Schedule for Lecture 3

- Secondary structure types and secondary structure preferences of certain amino acid types
- Artificial Neural Networks (ANN)
  - Idea
  - Types and architectures
  - Examples
- PSIPRED secondary structure prediction
- JUFO secondary structure prediction
Protein structure depends on amino acid sequence and interactions

(a) Primary structure

Backbone hydrogen bonds

(b) Secondary structure

Side chain interactions

(c) Tertiary structure
Secondary Structure: Build from Backbone Hydrogen Bonds

- **α-Helix:**
  - Periodicity = 3.6
  - Rise = 1.5Å
  - Pitch = 5.4Å

- **β-Sheet:**
  - Periodicity = 2
  - Translation = 3.4Å
  - Distance = 5.4Å
**α-Helix**

- Most abundant secondary structure
- 3.6 amino acids per turn
- Hydrogen bond formed between every fourth residue
- Average length: 10 amino acids, or 3 turns
- Varies from 5 to 40 amino acids
β-Sheet

- 5-10 amino acids in one portion of the chain with another 5-10 farther down the chain
- Interacting regions may be adjacent with a short loop, or far apart

Parallel β-sheet

Anti-Parallel β-sheet
Other Secondary Structure Types and Classifications

- 3-10 helix (N+3)
- $\alpha$-helix (N+4)
- $\pi$-helix (N+5)

- parallel sheet ↑↑↑
- anti-parallel sheet ↓↑↓
- mixed sheet ↓↑↑
- edge parallel strand ↑↑
- edge anti-parallel strand ↓↑

- type 1,2,3 turns
# Secondary Structure Propensities of Amino Acids in Numbers

## Table 1 Amino acid parameter sets

<table>
<thead>
<tr>
<th>Name</th>
<th>$\Xi^a$</th>
<th>$\alpha^b$</th>
<th>$\nu^c_\nu$</th>
<th>$\pi^d$</th>
<th>$I^e$</th>
<th>$\alpha^f$</th>
<th>$\beta^g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALA</td>
<td>1.28</td>
<td>0.05</td>
<td>1.00</td>
<td>0.31</td>
<td>6.11</td>
<td>0.42</td>
<td>0.23</td>
</tr>
<tr>
<td>GLY</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>6.07</td>
<td>0.13</td>
<td>0.15</td>
</tr>
<tr>
<td>VAL</td>
<td>3.67</td>
<td>0.14</td>
<td>3.00</td>
<td>1.22</td>
<td>6.02</td>
<td>0.27</td>
<td>0.49</td>
</tr>
<tr>
<td>LEU</td>
<td>2.59</td>
<td>0.19</td>
<td>4.00</td>
<td>1.70</td>
<td>6.04</td>
<td>0.39</td>
<td>0.31</td>
</tr>
<tr>
<td>ILE</td>
<td>4.19</td>
<td>0.19</td>
<td>4.00</td>
<td>1.80</td>
<td>6.04</td>
<td>0.30</td>
<td>0.45</td>
</tr>
<tr>
<td>PHE</td>
<td>2.94</td>
<td>0.29</td>
<td>5.89</td>
<td>1.79</td>
<td>5.67</td>
<td>0.30</td>
<td>0.38</td>
</tr>
<tr>
<td>TYR</td>
<td>2.94</td>
<td>0.30</td>
<td>6.47</td>
<td>0.96</td>
<td>5.66</td>
<td>0.25</td>
<td>0.41</td>
</tr>
<tr>
<td>TRP</td>
<td>3.21</td>
<td>0.41</td>
<td>8.08</td>
<td>2.25</td>
<td>5.94</td>
<td>0.32</td>
<td>0.42</td>
</tr>
<tr>
<td>THR</td>
<td>3.03</td>
<td>0.11</td>
<td>2.60</td>
<td>0.26</td>
<td>5.60</td>
<td>0.21</td>
<td>0.36</td>
</tr>
<tr>
<td>SER</td>
<td>1.31</td>
<td>0.06</td>
<td>1.60</td>
<td>-0.04</td>
<td>5.70</td>
<td>0.20</td>
<td>0.28</td>
</tr>
<tr>
<td>ARG</td>
<td>2.34</td>
<td>0.29</td>
<td>6.13</td>
<td>-1.01</td>
<td>10.74</td>
<td>0.36</td>
<td>0.25</td>
</tr>
<tr>
<td>LYS</td>
<td>1.89</td>
<td>0.22</td>
<td>4.77</td>
<td>-0.99</td>
<td>9.99</td>
<td>0.32</td>
<td>0.27</td>
</tr>
<tr>
<td>HIS</td>
<td>2.99</td>
<td>0.23</td>
<td>4.66</td>
<td>0.13</td>
<td>7.69</td>
<td>0.27</td>
<td>0.30</td>
</tr>
<tr>
<td>ASP</td>
<td>1.60</td>
<td>0.11</td>
<td>2.78</td>
<td>-0.77</td>
<td>2.95</td>
<td>0.25</td>
<td>0.20</td>
</tr>
<tr>
<td>GLU</td>
<td>1.56</td>
<td>0.15</td>
<td>3.78</td>
<td>-0.64</td>
<td>3.09</td>
<td>0.42</td>
<td>0.21</td>
</tr>
<tr>
<td>ASN</td>
<td>1.60</td>
<td>0.13</td>
<td>2.95</td>
<td>-0.60</td>
<td>6.52</td>
<td>0.21</td>
<td>0.22</td>
</tr>
<tr>
<td>GLN</td>
<td>1.56</td>
<td>0.18</td>
<td>3.95</td>
<td>-0.22</td>
<td>5.65</td>
<td>0.36</td>
<td>0.25</td>
</tr>
<tr>
<td>MET</td>
<td>2.35</td>
<td>0.22</td>
<td>4.43</td>
<td>1.23</td>
<td>5.71</td>
<td>0.38</td>
<td>0.32</td>
</tr>
<tr>
<td>PRO</td>
<td>2.67</td>
<td>0.00</td>
<td>2.72</td>
<td>0.72</td>
<td>6.80</td>
<td>0.13</td>
<td>0.34</td>
</tr>
<tr>
<td>CYS</td>
<td>1.77</td>
<td>0.13</td>
<td>2.43</td>
<td>1.54</td>
<td>6.35</td>
<td>0.17</td>
<td>0.41</td>
</tr>
</tbody>
</table>

