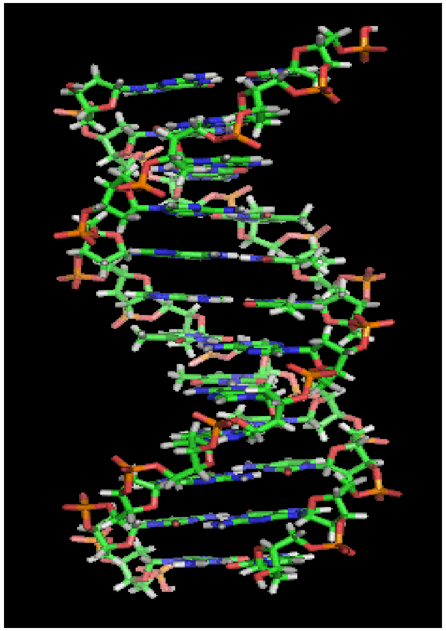
Genome Databases
(Chapter 10)

Peter Revesz

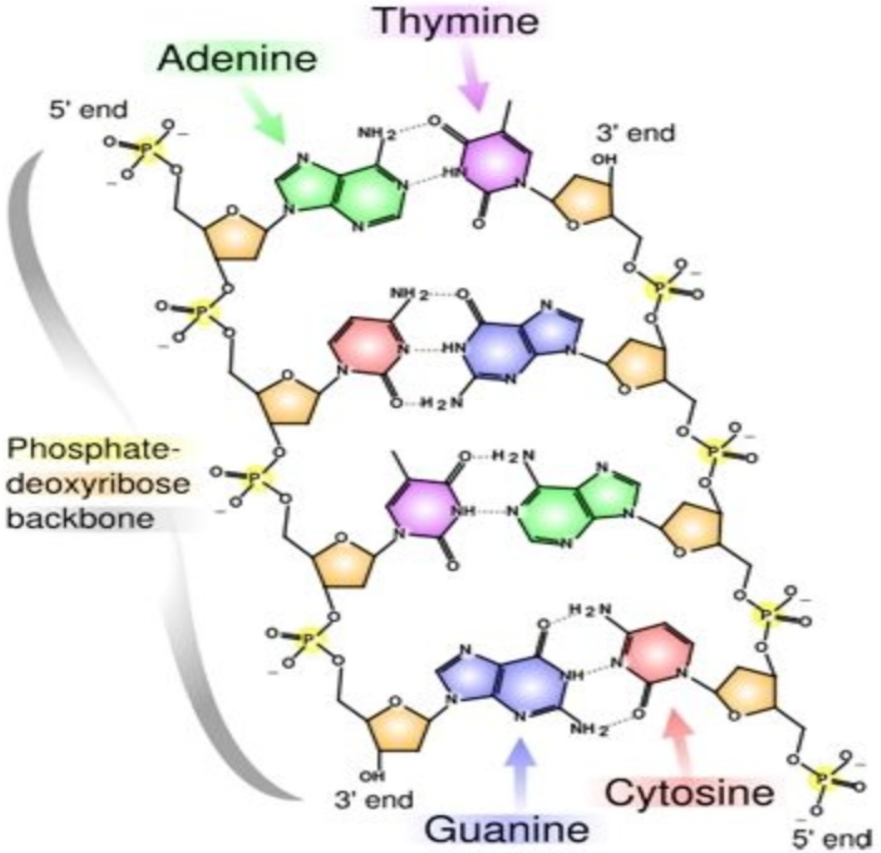
CSCE 413/813
Computer Science and Engineering
University of Nebraska – Lincoln

Biological Basics
(Chapter 10.1)

Deoxyribonucleic Acid (DNA)



Double helix structure
[Watson and Crick 1953]



Essential building blocks are four nucleotides (A, C, G and T). A pairs with T and C pairs with G.

Only the sense nucleotide string, which encodes proteins and other useful information, is enough to represent in a database. Example:

```
CATCGATCTCGGGAGGGATCCATTATCGATTCCCGGCTC
GGGGGATCCTTCCATCGATGGGCCCGAGGCGGATCCCTAC
TATCGATCCCGGGGGGATCCTTAATTCTCGAGAAGGCCTA
TCGATCAAGGATCCTATCGATCCCGAGTCCCGGGAT
```

