**Goal:**

1)Input size-genome level

2)make a visualizer

3)change input type from di(crystallized image) to tri/tetra(md simulated image)

**Understanding of the paper:**

**Test\_Seq.txt has 100 sequences of length 1001**

* Why promoter prediction is needed? Pg1

For genome annotation-to understand transcription regulatory networks

* Why has it been tough?? Pg1

Genomic sequences are highly adaptive and diversified to enable survival in diverse conditions.

Variability in the length of 5’ un-translated regions and presence of multiple TSS

High gene density, very less inter-genic spaces or having overlapping coding regions

* What is meant by parameter signals?

Structural/energetic properties like bendability, curvature etc.  
 STRUCTURAL: DNA shape is cumulation of certain parameters. So instead of taking their

Cumulative effect, we consider different parameters. (28)

ENERGETIC: hydrogen bond, stacking energy and solvation energy.

**UNDERSTANDING THE getParameterDetails file:**

**FLOW:**

getParameterDetails-> iterateSequences-> calculateParameters-> calculateMovingAverages-> normalizeMovingAverages->return

then iterateSequences-> combineStructEnergyParams->return

then iterateSequences -> transformStructEnerMap

**WHAT STORES WHAT: UNDERSTANDING THE FUNCTIONS**

**1)readSequenceFile**

Reads the sequence file, stores sequences of size 1001 nucleotides as array elements. Array “content” is an array of 100 such sequences.

We check if content[i] is not none and size is >1000

Content is changes to a dict called sequence\_map with index as the key and value: sequence of 1001 nucletides

**2)getParameterDetails()**

Takes as input the sequence\_map. Then runs the function iterateSequences which calls calculateParameters.

**getParameterDetails-> iterateSequences-> calculateParameters->**

Calculate parameters assigns 31 (‘a’,b,c,d,e,……,’ae’) parameter values to each nucleotide pair in the sequence. On passing one sequence of 1001 nucleotides, we get a param\_map for each letter of 1000 values each. (**1000 nicleotide pairs in 1001 nucleotides)**

The corresponding param\_map is now sent as parameter to calculateMovingAverages. We take a moving window size of 25. Similary normalizedAverage is calculated. A total of 975 values for each pair.

combineStructEnergyParams: takes in as input the parameter to define energyStructparams(understand better)

**The result from getParameterDetails is a map with keys: ['moving\_averages\_map', 'normalized\_params\_map', 'values\_map', 'combined\_params\_map']**

**3)pcaRegressionAlogorithm(param\_map)**

**iterateSequences** is called.It receives the output of getParameterDetails which is a map of maps.

The map’s keys are number of sequences. Each key stores another map containing scores of all 31 parameters corresponding to that sequence. Each parameter of the 31 parameters is also a key which stores a list of 975 scores corresponding to 1001 nucleotides in 1 sequence.

**Iterate** is run for each sequence then.

*Assumptions: Considering there will be no TSS up till 200 nucleotides of where tss ends.*

+ve dataset is the dataset where we know TSS will occur. Samples takes consist of 40nt and 80 nt. Ie out of 1001 nt in a sequence, we are taking +dataset from -35 to+5(40nt) and -75 to +5(80nt).

Similarly –e dataset is the one where we know there will be no TSS. Samples taken correspond to the +ve data set(ie one of 4nt and 80nt each). 40nt: +200 to +240 and 80nt:+200 to +280.

Each seq\_x\_map consists of tss and no\_tss regions ie 2

Returns: a map of 100 keys(each key for each sequence. Corresponding to each key, there is another map with 575 (key, value) pairs. (length-ITR\_WINDOW\_SIZE-ITR\_WINDOW\_SIZE-NO\_TSS\_WINDOW\_LENGTH)

**4)motifsAlgorithm(param\_map, seq):**

Basically runs on the enrgy and struct combined parameters.

Params = ['energyDecreasing\_params', 'structuralIncreasing\_params', 'structuralDecreasing\_params', 'energyIncreasing\_params']