* $\Xi^a$: Steric parameter (graph shape index)
* $\alpha^b$: Polarizability
* $\nu^c_\nu$: Volume (normalized van der Waals volume)
* $\pi^d$: Hydrophobicity
* $I^e$: Isoelectric point
* $\alpha^f$: Helix probability
* $\beta^g$: Sheet probability
# Secondary Structure Propensities of Amino Acids in Numbers

## Table 1 Amino acid parameter sets

<table>
<thead>
<tr>
<th>Name</th>
<th>$\Xi^a$</th>
<th>$\alpha^b$</th>
<th>$\nu_v^c$</th>
<th>$\pi^d$</th>
<th>$\eta^e$</th>
<th>$\alpha^f$</th>
<th>$\beta^g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALA</td>
<td>1.28</td>
<td>0.05</td>
<td>1.00</td>
<td>0.31</td>
<td>6.11</td>
<td>0.42</td>
<td>0.23</td>
</tr>
<tr>
<td>GLY</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>6.07</td>
<td>0.13</td>
<td>0.15</td>
</tr>
<tr>
<td>VAL</td>
<td>3.67</td>
<td>0.14</td>
<td>3.00</td>
<td>1.22</td>
<td>6.02</td>
<td>0.27</td>
<td>0.49</td>
</tr>
<tr>
<td>LEU</td>
<td>2.59</td>
<td>0.19</td>
<td>4.00</td>
<td>1.70</td>
<td>6.04</td>
<td>0.39</td>
<td>0.31</td>
</tr>
<tr>
<td>ILE</td>
<td>4.19</td>
<td>0.19</td>
<td>4.00</td>
<td>1.80</td>
<td>6.04</td>
<td>0.30</td>
<td>0.45</td>
</tr>
<tr>
<td>PHE</td>
<td>2.94</td>
<td>0.29</td>
<td>5.89</td>
<td>1.79</td>
<td>5.67</td>
<td>0.30</td>
<td>0.38</td>
</tr>
<tr>
<td>TYR</td>
<td>2.94</td>
<td>0.30</td>
<td>6.47</td>
<td>0.96</td>
<td>5.66</td>
<td>0.25</td>
<td>0.41</td>
</tr>
<tr>
<td>TRP</td>
<td>3.21</td>
<td>0.41</td>
<td>8.08</td>
<td>2.25</td>
<td>5.94</td>
<td>0.32</td>
<td>0.42</td>
</tr>
<tr>
<td>THR</td>
<td>3.03</td>
<td>0.11</td>
<td>2.60</td>
<td>0.26</td>
<td>5.60</td>
<td>0.21</td>
<td>0.36</td>
</tr>
<tr>
<td>SER</td>
<td>1.31</td>
<td>0.06</td>
<td>1.60</td>
<td>-0.04</td>
<td>5.70</td>
<td>0.20</td>
<td>0.28</td>
</tr>
<tr>
<td>ARG</td>
<td>2.34</td>
<td>0.29</td>
<td>6.13</td>
<td>-1.01</td>
<td>10.74</td>
<td>0.36</td>
<td>0.25</td>
</tr>
<tr>
<td>LYS</td>
<td>1.89</td>
<td>0.22</td>
<td>4.77</td>
<td>-0.99</td>
<td>9.99</td>
<td>0.32</td>
<td>0.27</td>
</tr>
<tr>
<td>HIS</td>
<td>2.99</td>
<td>0.23</td>
<td>4.66</td>
<td>0.13</td>
<td>7.69</td>
<td>0.27</td>
<td>0.30</td>
</tr>
<tr>
<td>ASP</td>
<td>1.60</td>
<td>0.11</td>
<td>2.78</td>
<td>-0.77</td>
<td>2.95</td>
<td>0.25</td>