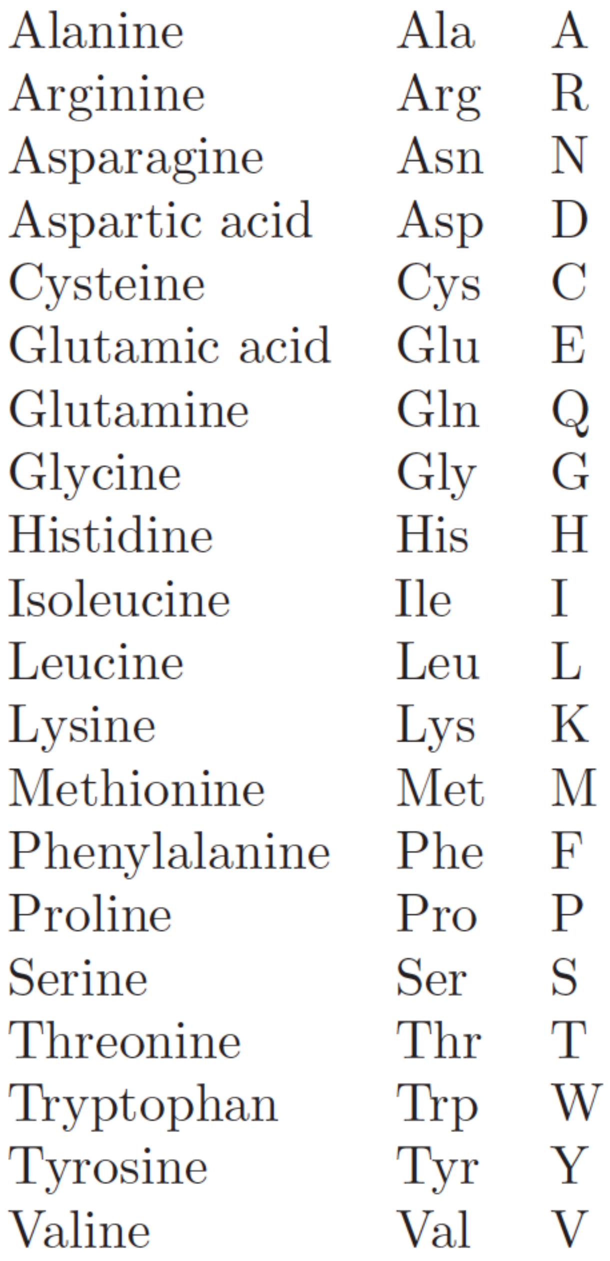
Genome Data Abstraction

View Level: Visualizations of proteins and DNA strings.

Logical Level: Relational database scheme extended with string data type.

Physical Level: The way data is actually stored in a computer.

