For my final project I chose to make a simple implementation of the GOR algorithm. The original GOR methods used the PDB files to obtain secondary structure information. While the PDB files do include secondary structure information, upon further examination of the files, the secondary structure is found lack information contained on the PDB website for the given structure. The PDB publishes the secondary structure as given by various methods, and I chose to get my secondary structure information instead from the DSSP: Dictionary of Secondary Structure assignments for Proteins[1]. The entire database can be downloaded in one text file. Using this file, I constructed a smaller database to use for the purpose of creating the initial database of structural information. One thing that must be mentioned is that I do simplify the types of secondary structures in DSSP to α-helices, β-sheets and coils. The β-bridges, 3/10 helices, pi helices, turns and bends are all recorded as coils for the sake of simplicity in structural prediction. The database constructed contains all 267 structures used in GOR IV with the same chain information[2]. Some of the PDB identifiers have been replaced since 1996. In total, nineteen PDB identifiers were updated to current PDB codes and a list of updated PDB codes can be found at the end of this summary. It was initially suggested for my project to look at the differences between the databases for GOR I and GOR IV to see how the prediction changed with just a change in the database. GOR I was created from a total of 25 structures which were globular in nature[3,4]. The descriptors of the original structures used were not specific enough to easily link to the PDB. In addition, approximately 20% of the original structures were of some form of hemoglobin/myoglobin.

I wrote my implementation of GOR using Perl, and thus it does have any specific compiling information. I developed this program using Perl 5.8.6 on a UNIX cluster. Included in the package of data are three executables: makesmall.pl, makedb.pl and gor.pl. These three files will take you through my entire process of getting the necessary DSSP sequences, building the database to be used in the GOR algorithm, and finally running the algorithm to produce a secondary structure prediction. Prior to running any of the programs, a file was created with the PDB identifiers and chain information with one chain per line. The PDB ID was first followed by a '.' and the chain, where 'X' stands for the only chain in the file. The makesmall.pl script will take a file with a list of PDB identifiers with chain information to pull out the information from the DSSP file representing the PDB database of 40355 proteins. The makedb.pl file uses once again the PDB identifier file along with the smaller DSSP database file to create the information values for three types of secondary structures: α-helices, β-sheets and coils, which will be saved to the specified output file. While it is not necessary to use the smaller DSSP file in makedb.pl, it saves computing time by not reading through the entire file. The final step is executing gor.pl with the information values from makedb.pl and a string representing the protein (must contain no spaces). The secondary structure prediction is given as output to the terminal. All perl scripts are directly executable provided that it can find Perl in /usr/bin/perl. If there is a different location for Perl, either the first line of each executable can be modified or each script can be called in the traditional method 'perl <filename> <arguments>'. In addition, the usage for each script can be found by just calling the script without any arguments.

Example Usages:

```
[jld@hpc0 gor]$ ./makesmall.pl pdb_id.txt PDBFIND2.TXT PDBFIND2.small
[jld@hpc0 gor]$ ./makedb.pl pdb_id.txt PDBFIND2.small ss_db.txt
[jld@hpc0 gor]$ ./gor.pl ss_db.txt
MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSYRKQVVIDGETCLLDILDTAGQEEYSAMRDQYMRTGEG
FLCVFAINNTKSFEDIHQYREQIKRVKDSDDVPMVLVGNKCDLAARTVESRQAQDLARSYGIPYIETSAKTRQGVED
AFYTLVREIRQH
```