<td>0.20</td>
</tr>
<tr>
<td>GLU</td>
<td>1.56</td>
<td>0.15</td>
<td>3.78</td>
<td>-0.64</td>
<td>3.09</td>
<td>0.42</td>
<td>0.21</td>
</tr>
<tr>
<td>ASN</td>
<td>1.60</td>
<td>0.13</td>
<td>2.95</td>
<td>-0.60</td>
<td>6.52</td>
<td>0.21</td>
<td>0.22</td>
</tr>
<tr>
<td>GLN</td>
<td>1.56</td>
<td>0.18</td>
<td>3.95</td>
<td>-0.22</td>
<td>5.65</td>
<td>0.36</td>
<td>0.25</td>
</tr>
<tr>
<td>MET</td>
<td>2.35</td>
<td>0.22</td>
<td>4.43</td>
<td>1.23</td>
<td>5.71</td>
<td>0.38</td>
<td>0.32</td>
</tr>
<tr>
<td>PRO</td>
<td>2.67</td>
<td>0.00</td>
<td>2.72</td>
<td>0.72</td>
<td>6.80</td>
<td>0.13</td>
<td>0.34</td>
</tr>
<tr>
<td>CYS</td>
<td>1.77</td>
<td>0.13</td>
<td>2.43</td>
<td>1.54</td>
<td>6.35</td>
<td>0.17</td>
<td>0.41</td>
</tr>
</tbody>
</table>

- $^a$ Steric parameter (graph shape index)
- $^b$ Polarizability
- $^c$ Volume (normalized van der Waals volume)
- $^d$ Hydrophobicity
- $^e$ Isoelectric point
- $^f$ Helix probability
- $^g$ Sheet probability
Secondary Structure Element
Interactions define Protein Topology
Neural Networks

- Neural Networks can be:
  - Biological models
  - Artificial models
- Desire to produce artificial systems capable of sophisticated computations similar to the human brain
  - Compute a known function
  - Approximate an unknown function
  - Pattern Recognition
  - Signal Processing
- Learn to do any of the above
Biological Neurons

- **NEURON**
  - The cell that perform information processing in the brain
  - Fundamental functional unit of all nervous system tissue
  - We are born with about 100 billion neurons
  - A neuron may connect to as many as 100,000 other neurons
  - Signals “move” via electrochemical signals

- The synapses release a chemical transmitter – the sum of which can cause a threshold to be reached – causing the neuron to “fire”

- Synapses can be inhibitory or excitatory
Biological Neurons

- Each consists of: SOMA, DENDRITES, AXON, and SYNAPSE
Brain vs. Computer

- Computers require hundreds of cycles to simulate a firing of a neuron
- The brain can fire all the neurons in a single step: “Parallelism”
- Serial computers require billions of cycles to perform some tasks but the brain takes less than a second (e.g. face recognition)