Amino Acids and Proteins

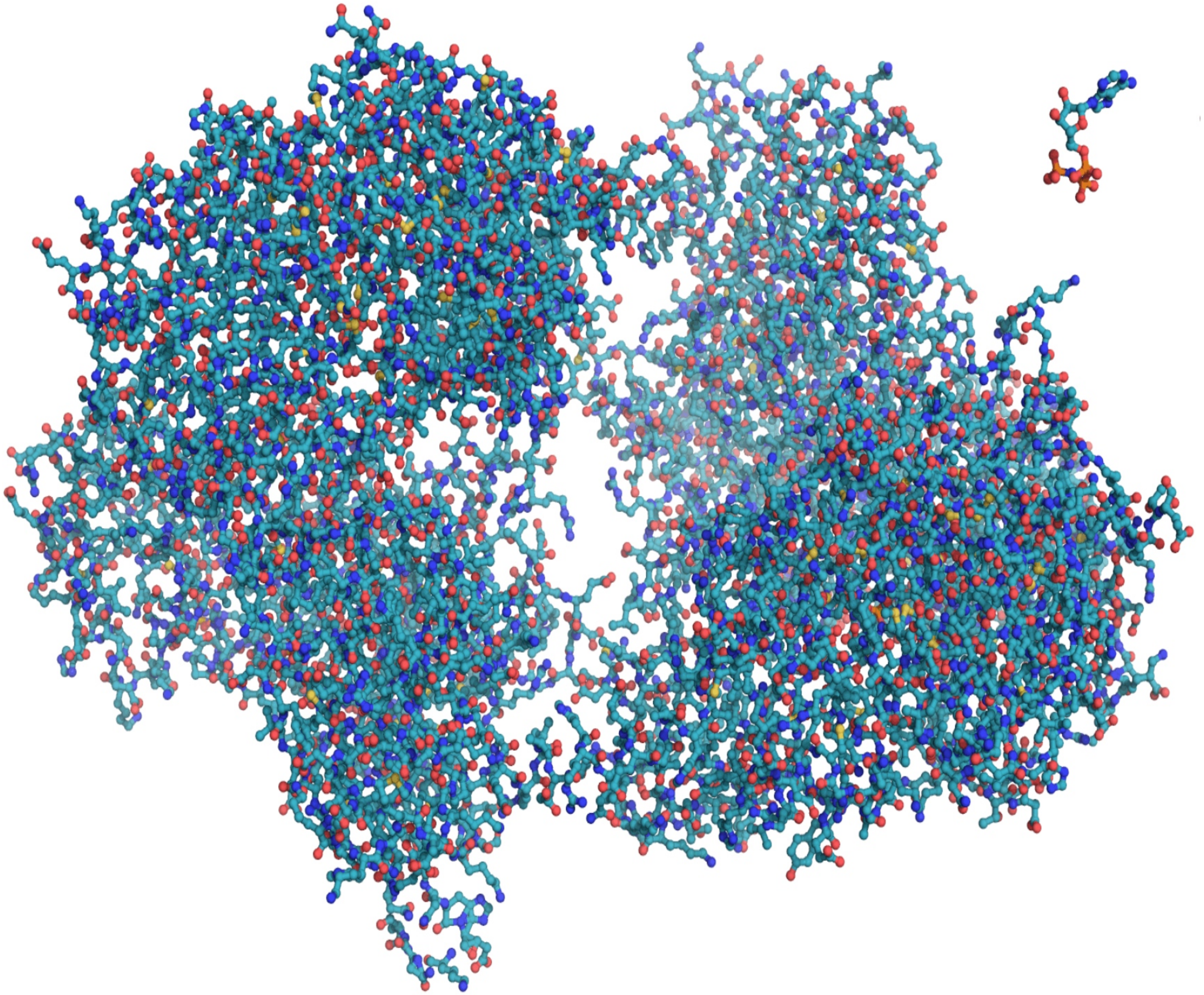


Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamic acid	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Proteins are chains of amino acids.

The amino acid chains fold into 3D structures that largely determine the biological properties of the proteins.

Hexokinase protein and Adenosine Triphosphate (ATP) [Wikipedia]



Function: Hexokinase phosphorylates six-carbon sugars and helps produce ATP, an energy transfer molecule that is essential for life.

Genes that encode hexokinase are found in every domain of life.

Standard Genetic Code

Translation from DNA to Protein

TTT	F	Phe	TCT	S	Ser	TAT	Y	Tyr	TGT	C	Cys
TTC	F	Phe	TCC	S	Ser	TAC	Y	Tyr	TGC	C	Cys
TTA	L	Leu	TCA	S	Ser	TAA	*	Ter	TGA	*	Ter
TTG	L	Leu	TCG	S	Ser	TAG	*	Ter	TGG	W	Trp
CTT	L	Leu	CCT	P	Pro	CAT	H	His	CGT	R	Arg
CTC	L	Leu	CCC	P	Pro	CAC	H	His	CGC	R	Arg
CTA	L	Leu	CCA	P	Pro	CAA	Q	Gln	CGA	R	Arg
CTG	L	Leu	CCG	P	Pro	CAG	Q	Gln	CGG	R	Arg
ATT	I	Ile	ACT	T	Thr	AAT	N	Asn	AGT	S	Ser
ATC	I	Ile	ACC	T	Thr	AAC	N	Asn	AGC	S	Ser
ATA	I	Ile	ACA	T	Thr	AAA	K	Lys	AGA	R	Arg
ATG	M	Met	ACG	T	Thr	AAG	K	Lys	AGG	R	Arg
GTT	V	Val	GCT	A	Ala	GAT	D	Asp	GGT	G	Gly
GTC	V	Val	GCC	A	Ala	GAC	D	Asp	GGC	G	Gly
GTA	V	Val	GCA	A	Ala	GAA	E	Glu	GGA	G	Gly
GTG	V	Val	GCG	A	Ala	GAG	E	Glu	GGG	G	Gly

Practice

Translate from DNA string to Amino Acid Sequence

DNA string:

CATCGATCTCGGGAGGGATCCATTATCGATTCCCGGGCTC
GGGGGATCCTTCCATCGATGGGCCCGAGGCGGATCCCTAC
TATCGATCCCGGGGGGATCCTTAATTCTCGAGAAGGCCTA
TCGATCAAGGATCCTATCGATCCCGAGTCCCGGGAT

Amino acid sequence:

HRSREGSIIDSRARGILPSMGPRRIPTIDPGGILNSREELSIKDPIDPESRD

Nitrogen Fixation

Nitrogen fixation: Conversion of atmospheric nitrogen to ammonium or other biologically useful form.