The protein that was run with my GOR instantiation is PDB identifier 121P which is found in the α/β class of the SCOP database. The top two lines of the example output from gor.pl and the third line is the actual secondary structure information from the PDB website: http://www.pdb.org/pdb/explore/sequence.do?structureId=121P. This protein was not one of the proteins used in the creating the database. I chose it due to the fact that it had quite the mixture of α-helices and β-sheets. The positions with * are the positions that were correctly predicted. In this example 65% of coils, 76% of β-sheets, and 87% of α-helices were correctly predicted. One possible reason for the lower accuracy of coil prediction could be due to the fact that turns, bends, 3/10 helices, pi helices and β-bridges are included in the definition of a coil.

```
MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSYRKQVVIDGETCLLDILDTAGQEEYSAMRDQYMRTGEG
CCHEEEEEEECCCCCCCHHHHHHHHCCCCCCCCCCEHHHCHEEEEECCCCEEEEHHCHCCHHHHHHHHHHHHCCCC
ceeeeeeeeccccccchhhhhhhhhccccccccccccceeeeeeeeeccceeeeeeeeeeccccccchhhhhhhhhhccee
** ****** *****  ******* **********      **** *   ****    **   ************

FLCVFAINNTKSFEDIHQYREQIKRVKDSDDVPMVLVGNKCDLAARTVESRQAQDLARSYGIPYIETSAKTRQGVED
EEEEEEECCCCCCHHHHHHHHHHHHEECCCCCCEEEEECCCCCHHHHHHHHHHHHHHHHHCCCCEEEEECCHHHCCEHH
eeeeeccccchhhhhhhhhhhhhhhcccccccceeeeeccccccccccchhhhhhhhhhhhcccceeeccccccchhhhh
*****  ***   ***********  ****************     ******************  **      **

AFYTLVREIRQH
HHHHHHHHHHHH
hhhhhhhhhhcc
**********
```

While this method is not as refined as GOR V, I think it performs fairly well. I do have some criticisms of this implementation. First, it does not include a scoring system or accuracy of prediction for each position. Second, while Garnier et al dismiss using the DSSP database due to the fact that it will split chains apart when an atomic position is missing in the structure does not seem to be an issue ten years later, at least for the PDB files that were previously defined. This database gives more secondary structure information, which I believe resulted in a better prediction. Third, simplifying the secondary structure may not have been done in best way. For example, classifying 3/10 helices and pi helices as coils could be part of the reason why there is such a high error in predicting the coils. Initial investigations of this did not reveal much difference in the error of the prediction.

References:

[1]*Biopolymers* 1983, **12**, 2577-637. **http://swift.cmbi.ru.nl/gv/dssp/**
[2]*Methods in Enzymology* 1996, **266**, 540-553.
[3]*Journal of Molecular Biology* 1978, **27**, 97-120.
[4]*Journal of Molecular Biology* 1976, **107**, 327-356.

Updated PDB Codes

```
2AAK Replaced 1AAK                    2GLT Replaced 1GLT
2ABK Replaced 1ABK                    2GMF Replaced 1GMF
1NOJ Replaced 1ABM                    2GSR Replaced 1GSR
2AYH Replaced 1AYH                    2MIN Replaced 1MIN
2END Replaced 1END                    2OLB Replaced 1OLB
1FNB Replaced 1FNR                    2OMF Replaced 1OMF
```

```
2PGD Replaced 1PGD
2SIM Replaced 1SIM
2VAA Replaced 1VAA
1BKS Replaced 1WSY
4AAH Replaced 3AAH
1CYO Replaced 3B5C
1G6N Replaced 3GAP
```

Description of all files included in the zipped file:

makesmall.pl – executable to make a smaller DSSP file
makedb.pl – executable to make the information calculations from the DSSP file
gor.pl – executable to give the secondary structure prediction.
pdb_id.txt – file containing all 267 PDB identifiers used to make the database
ss_db.txt – file containing the calculated information for each position in each type of secondary structure
PDBFIND2.TXT – DSSP file as downloaded off of their website on December 10, 2006.
PDBFIND2.small – Smaller DSSP file created from makesmall.pl
JamieDukeFinalProject.doc – this file
JamieDukeFinalProject.pdf – this file saved as a PDF just in case something happened to the doc file if it was opened on a UNIX machine