<table>
<thead>
<tr>
<th></th>
<th>Computer</th>
<th>Human Brain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational units</td>
<td>1 CPU, $10^5$ gates</td>
<td>$10^{11}$ neurons</td>
</tr>
<tr>
<td>Storage units</td>
<td>$10^9$ bits RAM, $10^{10}$ bits disk</td>
<td>$10^{11}$ neurons, $10^{14}$ synapses</td>
</tr>
<tr>
<td>Cycle time</td>
<td>$10^{-8}$ sec</td>
<td>$10^{-3}$ sec</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>$10^9$ bits/sec</td>
<td>$10^{14}$ bits/sec</td>
</tr>
<tr>
<td>Neuron updates/sec</td>
<td>$10^5$</td>
<td>$10^{14}$</td>
</tr>
</tbody>
</table>

6 December 2011 © Jens Meiler
Artificial Neuron = NODE

- Each neuron is connected to other nodes with an associated weight (strength) which typically multiplies the signal transmitted.
- Each neuron has a single threshold value.
- Weighted sum of all the inputs coming into the neuron is formed and the threshold is subtracted from this value = activation.
- Activation signal is passed through an activation function (a.k.a. transfer function) to produce the output of the neuron.
Artificial Neuron = NODE

Definition of a node:

• A node is an element which performs the function
\[ y = f_H(\sum (w_i x_i) + W_b) \]
What is and Artificial Neural Network (ANN)?

- An interconnected assembly of simple processing elements, units, neurons or nodes, whose functionality is loosely based on the animal neuron.
- The processing ability of the network is stored in the interunit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.
- A Neural Network is a system composed of many simple processing elements operating in parallel which can acquire, store, and utilize experiential knowledge.
ANN Characteristics

- **Basis:**
  - a crude low-level model of biological neural systems

- **Powerful:**
  - capable of modeling very complex functions/relationships
  - handles curse of dimensionality

- **Ease of Use:**
  - learns the structure for you, i.e. avoids need for formulating rules
  - user must deal with type of network, complexity, learning algorithms & inputs to use

- **Architecture:** the pattern of nodes and connections between them

- **Learning algorithm, or training method:** method for determining weights of the connections
ANN Architecture: Basic Concept

- A Neural Network generally maps a set of inputs to a set of outputs
- Number of inputs/outputs is variable
- The Network itself is composed of an arbitrary number of nodes with an arbitrary topology
ANN Architecture

- **Connectivity:**
  - fully connected
  - partially connected

- **Feedback**
  - feedforward network: no feedback
    - simpler, more stable, proven most useful
  - recurrent network: feedback from output to input units
    - complex dynamics, may be unstable

- **Number of layers i.e. presence of hidden layers**
Simple Perceptron: Architecture

- Binary logic application
- \( f_H(x) = u(x) \) [linear threshold]
- \( W_i = \text{random}(-1,1) \)
- \( Y = u(W_0X_0 + W_1X_1 + W_b) \)
- Now how do we train it?
Sigmoidal Activation Function

- $F(x) = \frac{1}{1 + e^{-k \sum (w_i x_i)}}$
- Shown for $k = 0.5, 1$ and $10$
- Using a nonlinear function which approximates a linear threshold allows a network to approximate nonlinear functions
Alternative Activation Functions

- Radial Basis Functions
  - Square
  - Triangle
  - Gaussian!
Feedforward, Fully-Connected with Two Hidden Layers

input layer  1\textsuperscript{st} hidden layer  2\textsuperscript{nd} hidden layer  output layer
Artificial Neural Networks maps Non-Linear Relations into Hyperspace

- Simplest problem that linear regression cannot solve: XOR

- With a linear model $y = w_0 + w_1 x_1 + w_2 x_2$ there is no solution for $w_{0,1,2}$ that fulfills all data points.

- Black and white points in diagram cannot be separated with a straight line.

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x'_1$</th>
<th>$x'_2$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Mapping Descriptor Space into Hyperspace
Back Propagation of Errors is the most popular Training Algorithm

- Teaching process of multi-layer neural network employing *backpropagation* algorithm. To illustrate this process the three layer neural network with two inputs and one output:
Back Propagation of Errors is the most popular Training Algorithm

- Each neuron is composed of two units. First unit adds products of weights coefficients and input signals. The second unit realizes a nonlinear function, called neuron activation function. Signal $e$ is the adder output signal, and $y = f(e)$ is the output signal of the nonlinear element. Signal $y$ is also the output signal of the neuron:

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- To teach the neural network we need training data set. The training data set consists of input signals \((x_1\) and \(x_2\)) assigned with corresponding target (desired output) \(z\). The network training is an iterative process. In each iteration weights coefficients of nodes are modified using new data from training data set. Modification is calculated using algorithm described below: Each teaching step starts with forcing both input signals from training set. After this stage we can determine output signals values for each neuron in each network layer. Pictures below illustrate how signal is propagating through the network, Symbols \(w_{(x_m)n}\) represent weights of connections between network input \(x_m\) and neuron \(n\) in input layer. Symbols \(y_n\) represents output signal of neuron \(n\).

\[
y_1 = f_1(w_{(x_1)1}x_1 + w_{(x_2)1}x_2)
\]

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- To teach the neural network we need training data set. The training data set consists of input signals \((x_1 \text{ and } x_2)\) assigned with corresponding target (desired output) \(z\). The network training is an iterative process. In each iteration weights coefficients of nodes are modified using new data from training data set. Modification is calculated using algorithm described below: Each teaching step starts with forcing both input signals from training set. After this stage we can determine output signals values for each neuron in each network layer. Pictures below illustrate how signal is propagating through the network, Symbols \(w_{(xm)n}\) represent weights of connections between network input \(x_m\) and neuron \(n\) in input layer. Symbols \(y_n\) represents output signal of neuron \(n\).

\[ y_2 = f_2(w_{(x_1)2}x_1 + w_{(x_2)2}x_2) \]

---

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- To teach the neural network we need training data set. The training data set consists of input signals ($x_1$ and $x_2$) assigned with corresponding target (desired output) $z$. The network training is an iterative process. In each iteration weights coefficients of nodes are modified using new data from training data set. Modification is calculated using algorithm described below: Each teaching step starts with forcing both input signals from training set. After this stage we can determine output signals values for each neuron in each network layer. Pictures below illustrate how signal is propagating through the network, Symbols $w_{(xm)n}$ represent weights of connections between network input $x_m$ and neuron $n$ in input layer. Symbols $y_n$ represents output signal of neuron $n$.

\[
y_3 = f_3(w_{(x1)3}x_1 + w_{(x2)3}x_2)
\]

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- Propagation of signals through the hidden layer. Symbols $w_{mn}$ represent weights of connections between output of neuron $m$ and input of neuron $n$ in the next layer.

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- Propagation of signals through the hidden layer. Symbols $w_{mn}$ represent weights of connections between output of neuron $m$ and input of neuron $n$ in the next layer.

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- Propagation of signals through the output layer.

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- In the next algorithm step the output signal of the network $y$ is compared with the desired output value (the target), which is found in training data set. The difference is called error signal $d$ of output layer neuron.

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- It is impossible to compute error signal for internal neurons directly, because output values of these neurons are unknown. The idea is to propagate error signal $d$ (computed in single teaching step) back to all neurons, which output signals were input for discussed neuron.

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- It is impossible to compute error signal for internal neurons directly, because output values of these neurons are unknown. The idea is to propagate error signal $d$ (computed in single teaching step) back to all neurons, which output signals were input for discussed neuron.

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- The weights' coefficients $w_{mn}$ used to propagate errors back are equal to this used during computing output value. Only the direction of data flow is changed - signals are propagated from output to inputs one after the other. This technique is used for all network layers. If propagated errors came from few neurons they are added. The illustration is below:

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- The weights' coefficients $w_{mn}$ used to propagate errors back are equal to this used during computing output value. Only the direction of data flow is changed - signals are propagated from output to inputs one after the other. This technique is used for all network layers. If propagated errors came from few neurons they are added. The illustration is below:

\[
\delta_2 = w_{24}\delta_4 + w_{25}\delta_5
\]

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- The weights' coefficients $w_{mn}$ used to propagate errors back are equal to this used during computing output value. Only the direction of data flow is changed - signals are propagated from output to inputs one after the other. This technique is used for all network layers. If propagated errors came from few neurons they are added. The illustration is below:

\[
\delta_3 = w_{34}\delta_4 + w_{35}\delta_5
\]

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- When the error signal for each neuron is computed, the weights coefficients of each neuron input node may be modified. In formulas below $df(e)/de$ represents derivative of neuron activation function (which weights are modified).

$$w'_{(x1)1} = w_{(x1)1} + \eta \frac{df_1(e)}{de} x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta \frac{df_1(e)}{de} x_2$$

http://galaxy.agh.edu.pl/~vlsi/Al/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- When the error signal for each neuron is computed, the weights coefficients of each neuron input node may be modified. In formulas below $df(e)/de$ represents derivative of neuron activation function (which weights are modified).