Evidence of enzymes for nitrogen fixation (nitrogenase) were recently found in rocks that are over 3 billion years old.

Diazotrophe: Bacteria and archaea that can fix atmospheric nitrogen.

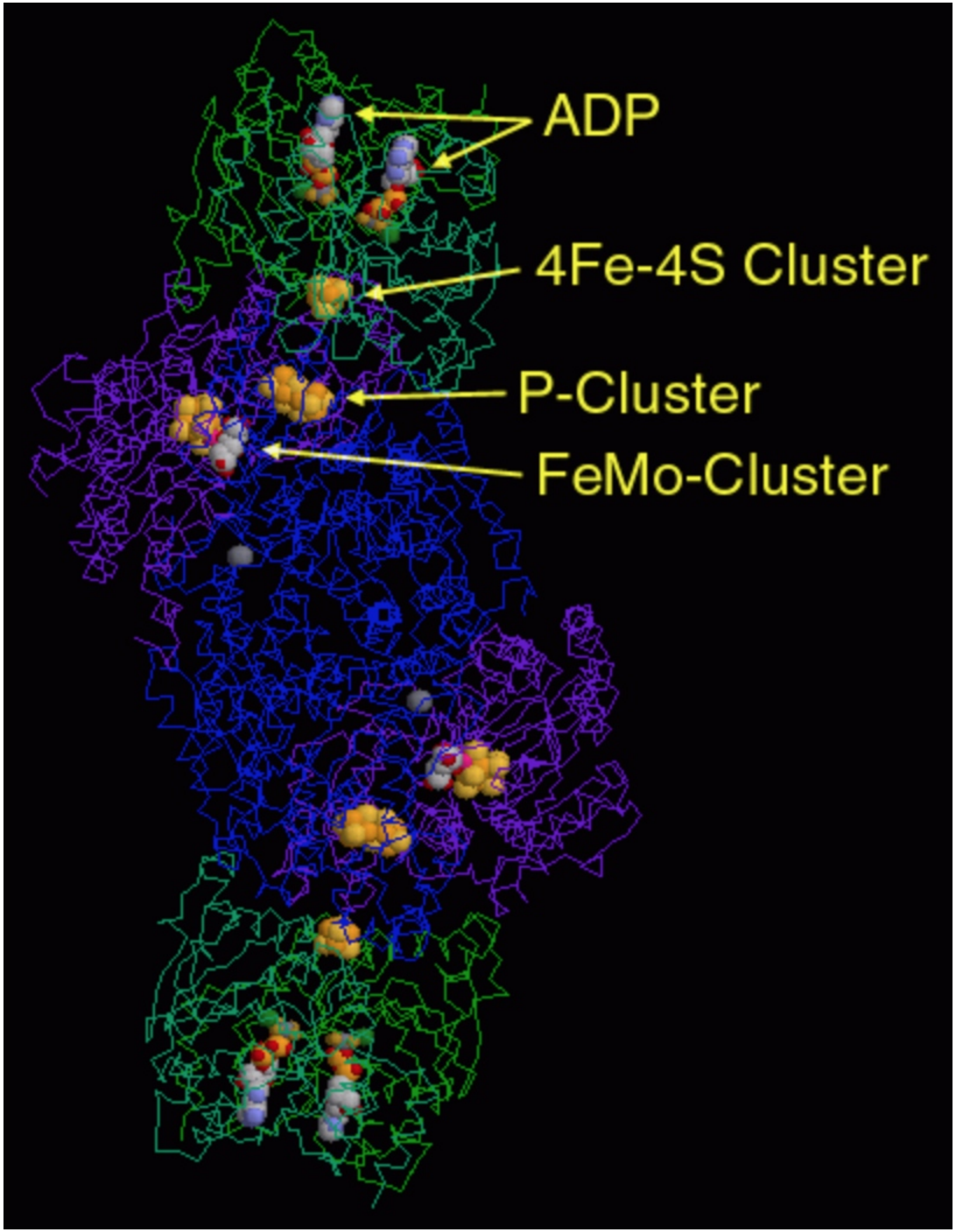
Organism: *Staphylococcus epidermidis (strain ATCC 12228)*

Protein Name: Nitrogen fixation protein NifU

Uniprot Database ID: SE_0630
Protein Database ID: Q8CPV7

Amino Acid Sequence: MPTENPTMFD QVAEVIERLR PFLLRDGGDC
TLVDVEDGIV KLQLHGACGT CPSSTITLKA GIERALHEEV PGVIEVEQVF

Nitrogenase Example [from PDB]



The sites that bind ADP, nitrogen and other molecules are highlighted. These binding sites or active sites are essential for biological function.

