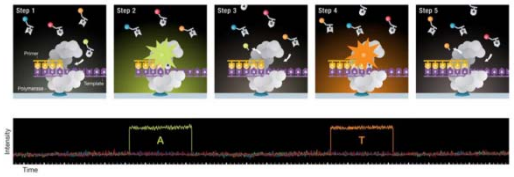
Next Generation Sequences & Chloroplast Assembly

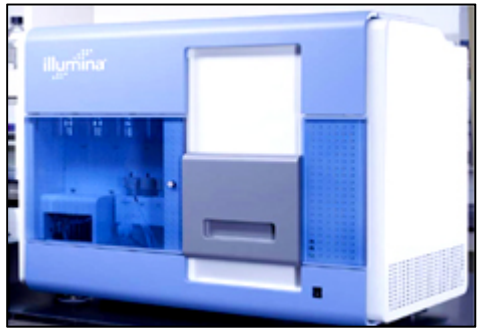
Next Generation Sequencing (NGS) Technologies

GS-Titanium; Roche 454



SMRT; Pacific Bioscience

SOLiD; ABI



Solexa; Illumina

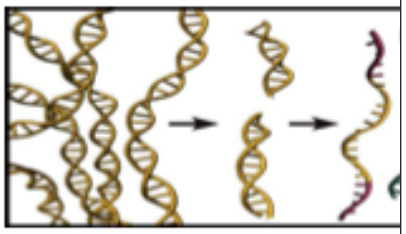
Helicos; Helicos Bioscience

- Next (or Current) generation sequencing technologies have accelerated the speed of genome sequencing projects and have broaden application range of genome sequences.

