31
   28 A    2  3  0 -1 -4  3  1 -2  0 -4 -2  2 -2 -5 -3  0 -1 -5 -4 -4   20  16   4   3   0  16   9   3   2   1   5  12   1   0   0   5   2   0   0   0  0.42 0.29
   29 A    2  2 -1 -2 -3  4  0 -2  4 -3 -1  1  0 -4 -4  0 -1 -5 -1 -2   18  13   3   1   1  19   5   2   9   1   7   6   2   0   0   7   3   0   2   2  0.37 0.28
   30 G   -4 -4  1 -2 -7 -2 -5  7  1 -7 -7 -2 -4 -7 -6 -3 -6 -7 -4 -6    1   1   6   3   0   2   1  78   3   0   0   2   0   0   0   1   0   0   1   0  1.90 0.73
   31 Y   -2  1 -3 -4 -1  0 -2 -4  2  1  2  1  1  1 -4 -3 -2  1  3  1    2   8   1   0   1   4   1   1   6   8  23  10   2   5   1   1   1   1  12  10  0.30 0.22
   32 R   -2  4  1  0 -1  2 -2 -4  1 -4 -4  2 -4 -6 -2  3  2 -6 -5 -3    2  25   7   6   2   8   1   0   3   1   1  11   0   0   1  20  11   0   0   2  0.59 0.37
   33 T    3 -5 -5 -6  0 -5 -5 -5 -6  1 -4 -5 -2 -4 -5  0  4 -6 -3  4   25   0   0   0   2   0   0   0   0   8   0   0   1   1   0   5  27   0   1  31  0.96 0.51
```

Weighted Observed Percentages Rounded



```
entages rounded down, information per position, and relative weight of gapless re
 A   R   N   D   C   Q   E   G   H   I   L   K   M   F   P   S   T   W   Y   V
16   0  12   0   0   0   0   0   0   0   0  13   0   0   0  33  26   0   0   0
 5  18  13   0   0   8   3   1   1   0   0  47   0   1   1   1   1   0   0   0
16  22   1   0   1   2   0  27   3   0   1   5   0   0   0  15   6   0   0   0
 0   0   0   0   2   0   0   0   0   8  21   0   1  36   0   0   0   7  22   3
 0   0   0   0   0   0   0   0   0   1   1   0   0  96   0   0   0   1   1   0
 0   0   0   0   0   0   0   0   0  41   2   0   0   0   0   0   0   0   0  56
10   0   0   0   0   0   0   0   0   1   1   0   0   0   0   0  87   0   0   1
 3   0   0   0   0   0   0  97   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1  98   0   0   0
  0   0   0  97   0   0   1   2   0   0   0   0   0   0   0   0   0   0   0   0
 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2  98   0   0   0
 6   2  11  29   0   0  28  23   0   0   0   0   0   0   0   2   0   0   0   0
 8   0   0   0   3   0   0   0   0  17   0   0   0   0   0   3   0   0   0  68
 0   0   0   0   0   0   0 100   0   0   0   0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0 100   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1  99   0   0   0
 2   2   0   0   1   0   1   0   3  10  13   0   2  10   0   0   8   3   8  37
19   0   0   0   2   0   0   1   0  14   1   0   0   0   0   5   0   0   0  57
20   0   0   0   1   0   0   0   0   0   0   0   0   0   0  52  25   0   0   1
24  20   0   0  19   1   0   5   0   1   4   3   0   0   0  11   7   0   0   4
63   2   1   0   1   1   0  16   0   2   5   1   0   0   0   2   5   0   0   2
 0   0   0   0   0   0   0   0   0  17  79   0   2   1   0   0   0   0   0   1
15   0   0   0   1   0   1   0   0  11  62   0   4   0   0   0   1   0   0   5
11  12   2   1   1  37   3   3  16   0   1   5   1   0   0   3   2   0   1   0
48   2   1   0   2   1   1   5   2   1  10   4   1   2   0   4   4   1   5   5
25   0   0   0   2   0   0   0   0   4  45   0   2  18   0   0   0   1   1   1
23  21  11   0   0  10   1   3   2   4   3   8   1   0   1   5   4   0   0   4
20  16   4   3   0  16   9   3   2   1   5  12   1   0   0   5   2   0   0   0
18  13   3   1   1  19   5   2   9   1   7   6   2   0   0   7   3   0   2   2
 1   1   6   3   0   2   1  78   3   0   0   2   0   0   0   1   0   0   1   0
 2   8   1   0   1   4   1   1   6   8  23  10   2   5   1   1   1   1  12  10
 2  25   7   6   2   8   1   0   3   1   1  11   0   0   1  20  11   0   0   2
25   0   0   0   2   0   0   0   0   8   0   0   1   1   0   5  27   0   1  31
```
]

Protein Structural Similarity Prediction using LR

- We have real-valued output (i.e. TM-score)
  - Continuous Y
- We can use PSSM, SS, SA as features (X)
  - From PSSM calculate avg. (PSSM_percentages) / 100 for each of the 20 aa's
  - Predict SS, SA from the sequences
  - SS(H) = #H / len; SS(E) = #E / len; SS(C) = #C / len
  - SA(E) = #E/ len; SA(B) = #B / len
- We can train LR to predict TM-score given two protein sequences
  - Estimate w's
  - Gradient descent algorithm
- Calculate accuracy to estimate performance
  - Mean squared error = (true TM-score – predicted TM-score)^2