The individual amino acids of a protein can be replaced by other amino acids without disturbing the biological function as long as the structure of the binding sites remain unchanged.

The replaceability allows harmless (i.e., biological function preserving) variations by genetic mutation of the DNA that encodes the nitrogenese genes.

Nitgorenase enzymes in various organisms may have a common evolutionary origin (over 3 billion years ago).

The Genome Map Assembly Problem

(Chapter 10.2)

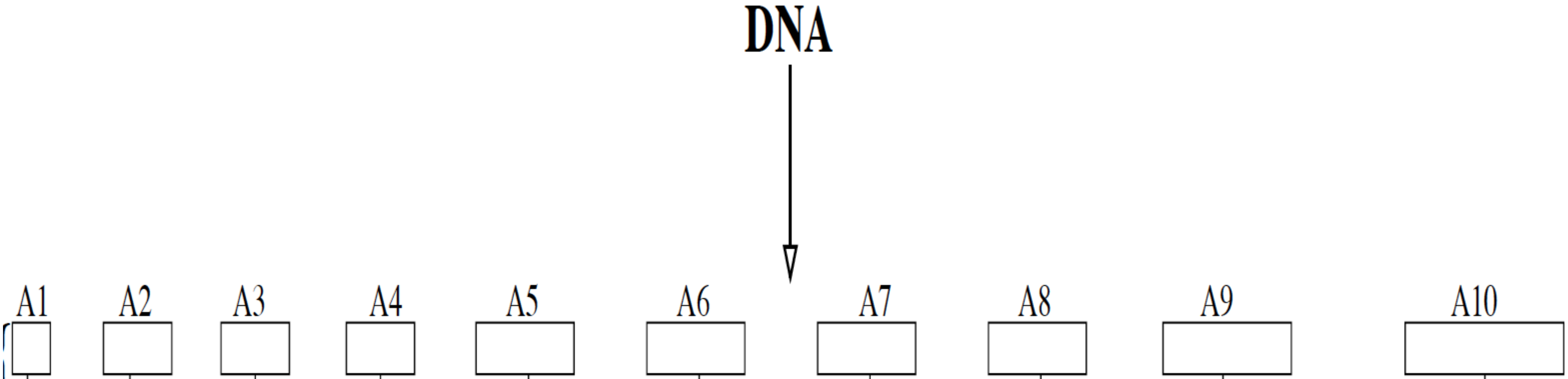
Restriction Enzymes

Restriction enzymes cut a genome always at particular sites. For example, a restriction enzyme *a* may cut the genome always exactly at locations between three Cs followed by three Ts. We indicate this as follows:

....CCC | TTT....

Suppose that we apply the restriction enzyme *a* to some DNA string that has exactly nine occurrences of the above pattern. Then we may get the ten fragments as shown below.

The fragments float in a solution and their order is lost. We can quickly determine their lengths in terms of number of nucleotides. Using fragment length information to reconstruct the original order is called the genome map assembly problem.

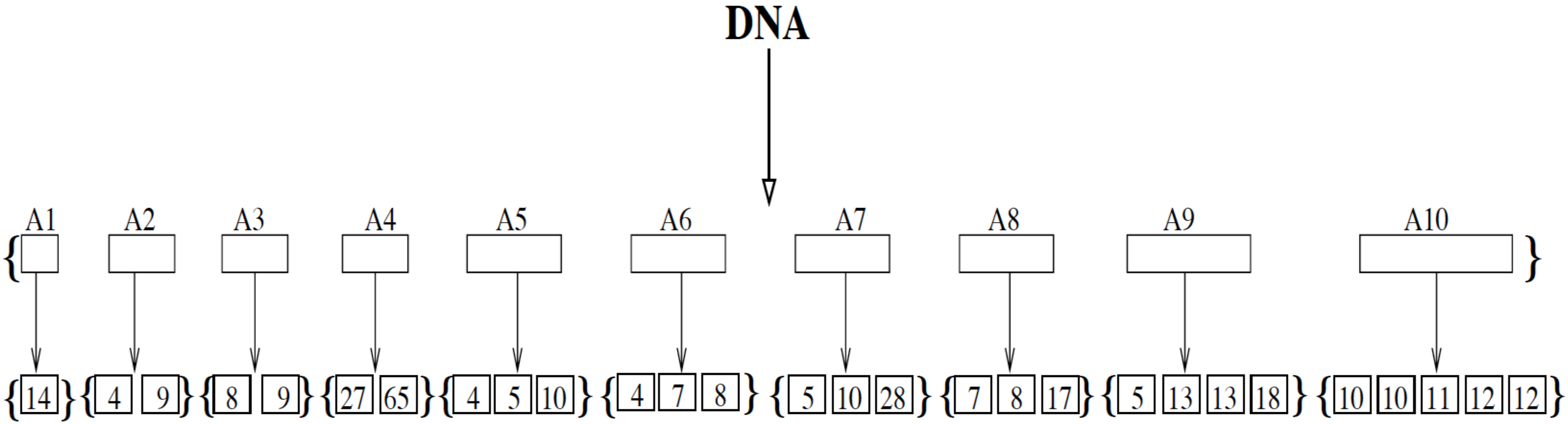


Fragment Length Data

Cutting by enzyme *a,* we obtained the subsequences A1, …, A10. That is not enough information to do a genome map assembly. However, we can isolate these subsequences and cut them again using another restriction enzyme *b* that always cuts between three GGGs and three AAAs.

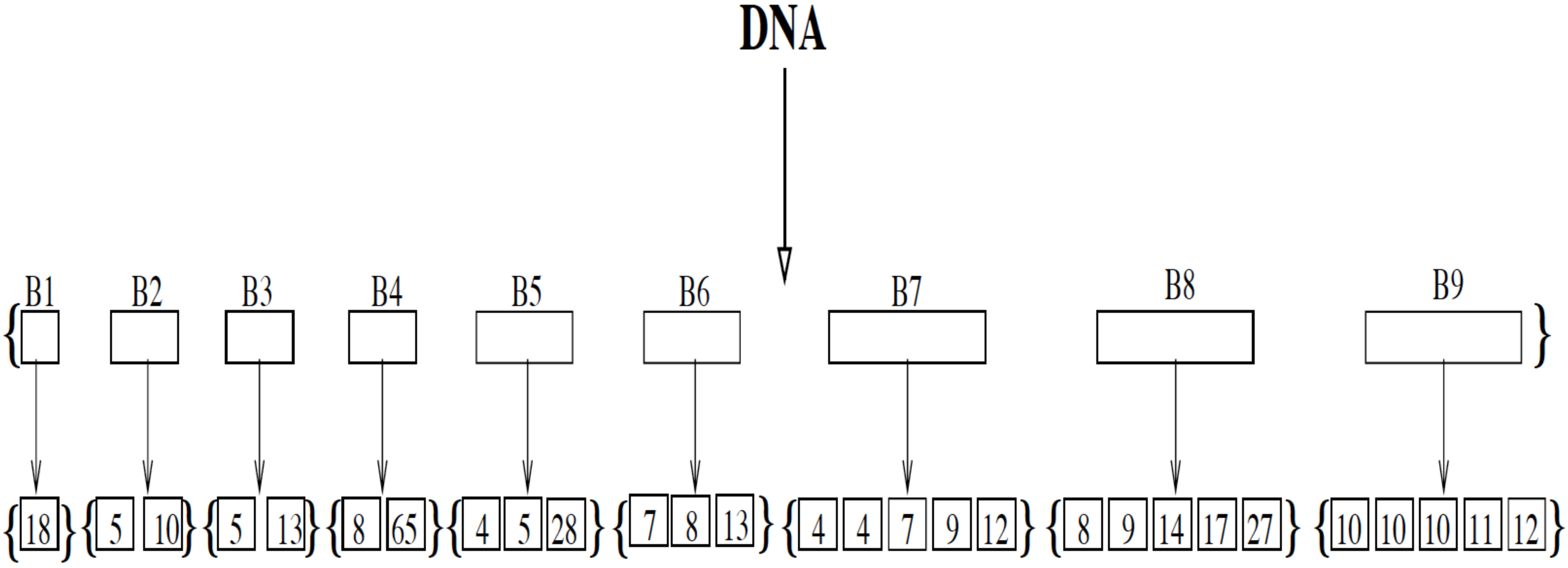
....GGG | AAA....

We can also apply a third enzyme *c* that cuts between two Cs and two Gs. After cutting by *b* and *c* we can measure the lengths of the various fragments. Suppose we obtain the following measures:

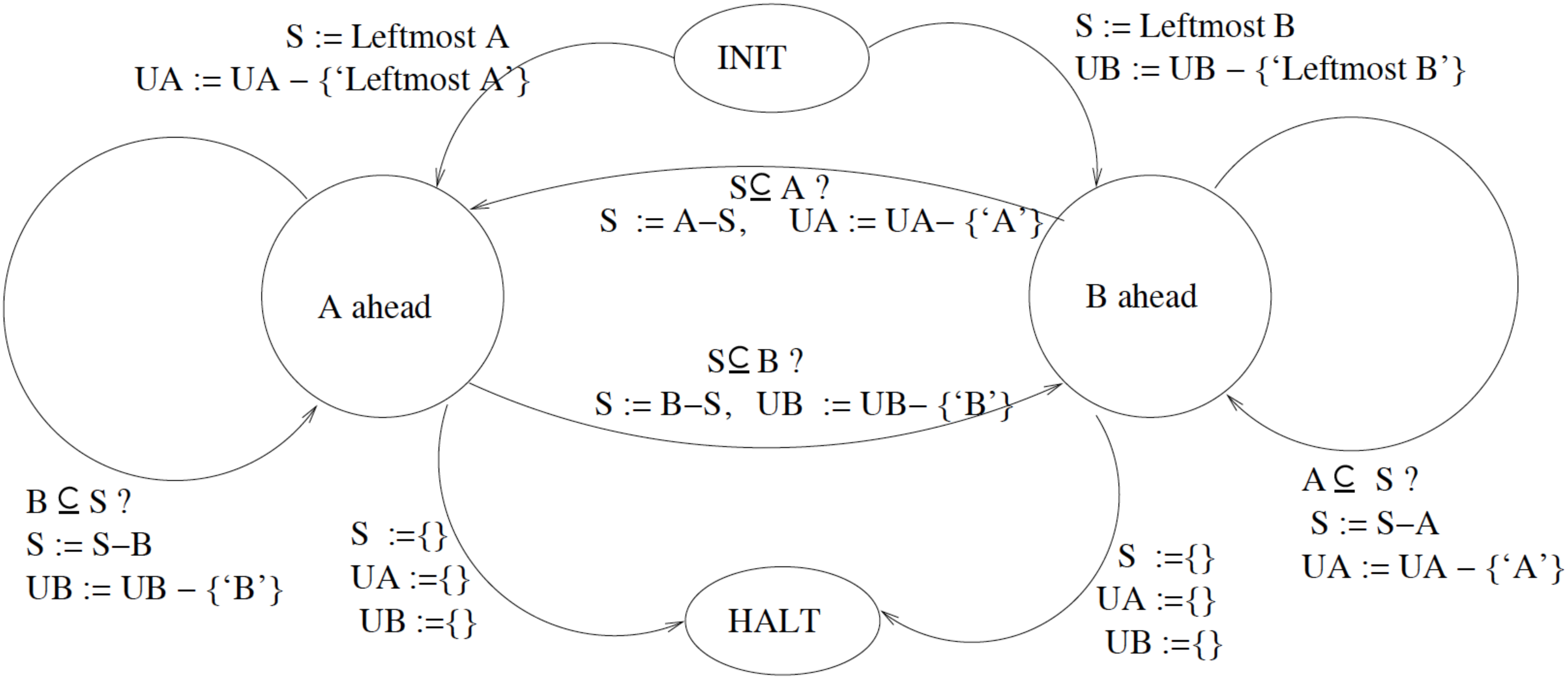


Fragment Length Data

We still do not have enough information for genome map assembly. However, we can take another copy of the DNA string and now cut it up by first applying restriction enzyme *b*, isolate the subsequences B1, …, B9, and then applying restriction enzymes *a* and *c* and then measure the fragments. Suppose we obtain the following. Note that the length values are the same in the bottom rows but in different groupings and order. Within any subsequence, the order again can be changed because the fragments simply float in a solution before being measured.



Genome Map Assembly Automaton



Definition: A bag is like a set where repetitions can occur. A big bag is a bag of bags.

In the above automaton:

UA = {A1, A2, A3, A4, A5, A6, A7, A8, A9, A10}

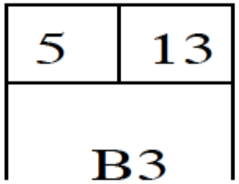
UB = {B1, B2, B3, B4, B5, B6, B7, B8, B9}

A one element of UA

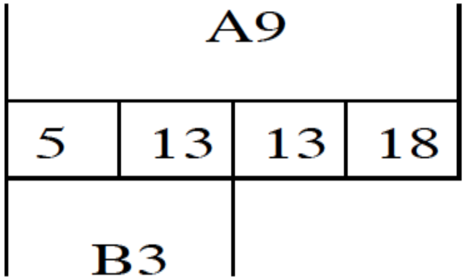
B one element of UB

S current set of elements considered

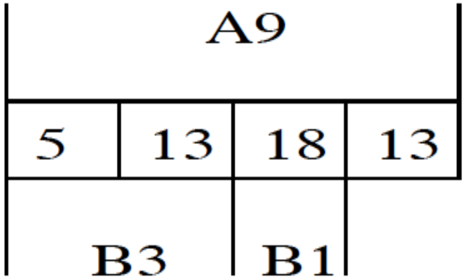
Genome Map Assembly Example



The process of genome map assembly is almost like matching two rows of dominos with A bags on the top and B bags on the bottom.

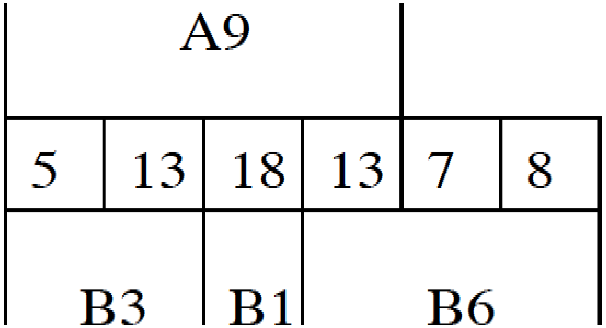


Suppose we start with the B3 on the bottom, which implies that we first move to the “B ahead” state. In that state we look for an A that can help us to catch up with B. In this case, A9 can be put on the top because it matches all the elements of B3. Note that S = A9 \ B3 = {13, 18} will be the set of values by which the A row on the top will be ahead of the B row on the bottom. Hence we move to the “A ahead” state.

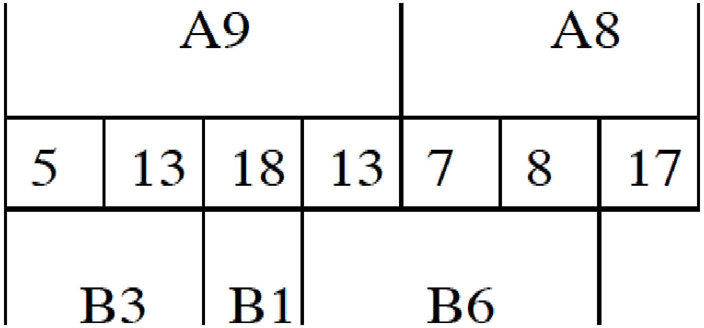


Now we try to find among the Bs something that is able to match the values of S. In this case we can find that B1 matches the element 18 in S. Therefore we place it after B3 in the bottom row and update S to be S = S \ B1 = {13}. We still remain in the “A ahead” state.

Genome Map Assembly Example



Now we again try to find among the Bs something that is able to match the values of S. In this case we can find that B6 matches the element 13 in S. Therefore we place it after B1 in the bottom row and update S to be S = B6 \ S = {7, 8}. This is the set of values by which the B row is now ahead. Hence we move to the “B ahead” state.



Now we look for some A bag and find that A8 matches both 7 and 8 within S.

….

We continue this process until the A and the B rows match completely and all the As and Bs are used exactly once. If we don't succeed, then we need to backtrack and try another piece.

Genome Map Assembly Example

At the end we can arrive to the following solution:

<table>
<tr><td colspan="4">A9</td><td colspan="3">A8</td><td colspan="2">A3</td><td colspan="1">A1</td><td colspan="2">A4</td><td colspan="3">A6</td><td colspan="2">A2</td><td colspan="5">A10</td><td colspan="3">A7</td><td colspan="3">A5</td></tr>
<tr><td>5</td><td>13</td><td>18</td><td>13</td><td>7</td><td>8</td><td>17</td><td>8</td><td>9</td><td>14</td><td>27</td><td>65</td><td>8</td><td>7</td><td>4</td><td>9</td><td>4</td><td>12</td><td>11</td><td>12</td><td>10</td><td>10</td><td>10</td><td>28</td><td>5</td><td>4</td><td>5</td><td>10</td></tr>
<tr><td colspan="2">B3</td><td colspan="1">B1</td><td colspan="3">B6</td><td colspan="5">B8</td><td colspan="2">B4</td><td colspan="5">B7</td><td colspan="5">B9</td><td colspan="3">B5</td><td colspan="2">B2</td></tr>
</table>