NGS: 454 Technology

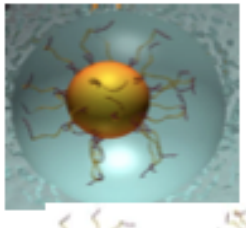
DNA Isolation

DNA Fragmentation



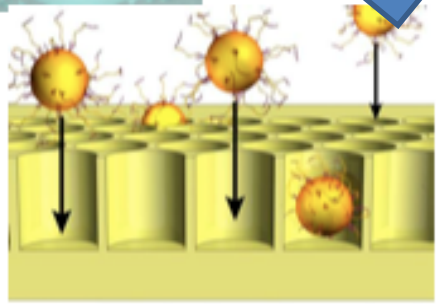
1) Prepare Adapter Ligated ssDN

DNA Amplification



2) Clonal
on 28 µm

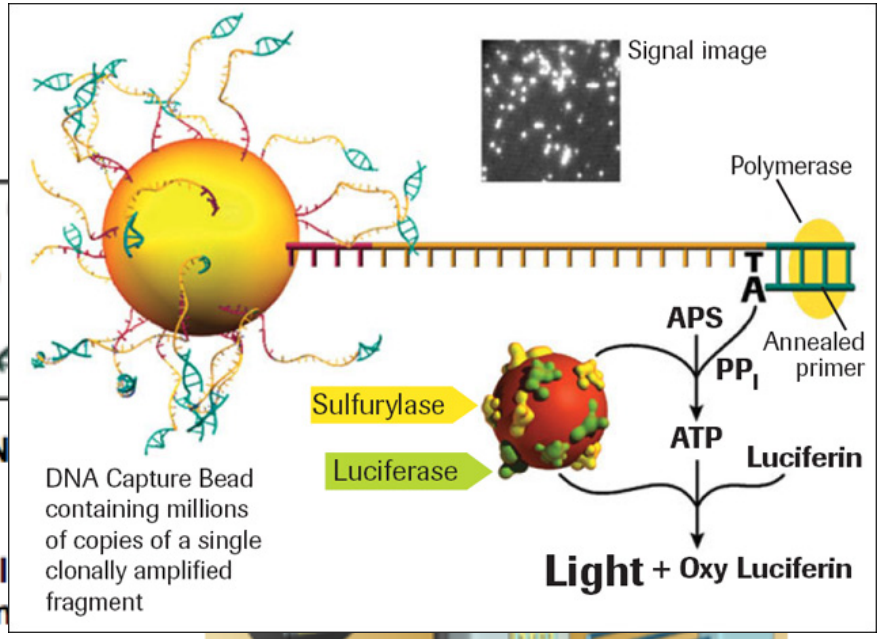
DNA Sequencing

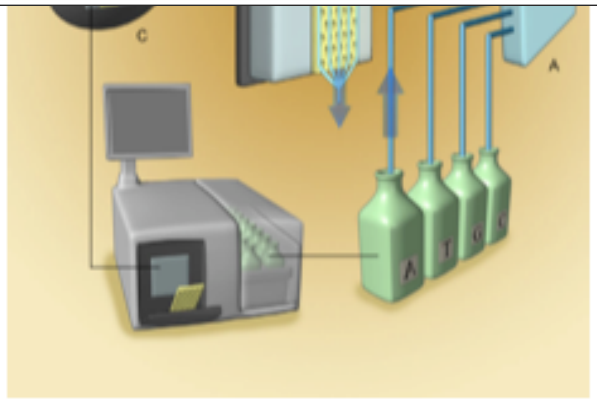