\[
\begin{align*}
    w'_{(x1)2} &= w_{(x1)2} + \eta \delta_2 \frac{df_2(e)}{de} x_1 \\
    w'_{(x2)2} &= w_{(x2)2} + \eta \delta_2 \frac{df_2(e)}{de} x_2
\end{align*}
\]

http://galaxy.agh.edu.pl/~vlsi/Al/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- When the error signal for each neuron is computed, the weights coefficients of each neuron input node may be modified. In formulas below $df(e)/de$ represents derivative of neuron activation function (which weights are modified).

$$w'_{(x1)3} = w_{(x1)3} + \eta \delta_3 \frac{df_3(e)}{de} x_1$$

$$w'_{(x2)3} = w_{(x2)3} + \eta \delta_3 \frac{df_3(e)}{de} x_2$$

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- When the error signal for each neuron is computed, the weights coefficients of each neuron input node may be modified.

\[ w'_{14} = w_{14} + \eta \delta_4 \frac{df_4(e)}{de} y_1 \]

\[ w'_{24} = w_{24} + \eta \delta_4 \frac{df_4(e)}{de} y_2 \]

\[ w'_{34} = w_{34} + \eta \delta_4 \frac{df_4(e)}{de} y_3 \]

http://galaxy.agh.edu.pl/~vlsi/ai/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- When the error signal for each neuron is computed, the weights coefficients of each neuron input node may be modified.

\[
w'_{15} = w_{15} + \eta \delta_5 \frac{d f_5(e)}{de} y_1 \\
w'_{25} = w_{25} + \eta \delta_5 \frac{d f_5(e)}{de} y_2 \\
w'_{35} = w_{35} + \eta \delta_5 \frac{d f_5(e)}{de} y_3
\]

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Back Propagation of Errors is the most popular Training Algorithm

- When the error signal for each neuron is computed, the weights coefficients of each neuron input node may be modified. In formulas below $df(e)/de$ represents derivative of neuron activation function (which weights are modified).

\[
\begin{align*}
    w'_{46} &= w_{46} + \eta \delta \frac{df_6(e)}{de} y_4 \\
    w'_{56} &= w_{56} + \eta \delta \frac{df_6(e)}{de} y_5
\end{align*}
\]

http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
Simple ANN versus Support Vector Machine (SVM)

- Artificial Neural Network
- Support Vector Machine

Models

Support Vectors

Model
Protein Secondary Structure Prediction Based on PSSMs

COMMUNICATION

Protein Secondary Structure Prediction Based on Position-specific Scoring Matrices

David T. Jones

Department of Biological Sciences, University of Warwick, Coventry CV4 7AL United Kingdom

A two-stage neural network has been used to predict protein secondary structure based on the position specific scoring matrices generated by PSI-BLAST. Despite the simplicity and convenience of the approach used, the results are found to be superior to those produced by other methods, including the popular PHD method according to our own benchmarking results and the results from the recent Critical Assessment of Techniques for Protein Structure Prediction experiment (CASP3), where the method was evaluated by stringent blind testing. Using a new testing set based on a set of 187 unique folds, and three-way cross-validation based on structural similarity criteria rather than sequence similarity criteria used previously (no similar folds were present in both the testing and training sets) the method presented here (PSIPRED) achieved an average Q8 score of between 76.5% to 78.3% depending on the precise definition of observed secondary structure used, which is the highest published score for any method to date. Given the success of the method in CASP3, it is reasonable to be confident that the evaluation presented here gives a fair indication of the performance of the method in general.

© 1999 Academic Press
Protein Secondary Structure Prediction Based on PSSMs

Sequence

Sliding window of 15 amino acids
20 probabilities from PSSM
Protein Secondary Structure Prediction Based on PSSMs

1st Network
315 inputs
75 hidden units
3 outputs

Window of 15 x 3 outputs fed to 2nd network

2nd Network
60 inputs
60 hidden units
3 outputs

Final 3-state Prediction

Conf: [.....]
Pred: 
Pred: CCCEEEHHHCCCCCCCCCCCCCCCCCCCC
AA: MKSFLVVTILALPGLGAQEQNPQTPRCEKDRFFS

Conf: [.....]
Pred:
Pred: CCCEEEHHHCCCCCCCCCCCCCCCCCCCC
AA: DKIAKYIPQYVLSPSGLNNQYPKVALNNQFLPY
Local influences captured by protein primary structure:

Non-local influences captured by protein tertiary structure?


Local influences captured by protein primary structure:

Consider all AA that are distant in sequence but Ca-Ca:
