Genome-Wide Compilation of *Arabidopsis* Polyadenylation Signals Using Cray SV1 Accelerated Pattern Recognition

Quinn Li (Miami University)

Dave Strenski (Cray)

Eric Stahlberg (OSC)

Johnny Loke (Miami University)

Chris Wood (Miami University)

What is Arabidopsis ?

Mustard

Cauliflower

Cabbage

Canola

Arabidopsis

(mouse ear cress)

Why Arabidopsis?

- Model organism
- Simple, but has everything that makes up a plant
- Small genome and fast growing, which is good for genetic studies
- Genome sequenced

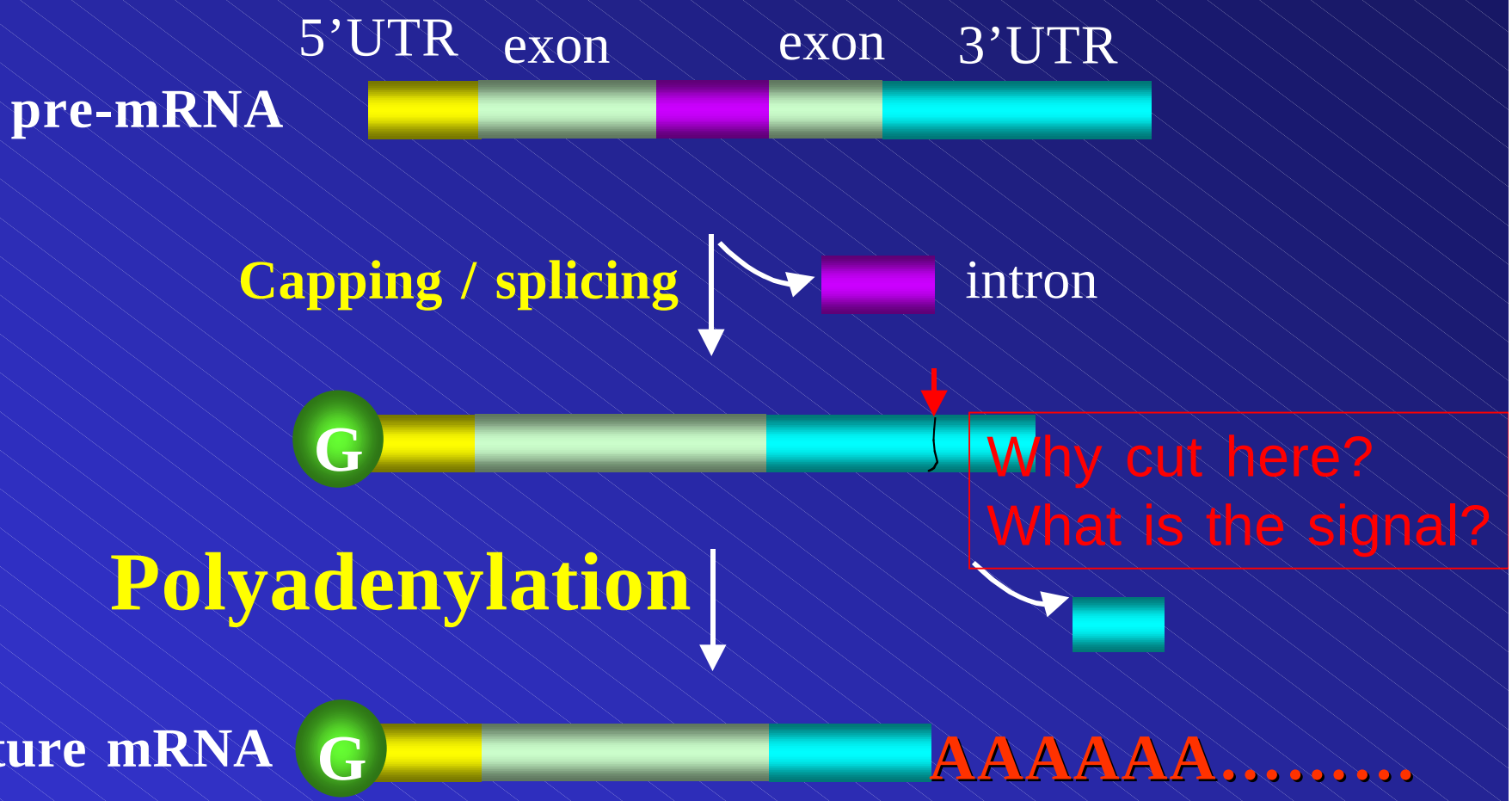
What is Genome?