3) Load beads and enzymes in PicoTiter Plate™

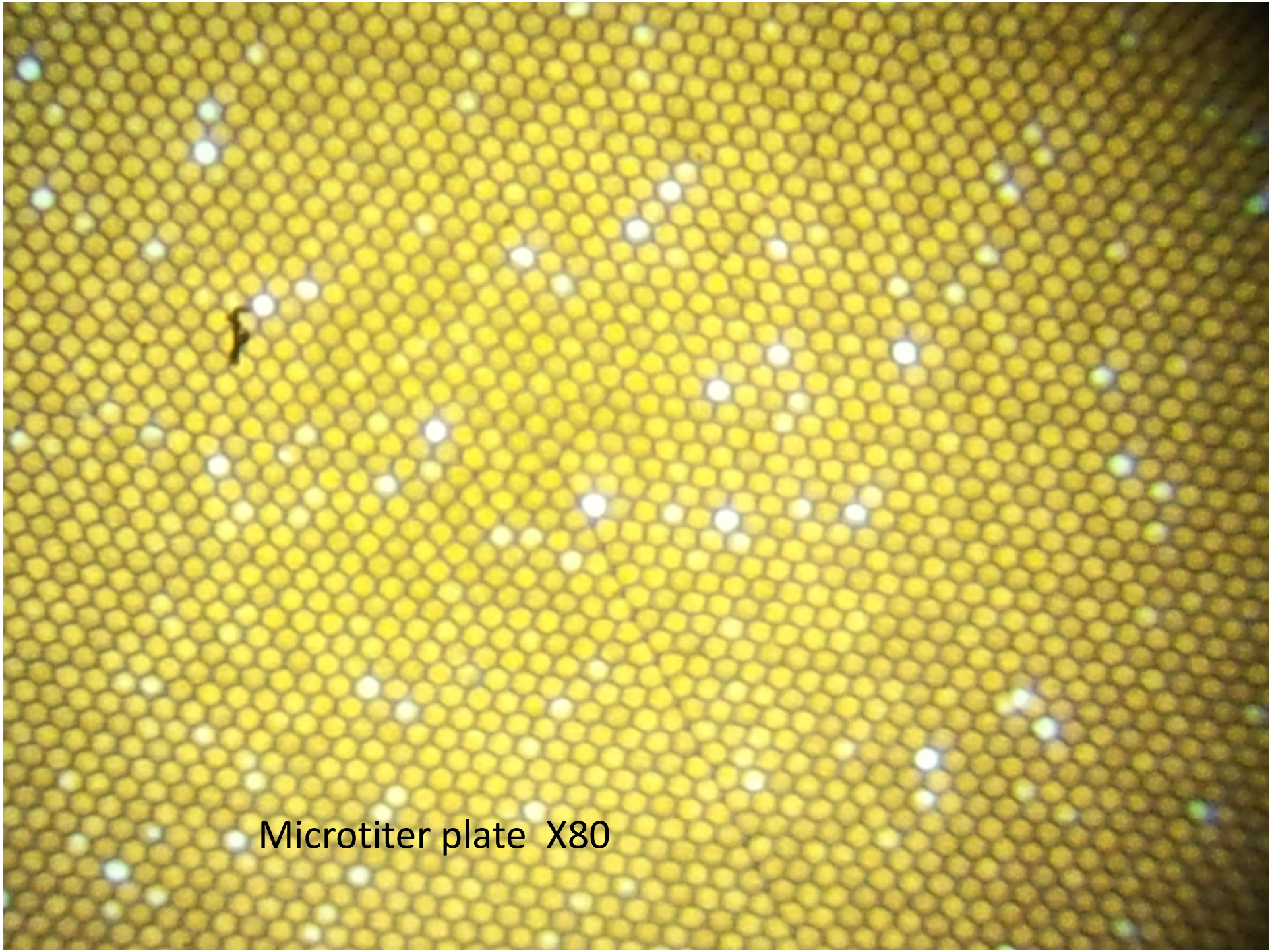
Bioinformatics

- Contigs are constructed

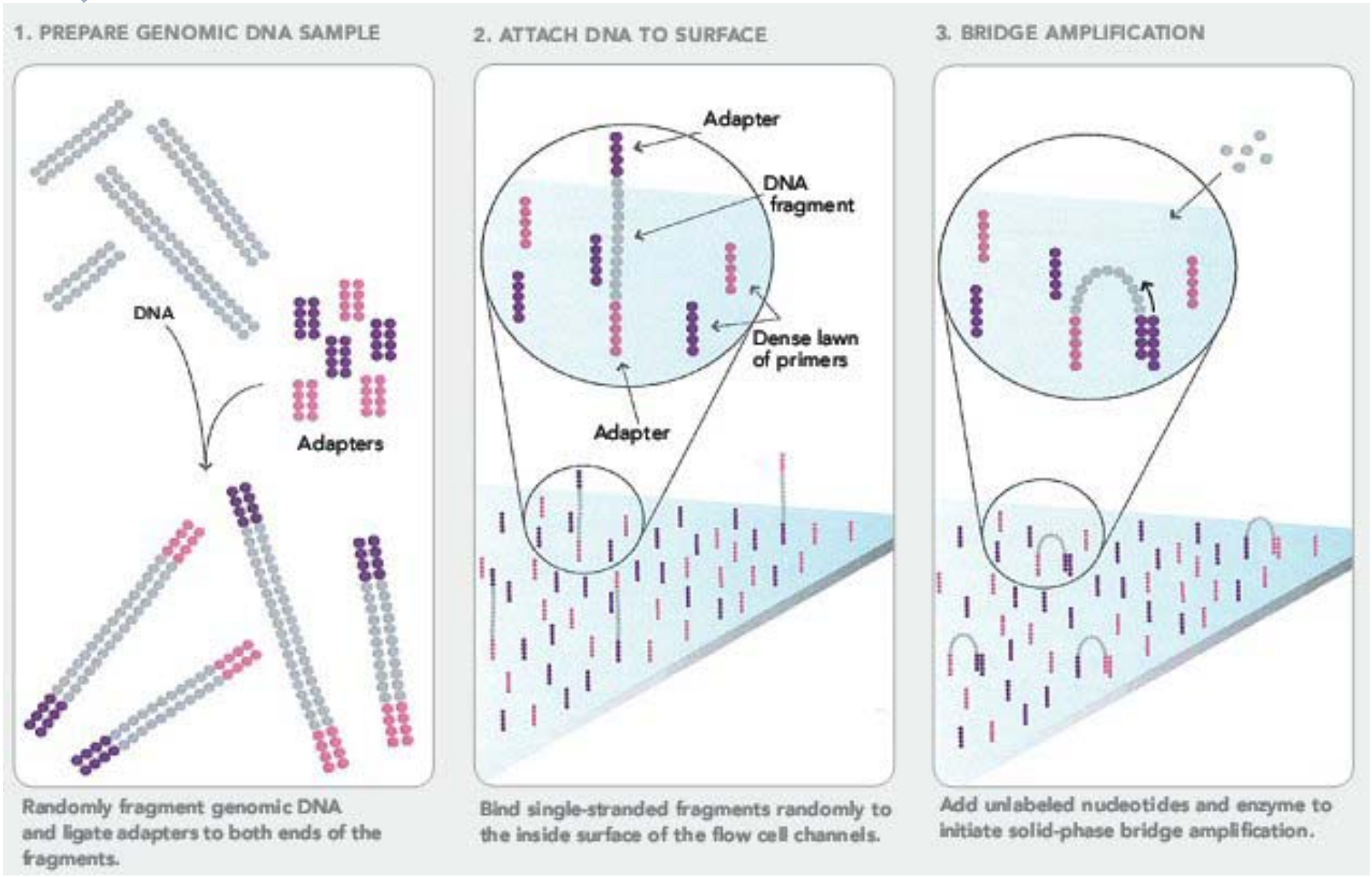




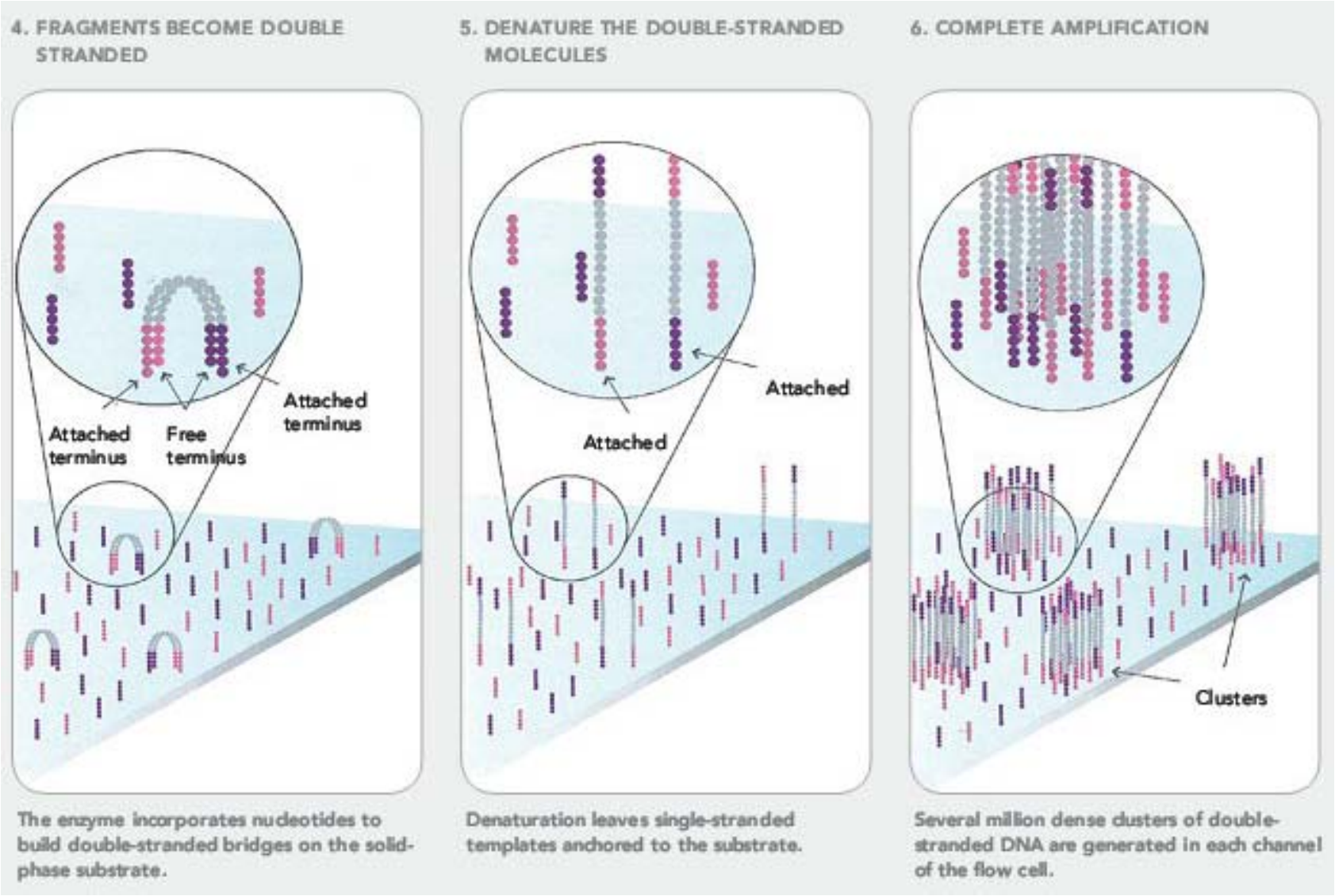
4) Perform Sequencing by synthesis on the 454 Instrument


Microtiter plate X80

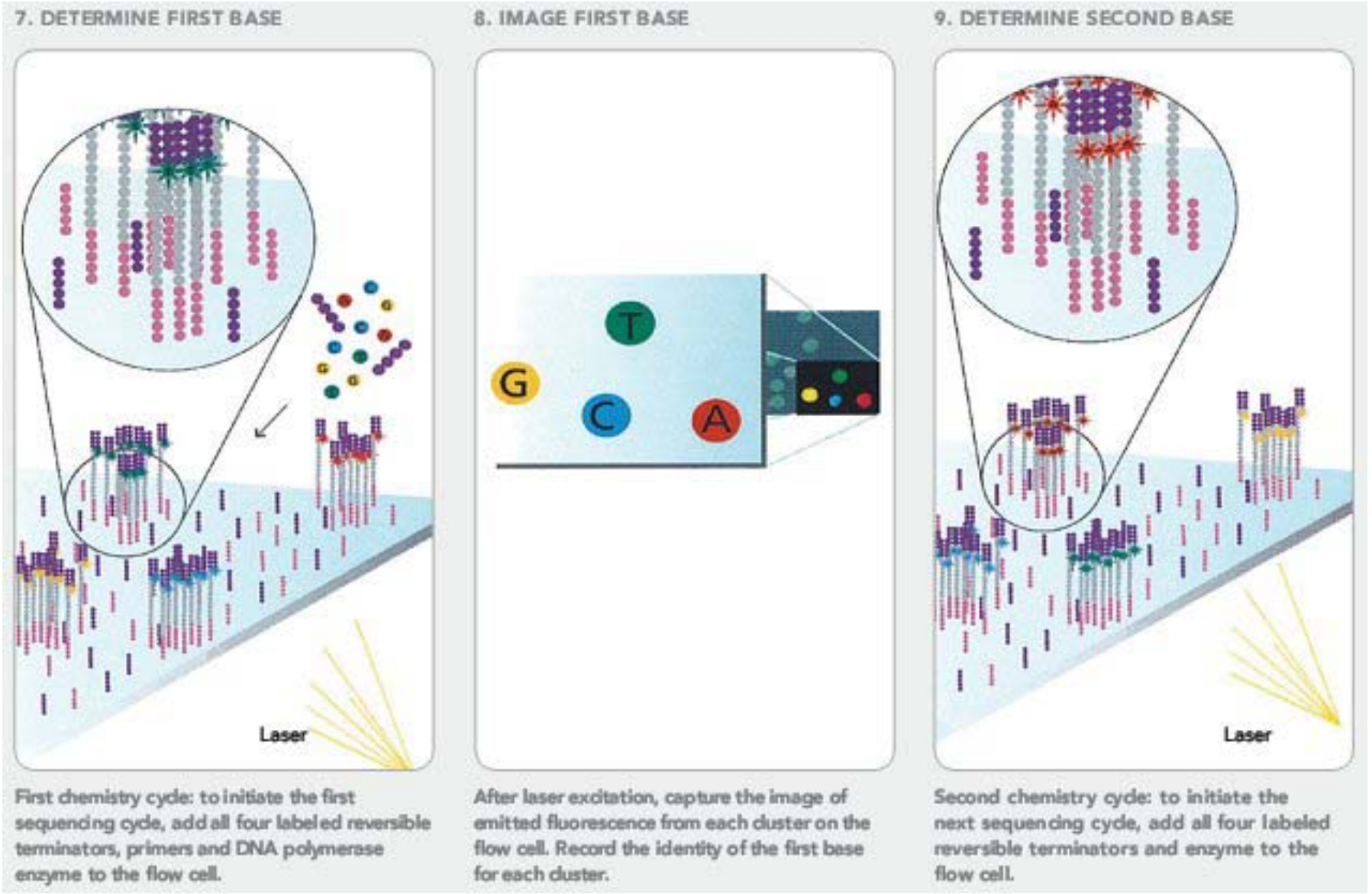
NGS: Solexa Technology



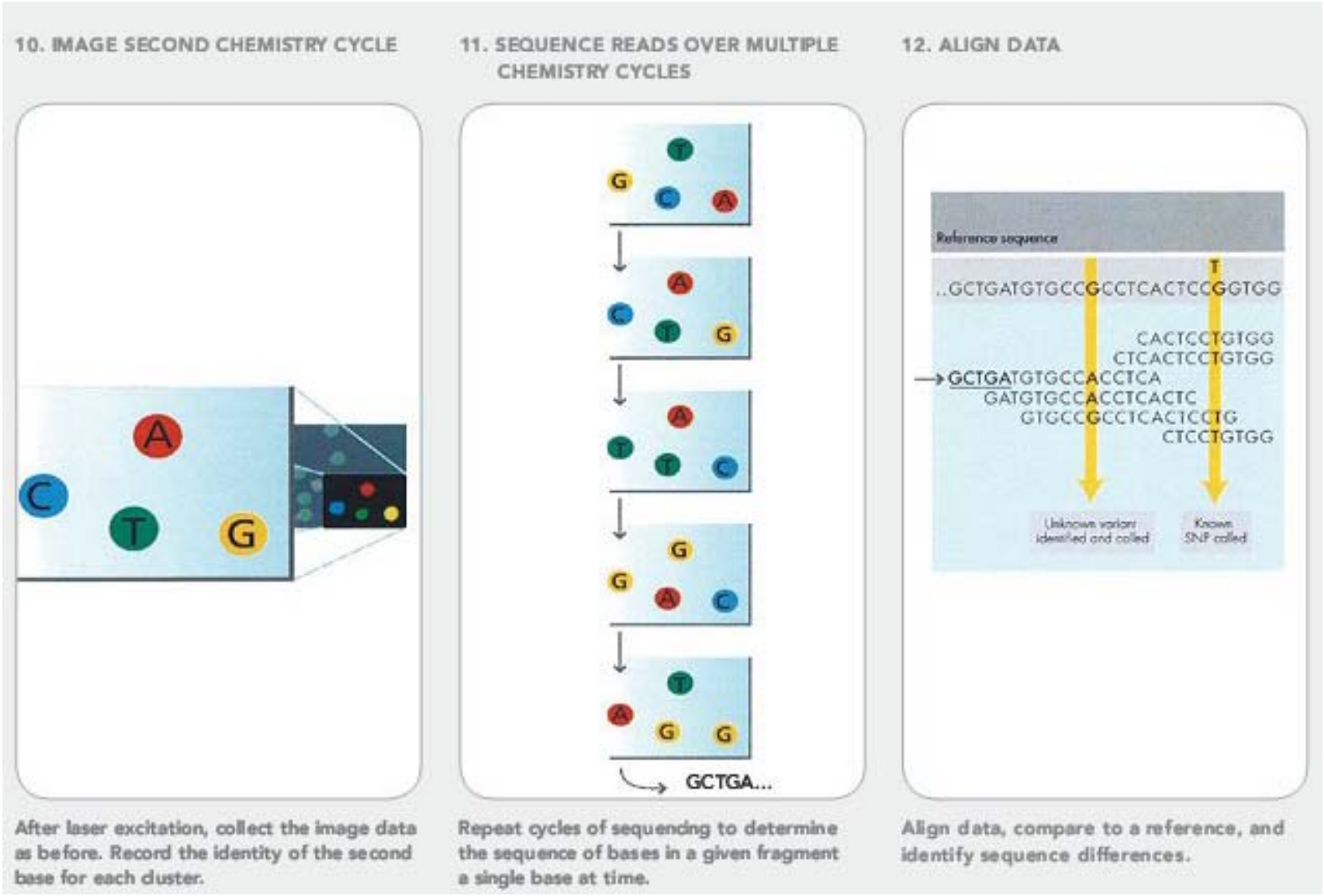
NGS: Solexa Technology



NGS: Solexa Technology



NGS: Solexa Technology



After laser excitation, collect the image data as before. Record the identity of the second base for each cluster.

Repeat cycles of sequencing to determine the sequence of bases in a given fragment a single base at time.

Align data, compare to a reference, and identify sequence differences.

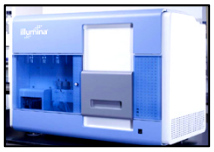
Capacities of Next Generation Sequencers

384 x 700 bp = 268,800 bp = **269Kb** (per one reaction / 1 hr)

ABI 3730; ABI

950,000 x 450 bp = 405,000,000 bp = **405Mb** (per one reaction / 2-3 days)

GS-Titanium; Roche 454



35,000,000 x 7 x (151 x 2) bp = 73,990,000,000 bp = **74.0 Gb** (per one reaction / 12 days)

Solexa GA2; Illumina

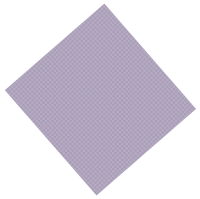
70,000,000 x 14 x (101 x 2) bp = 197,960,000,000 bp = **198.0 Gb** (per one reaction / 8 days)

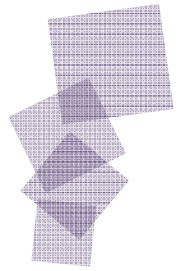
HiSeq2000; Illumina

1,400,000,000 x 75 bp (50+25) = 105,500,000,000 bp = **105.5Gb** (per one reaction / 11 days)

SOLiD 4; ABI

Genome Assembly Processes With NGS Sequences

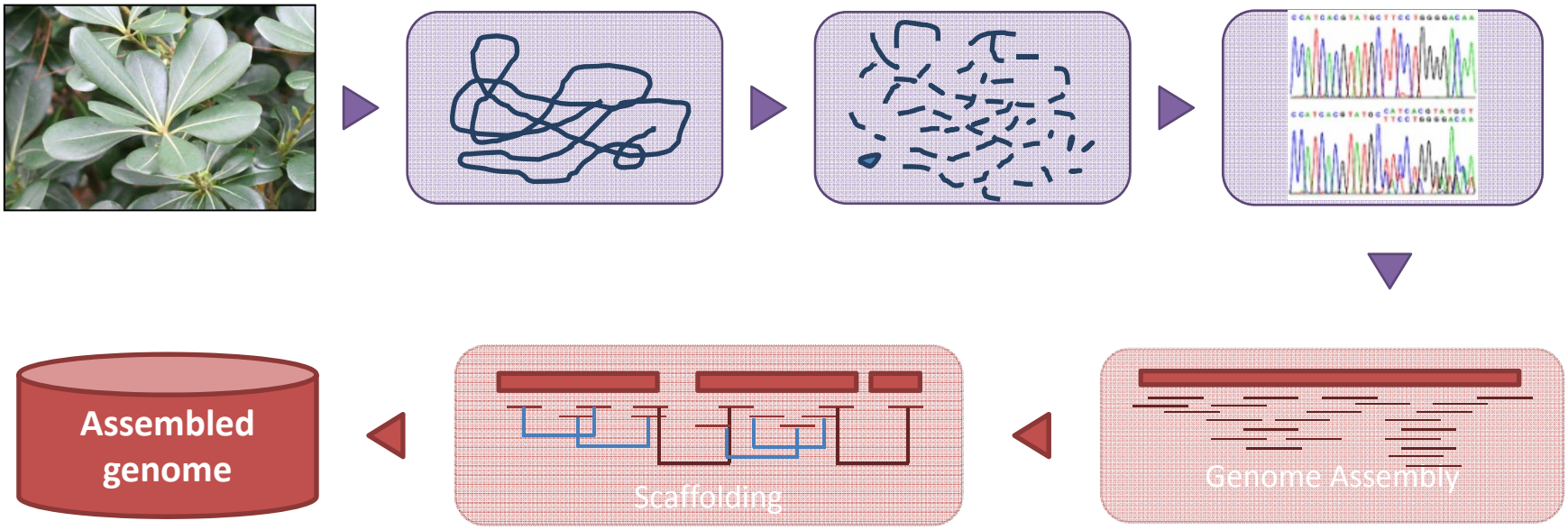




Whole Shotgun Sequence Strategy

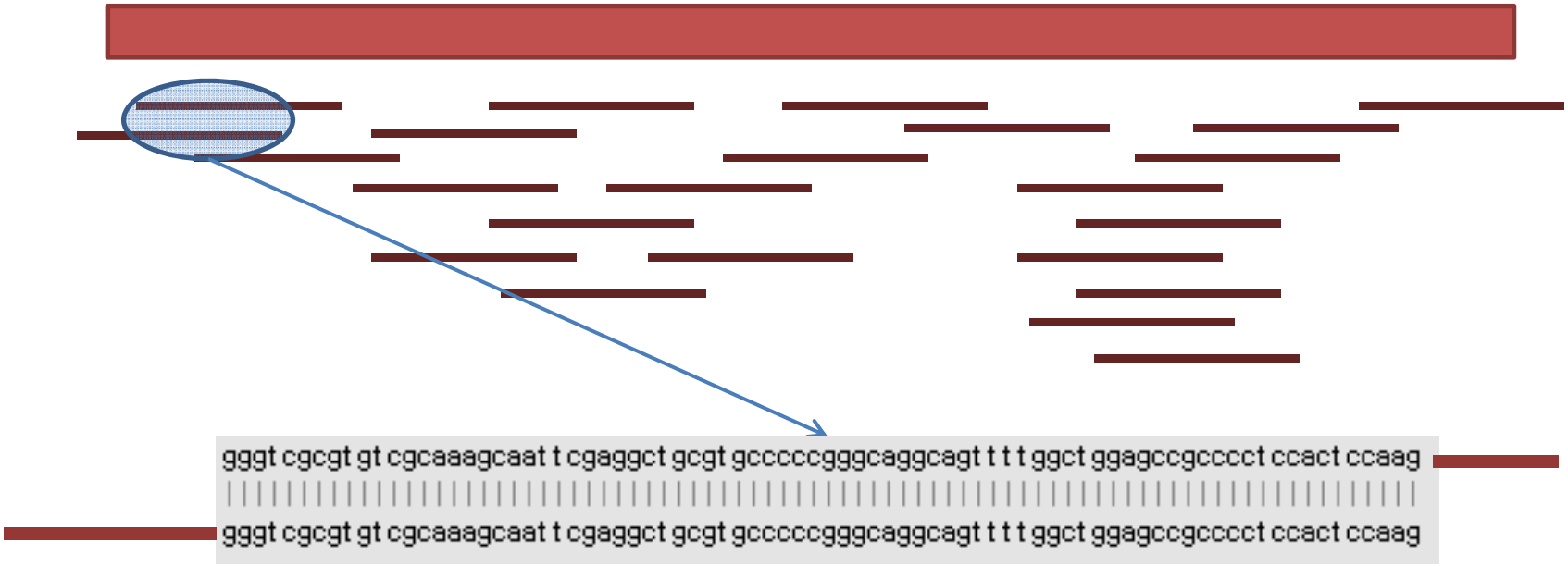
- Assembly process is essential for genome project because read length of each sequence is less than 1 kb.

- Assembly process was conducted by several popular programs, such as phrap and PCAP3, for Sanger sequences.



Genome Assembly Process

- We can perform genome assembly manually!



23 sequences should be **compared** with each other!

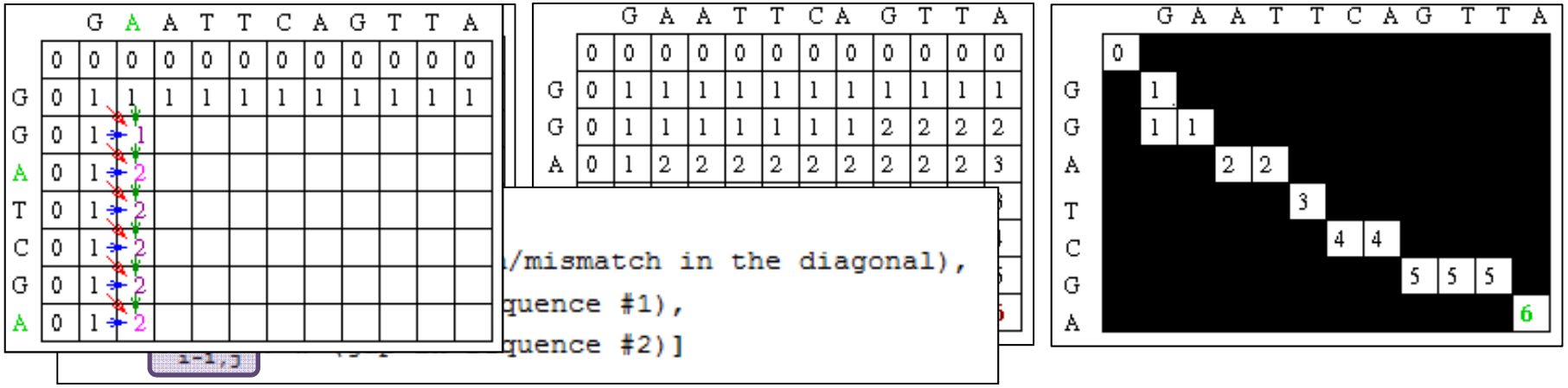
$_{23}C_2$ = 23*22 / 2 = 253 comparison!

How to Find Overlapped Sequences?

- Using **dynamic algorithm**, we can make the program for finding similar sequences.

- Complexity of this algorithm is $O(n^3)$.

G A A T T C A G T T A (sequence #1)
G G A T C G A (sequence #2)



G _ A A T T C A G T T A
| | | | | |
G G _ A _ T C _ G _ _ A

http://www.avatar.se/molbioinfo2001/dynprog/dynamic.html

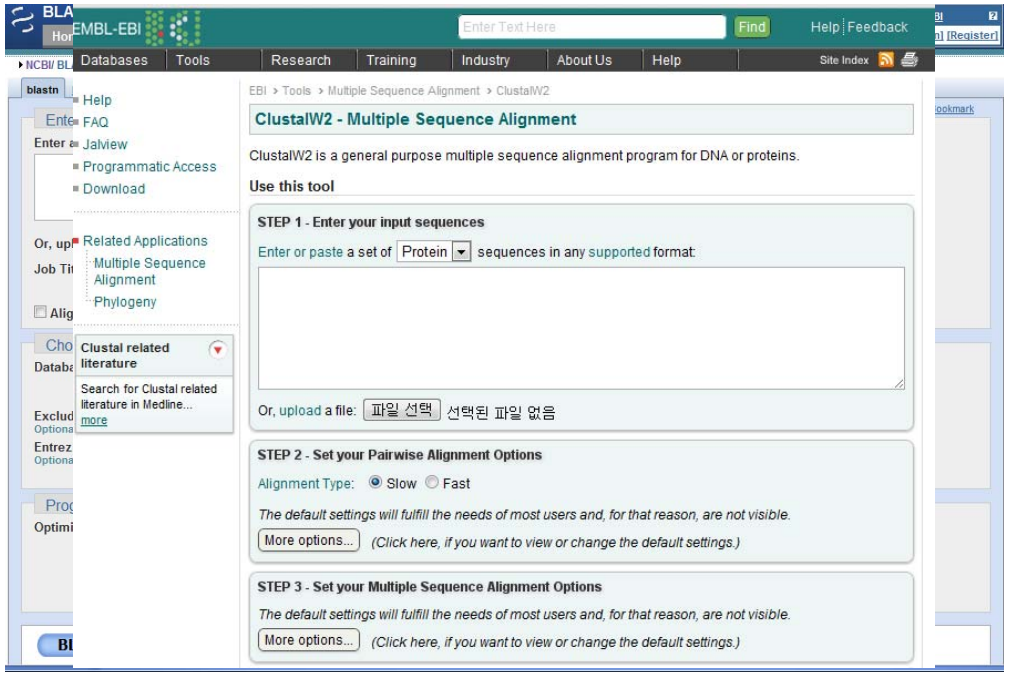
Famous Bioinformatics Tools for Alignments

- Global alignment : ClustalW, T-coffee, and MUSCLE

- Local alignment : FASTA, and BLAST (Basic Local Alignment Searching Tools) provided by NCBI.

- Pair-wise alignment : BLAST and FASTA

- Multiple sequence alignment : ClustalW, T-coffee, MUSCLE, and etc.