- The total genetic content of an organism
- Make up of DNA
- DNA use only four characters: A, T, G, C
- Different orders, “sequences”, of the characters become genes, “paragraphs”
- The “words” in the paragraph are the special domains defining the cell’s function

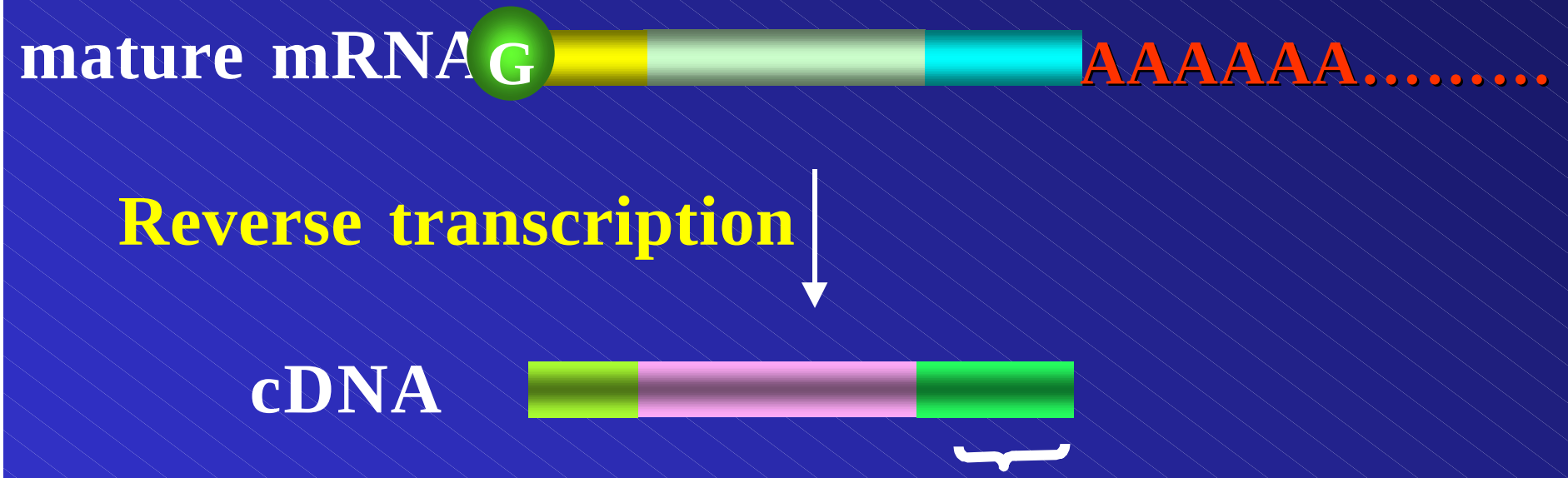
Gene Expression

Book		Gene (DNA)
↓	Transcription & *Processing*	↓
Note		RNA
↓	Translation	↓
Action		Protein
↓	Function	↓
Outcome		Phenotype

RNA processing

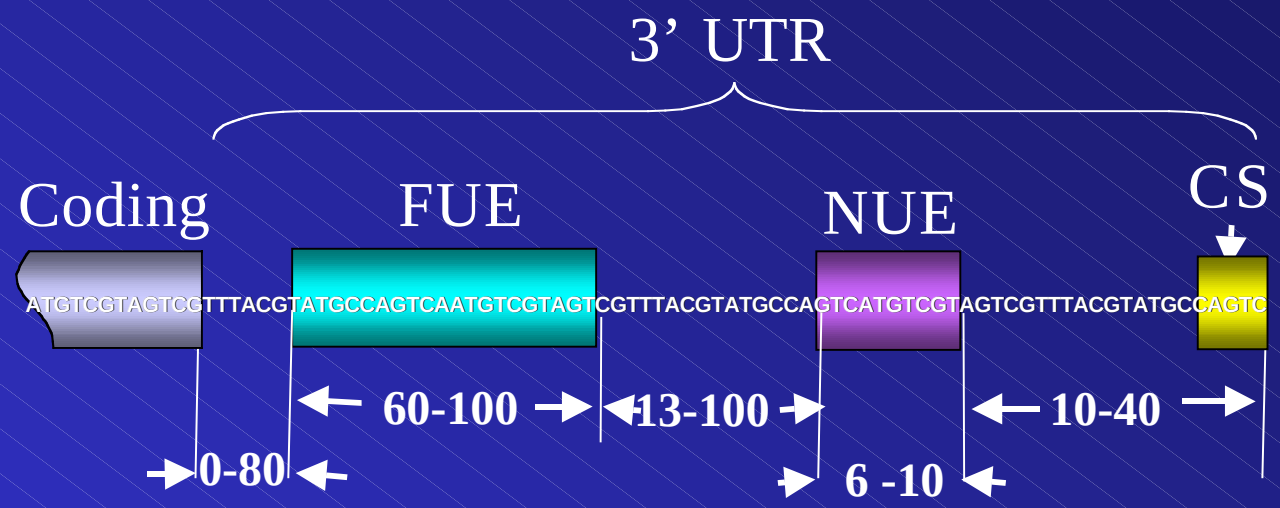


Database Available (cDNA library)



Total 16,225 cDNA in the database

Working model of poly(A) signals



NUE: Near Upstream Element
FUE: Far Upstream Element
CS: Cleavage Site
3’UTR: 3’ UnTranslated Region
Coding: code for protein

Current Project

- Find the consensus sequences that act as signals
- This happens to most of the 26,000 genes (mRNA) in Arabidopsis/all plants
- The signal should have some commonality: pattern or similarity?

Current Approaches

- Select sequences
- Align sequences (exponential problem with number of sequences)
- Create Hidden Markov Models and find consensus
- PROBLEM: too many sequences to align!
- PROBLEM: need to know what to look for first

Using Cray Bioinformatics Libraries to Meet Challenges

- Why Cray libraries
  - Fast and convenient programming interface
  - Large, shared memory, address space for programs
  - Large secondary storage space can store several sets of information at one time

What are Cray Bioinformatics Libraries?

- A set of routines that accesses the special bit manipulation hardware: popcnt, leadz, vector shift, bit matrix multiply (BMM)
- Rapidly search for sequences (patterns), reverse sequences, and slightly off sequences
- Works for DNA and proteins
- Defined API, open source

Approach to Locating Poly(A) Signals

We do not know the pattern nor location

Our problem has 16255 sequences ranging in length from 50 to 3118 characters, with the average being 231, total characters about 3.7 million

Approach to Locating Poly(A) Signals

Here's an example: (next set of slides)

- Each line is a different sequence
- Green is the NUE region of interest
- Red is a matching pattern

Sequences all end with cleavage site

AGTCGTCGTTCTCTAAATAGGCTAGC
AGCTTCGCTATCTAGCTAGCTAGTACCC
ATGACATATAACAGAAATAGCATAT
GTTACATGTACATATCTATAAGATACTAC
CGTAATACAAAATAGTACCAAAT
GTATACATAAATAGACAAGAT
AATACAGATTCAGAGACATATACACAG
ATACAGATTCCACAGAAGATACAGAT
ATACGATCATGCTATACATATCGATC

AGTCGTCGTTCTCTAAATAGGCTAGC

AGCTTCGCTATCTAGCTAGCTAGTACCC

ATGACATATAACAGAAATAGCATAT

GTTACATGTACATATCTATAAGATACTAC

CGTAATACAAAATAGTACCAAAT

GTATACATAAATAGACAAGAT

AATACAGATTCAGAGACATATACACAG

ATACAGATTCCACAGAAGATACAGAT

ATACGATCATGCTATACATATCGATC

AGTCGTCGTTCTCTAAATAGGCTAGC

AGCTTCGCTATCTAGCTAGCTAGTACCC

ATGACATATAACAGAAATAGCATAT

GTTACATGTACATATCTATAAGATACTAC

CGTAATACAAAATAGTACCAAAT

GTATACATAAATAGACAAGAT

AATACAGATTCAGAGACATATACACAG

ATACAGATTCCACAGAAGATACAGAT

ATACGATCATGCTATACATATCGATC

AGTCGTCGTTCTCTAAATAGGCTAGC

AGCTTCGCTATCTAGCTAGCTAGTACCC

ATGACATATAACAGAAATAGCATAT

GTTACATGTACATATCTATAAGATACTAC

CGTAATACAAAATAGTACCAAAT

GTATACATAAATAGACAAGAT

AATACAGATTCAGAGACATATACACAG

ATACAGATTCCACAGAAGATACAGAT

ATACGATCATGCTATACATATCGATC


AGTCGTCGTTCTCTAAATAGGCTAGC
AGCTTCGCTATCTAGCTAGCTAGTACCC
ATGACATATAACAGAAATAGCATAT
GTTACATGTACATATCTATAAGATACTAC
CGTAATACAAAATAGTACCAAA
GTATACATAAATAGACAAGAT
AATACAGATTCAGAGACATATACACAG
ATACAGATTCCACAGAAGATACAGAT
ATACGATCATGCTATACATATCGATC

AGTCGTCGTTCTCTAAATAGGCTAGC

AGCTTCGCTATCTAGCTAGCTAGTACCC

ATGACATATAACAGAAATAGCATAT

GTTACATGTACATATCTATAAGATACTAC

CGTAATACAAAATAGTACCAAA

GTATACATAAATAGACAAGAT

AATACAGATTCAGAGACATATACACAG

ATACAGATTCCACAGAAGATACAGAT

ATACGATCATGCTATACATATCGATC

Approach to Locating Poly(A) Signals

Looking at all alignments, what pattern is the most common in the NUE region

Read and Compress the Data

The FASTA file has the following format:

> Sequence One Title

AGTCTCTGATGACTGATCATGATC

TCAGACGCTACCGTACGACTCTAC

CGTAGATC

> Sequence Two Title

AGCATCATCAGCTTATGACGGCTA

ACGTATTCGATTCAATCTAC

Read and Compress the Data

- Use cb_read_fasta to read the file
- Returns data array with all the sequence information stored with sequences starting on multiples of 4 word boundaries
- Returns pointers to a structure holding information about the start and length of the sequences, and header information

Read and Compress the Data

A = hex:41 = binary:0100 0001

C = hex:43 = binary:0100 0011

G = hex:47 = binary:0100 0111

T = hex:54 = binary:0101 0100

a = hex:61 = binary:0110 0001

c = hex:63 = binary:0110 0011

g = hex:67 = binary:0110 0111

t = hex:74 = binary:0111 0100

Read and Compress the Data

ACGT

0100 0001 0100 0011 0100 0111 0101 0100

ACGT = 00 01 11 10

Use cb_read_fasta and cb_compress

One forth the storage

One forth the I/O to move around

Find the most Common Pattern

ascii	binary	decimal
AAAAAA	000000000000	0
AAAAAC	000000000001	1
AAAAAT	000000000010	2
GCTAGC	110110001101	3469
GGGGGC	111111111101	4093
GGGGGT	111111111110	4094
GGGGGG	111111111111	4095

Find the most Common Pattern

AAAAAA = pat_freq(0) = 0

AAAAAC = pat_freq(1) = 0

: : :

GCTAGC = pat_freq(3469) = 0

: : :

GGGGGT = pat_freq(4094) = 0

GGGGGG = pat_freq(4095) = 0

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=0	AATAGG=0	ATAGGC=0
AGTCGT=0	AGGCTA=0	CTAAAT=0
CGTCGT=0	CGTTCT=0	TAAATA=0
TAGGCT=0	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=0	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=0		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=0	AATAGG=0	ATAGGC=0
AGTCGT=0	AGGCTA=0	CTAAAT=0
CGTCGT=0	CGTTCT=0	TAAATA=0
TAGGCT=0	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=0	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=1		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=0	AATAGG=0	ATAGGC=0
AGTCGT=0	AGGCTA=1	CTAAAT=0
CGTCGT=0	CGTTCT=0	TAAATA=0
TAGGCT=0	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=0	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=1		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=0	AATAGG=0	ATAGGC=0
AGTCGT=0	AGGCTA=1	CTAAAT=0
CGTCGT=0	CGTTCT=0	TAAATA=0
TAGGCT=1	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=0	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=1		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=0	AATAGG=0	ATAGGC=1
AGTCGT=0	AGGCTA=1	CTAAAT=0
CGTCGT=0	CGTTCT=0	TAAATA=0
TAGGCT=1	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=0	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=1		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=0	AATAGG=1	ATAGGC=1
AGTCGT=0	AGGCTA=1	CTAAAT=0
CGTCGT=0	CGTTCT=0	TAAATA=0
TAGGCT=1	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=0	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=1		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=1	AATAGG=1	ATAGGC=1
AGTCGT=0	AGGCTA=1	CTAAAT=0
CGTCGT=0	CGTTCT=0	TAAATA=0
TAGGCT=1	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=0	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=1		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=1	AATAGG=1	ATAGGC=1
AGTCGT=0	AGGCTA=1	CTAAAT=0
CGTCGT=0	CGTTCT=0	TAAATA=1
TAGGCT=1	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=0	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=1		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=1	AATAGG=1	ATAGGC=1
AGTCGT=0	AGGCTA=1	CTAAAT=1
CGTCGT=0	CGTTCT=0	TAAATA=1
TAGGCT=1	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=0	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=1		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCTAGC

AAATAG=1	AATAGG=1	ATAGGC=1
AGTCGT=0	AGGCTA=1	CTAAAT=1
CGTCGT=0	CGTTCT=0	TAAATA=1
TAGGCT=1	TCTAAA=0	TCGTCG=0
TCGTTC=0	TTCTAA=1	GCTAGC=1
GTCGTT=0	GTCGTC=0	GTTCTA=0
GGCTAG=1		

Find the most Common Pattern

AGTCGTCGTTCTCTAAATAGGCT AGC

Once the first sequence's NUE region is searched, move on to the next sequence.

Searching was so fast the region was expanded to 1->50

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATATGCTA

ATGCTA=1	**TATGCT=0**	**ATATGC=0**
TATATG=0	**ATATAT=0**	**TATATA=0**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=0**
TATATG=0	**ATATAT=0**	**TATATA=0**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=0	**ATATAT=0**	**TATATA=0**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=1	**ATATAT=0**	**TATATA=0**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=1	**ATATAT=1**	**TATATA=0**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATA<u>TATATA</u>TGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=1	**ATATAT=1**	**TATATA=1**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=1	**ATATAT=2**	**TATATA=1**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=1	**ATATAT=2**	**TATATA=2**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=1	**ATATAT=3**	**TATATA=2**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=1	**ATATAT=3**	**TATATA=3**
CTATAT=0	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=1	**ATATAT=3**	**TATATA=3**
CTATAT=1	**TCTATA=0**	**GTCTAT=0**

Troubles with Multiple Matches

TGCAGTCAGTCTATATATATATATGCTA

ATGCTA=1	**TATGCT=1**	**ATATGC=1**
TATATG=1	**ATATAT=3**	**TATATA=3**
CTATAT=1	**TCTATA=1**	**GTCTAT=0**

Troubles with Multiple Matches

Biased towards repeating pattern

Added code to count patterns only once per sequence.

Results – 6 character length

Multi-count		Single-count	
TTTTTT	3394	AATAAA	2021
AATAAA	2242	TTTGTT	2021
TTTGTT	2224	TTTTGT	1993
TTTTGT	2122	TTGTTT	1859
TTGTTT	2025	TGTTTT	1806
AAAAAA	2006	ATTTTT	1795
ATATAT	1945	TTTCTT	1720
TGTTTT	1937	TTTTAT	1695
ATTTTT	1897	TATTTT	1662
TTTCTT	1858	TTTTTT	1629

Results – 7 character length

Multi-count		Single-count	
TTTTTTT	1717	TTTGTTT	972
TTTGTTT	1043	AATAAAA	945
AATAAAA	988	TTTTGTT	940
AAAAAAA	982	TTTTTTT	819
TTTTGTT	980	TTGTTTT	796
TATATAT	951	TTTTCTT	767
ATATATA	895	TTTCTTT	758
AAATAAA	835	AAATAAA	757
TTGTTTT	832	TTTTTGT	724
TTTCTTT	803	TTTATTT	716

Results – 8 character length

Multi-count		Single-count	
TTTTTTTT	898	TTTTGTTT	436
ATATATAT	492	TTTGTTTT	429
TATATATA	479	TTTTTTTT	406
AAAAAAAA	475	TTTTTCTT	357
TTTTGTTT	459	AAATAAAA	347
TTTGTTTT	447	TTTTCTTT	341
AAATAAAA	363	TTTTTGTT	337
TTTTTCTT	361	ATTTTTTT	326
TTTTCTTT	349	TTTCTTTT	322
TTTTTGTT	338	TAATAAAA	319

Results – 9 character length

Multi-count		Single-count	
TTTTTTTTT	499	TTTTTTTTT	234
TATATATAT	290	TTTTGTTTT	194
ATATATATA	274	TATATATAT	177
AAAAAAAAA	229	TTTGTTTTT	168
TTTTGTTTT	206	TTTTTCTTT	162
TTTGTTTTT	169	ATATATATA	161
TTTTTCTTT	164	TTTTCTTTT	155
ATAAATAAA	158	ATTTTTTTT	154
TTTTCTTTT	158	TTTTTGTTT	154
TTTTTGTTT	155	ATAAATAAA	146

Results – 10 character length

Multi-count		Single-count	
TTTTTTTTTT	273	TTTTTTTTTT	123
ATATATATAT	178	ATATATATAT	93
TATATATATA	171	TATATATATA	90
AAAAAAAAAA	90	ATTTTTTTTT	82
AATAAATAAA	83	TTTTTTTTTG	81
ATTTTTTTTT	82	TTTTGTTTTT	79
TTTTTTTTTG	81	TTTTTTTCTT	78
TTTTGTTTTT	79	AATAAATAAA	76
TTTTTTTCTT	78	GTTTTTTTTT	74
TTTTTCTTTT	74	TTTTTCTTTT	73

Results – 2D Matrix

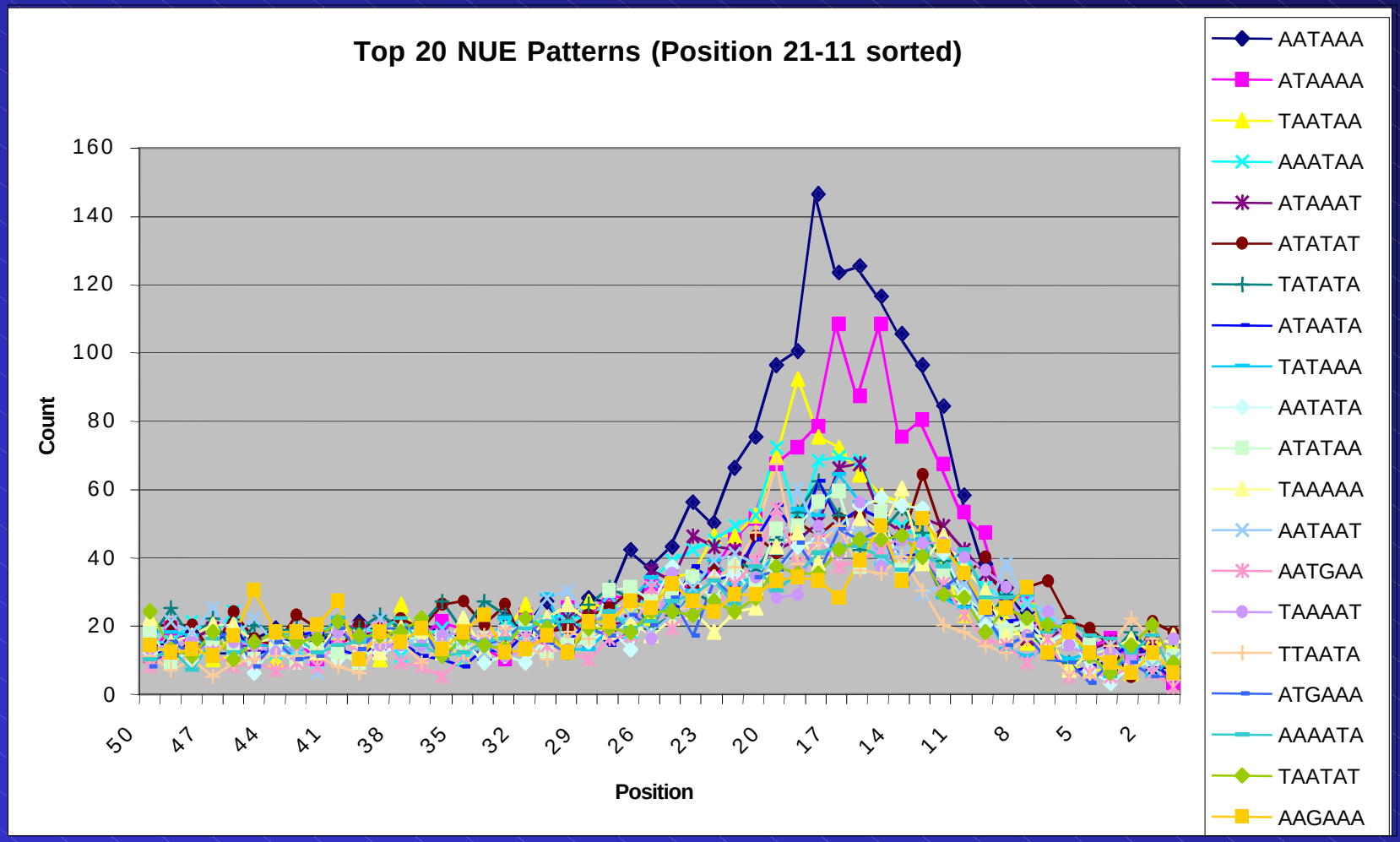
Needed more information about the locations

Expanded the pat_freq array into a 2D matrix

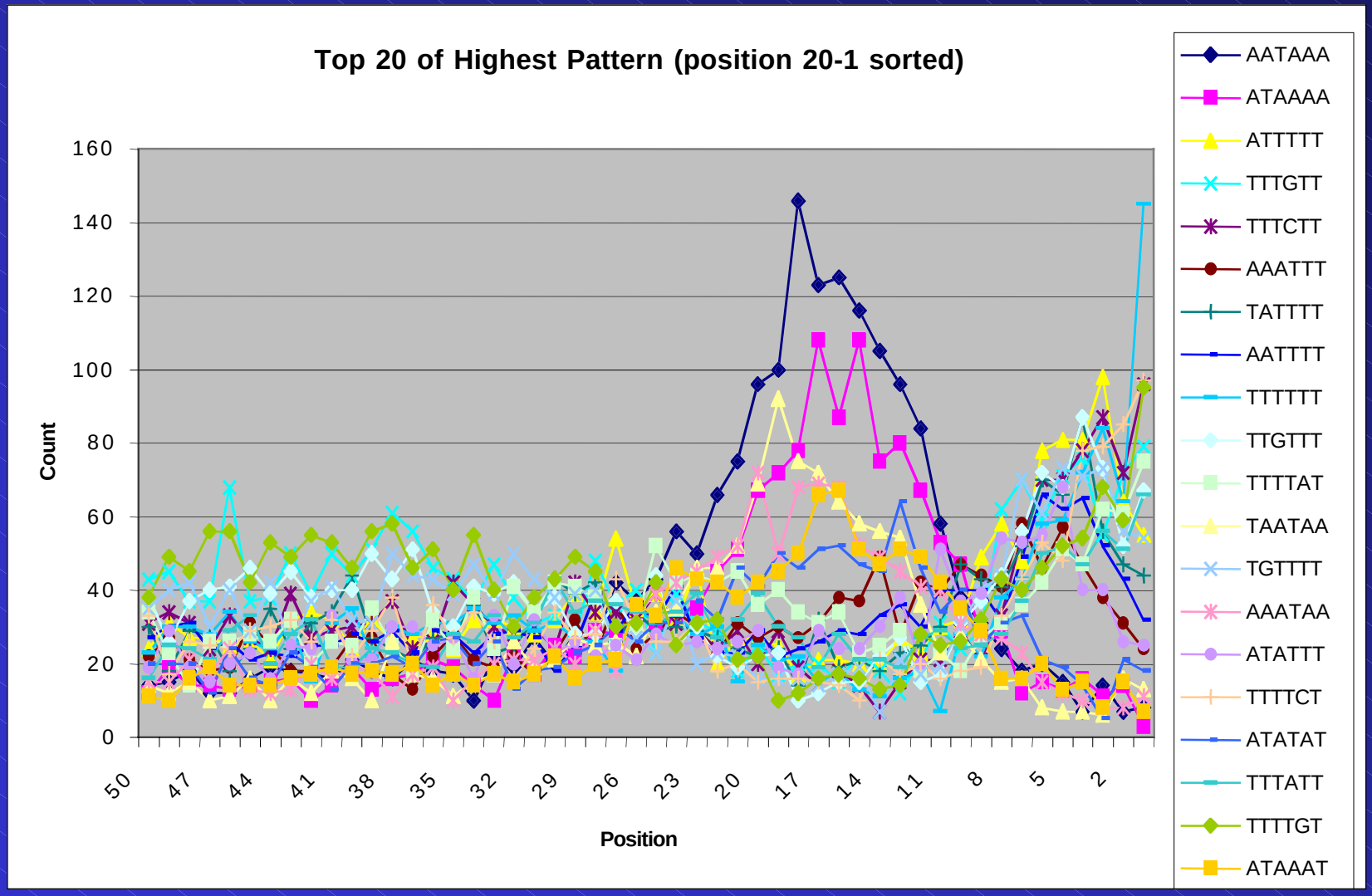
Now we can count the most common pattern on a location basis

Graphs show frequency by location

Summary Findings



Summary Findings



Conclusions

- Search revealed the NUE patterns that are in the expected range of location
- Discovered unknown consensus of patterns near the cleavage site

Next step

- Expand the search to the FUE region
- Tie the NUE and FUE searches together
- Package the program so it more general purpose for others to use
- Install at OSC

Acknowledgements

- Dr. Brain Haas (TIGR) for providing raw databases
- Cray Inc, for SV1ex computer time
- OSC for SV1 computer time and support