Example of Genome Assembly: *Vitis Vinifera*

Vol 449|27 September 2007|doi:10.1038/nature06148 nature

LETTERS

The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla

The French–Italian Public Consortium for Grapevine Genome Characterization*

A total of 6.2 million end-reads were produced by our consortium, representing an 8.4-fold coverage of the genome. Within th
bly, performed with Arachne[12], 316 supercontigs represent
allelic haplotypes that constitute 11.6 million bases (Mb

Pair-wise comparison of 6,200,000 reads

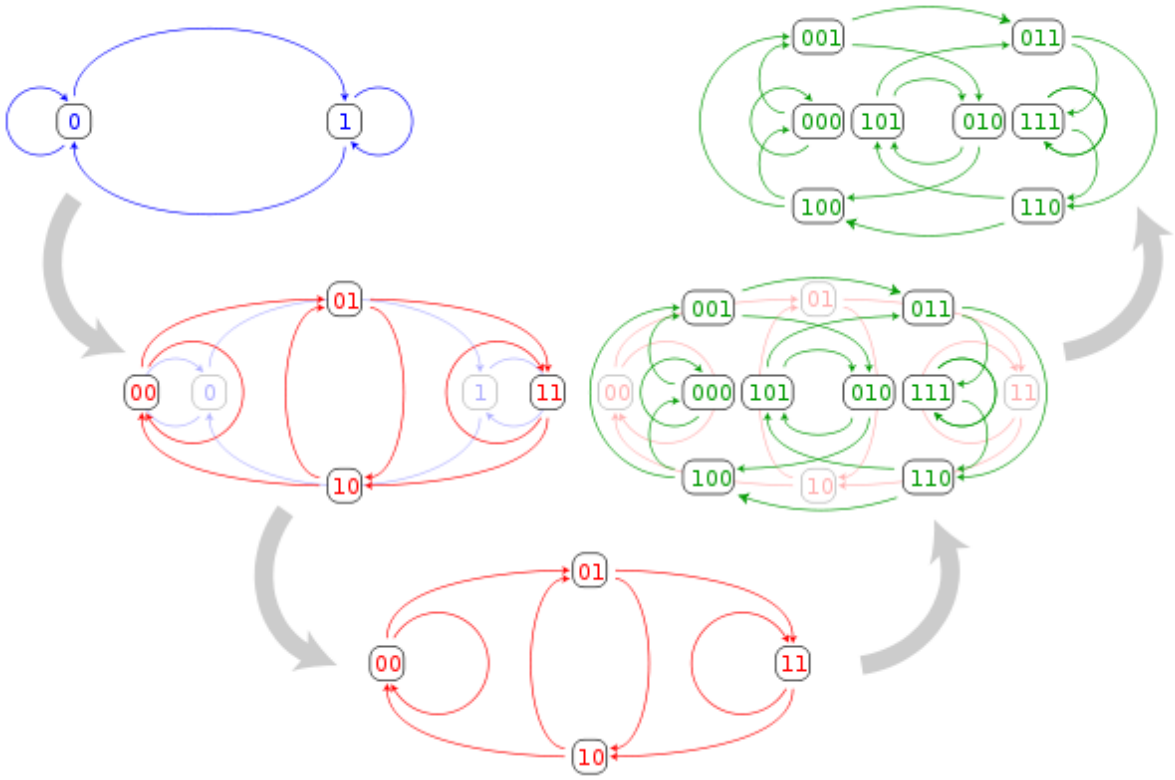
$6{,}200{,}000C_2$ = **19,219,996,900,000** comparison

De brujin Graph: Alternative Method For Alignment

$$S := \{s_1, \ldots, s_m\}$$

$$V = S^m = \{(s_1, \ldots, s_1, s_1), (s_1, \ldots, s_1, s_2), \ldots, (s_1, \ldots, s_1, s_m), (s_1, \ldots, s_2, s_1), \ldots, (s_m, \ldots, s_m, s_m)\}.$$

$$E = \{((v_1, v_2, \ldots, v_n), (w_1, w_2, \ldots, w_n)) : v_2 = w_1, v_3 = w_2, \ldots, v_n = w_{n-1}\}.$$



De brujin Graph Algorithm For Alignment (1)

- This algorithm has been utilized for finding overlapped short-read sequences quickly.

- This algorithm consists of three parts:

    i) **Generating k-mer sequences**

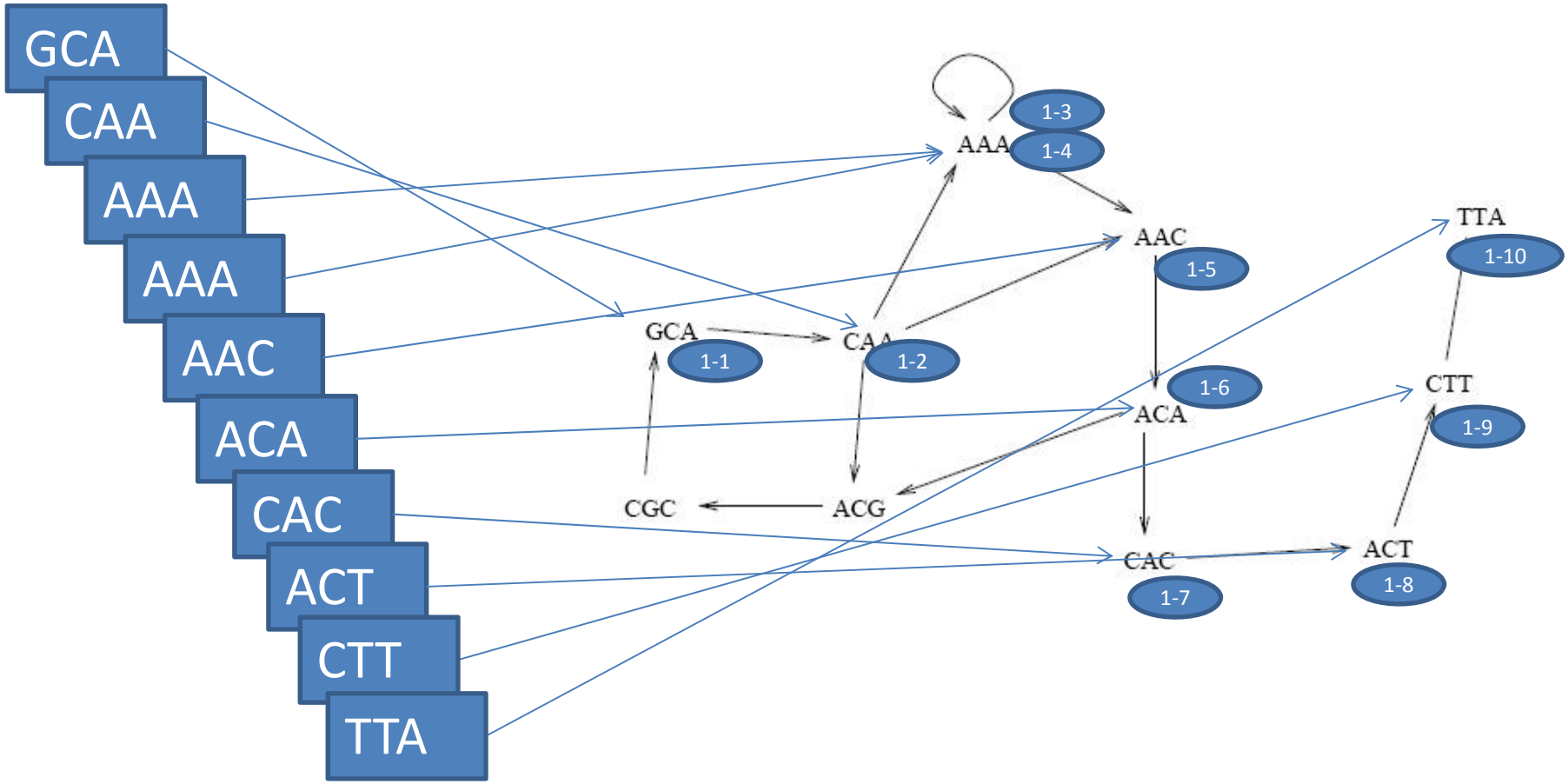
    ii) **Constructing de brujin graph**

    iii) **Resolving the graph with generating sequences**

De brujin Graph Algorithm For Alignment (2)

GCAAAACACTTA…

K = 3

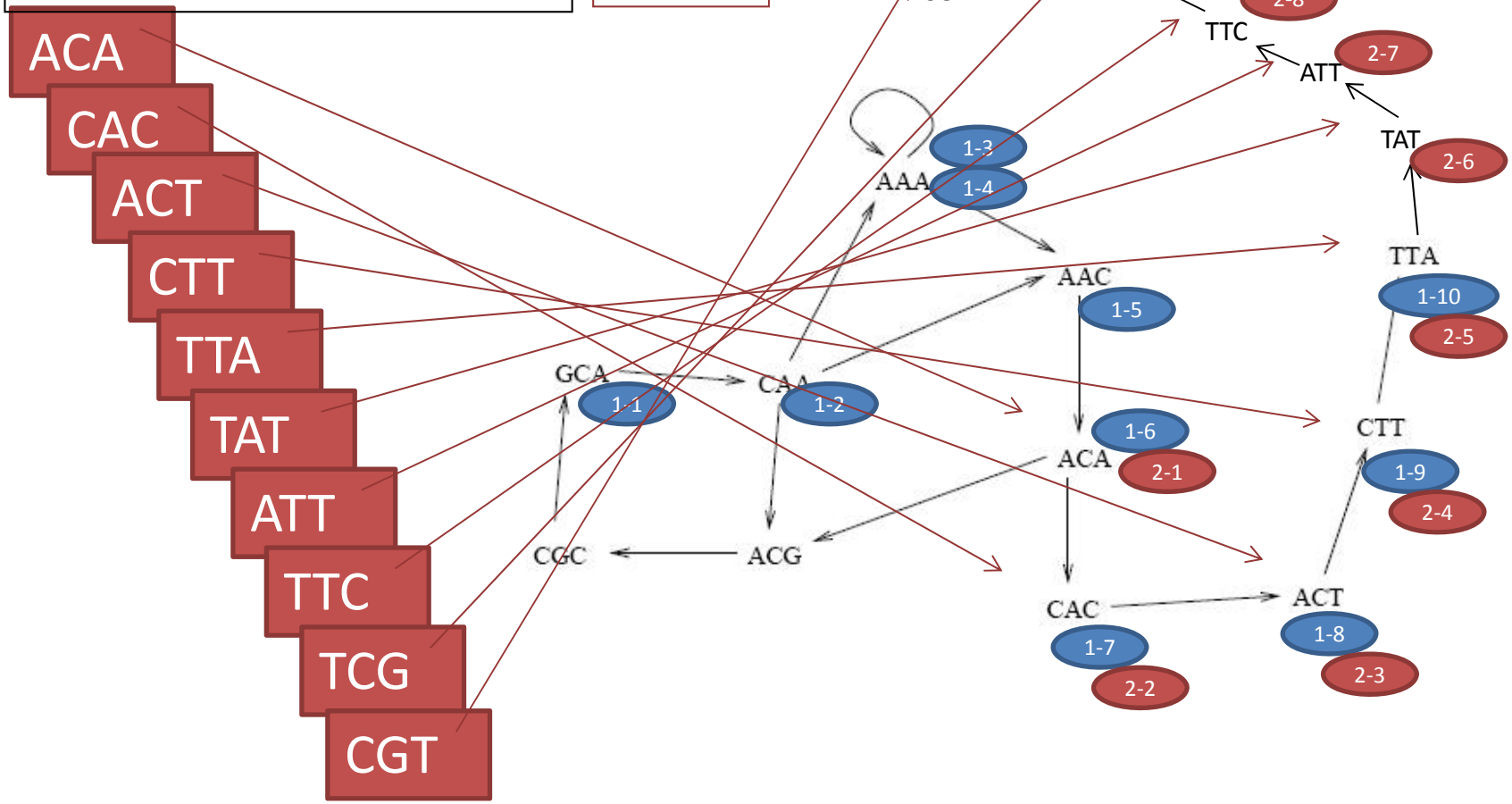


De brujin Graph Algorithm For Alignment (3)

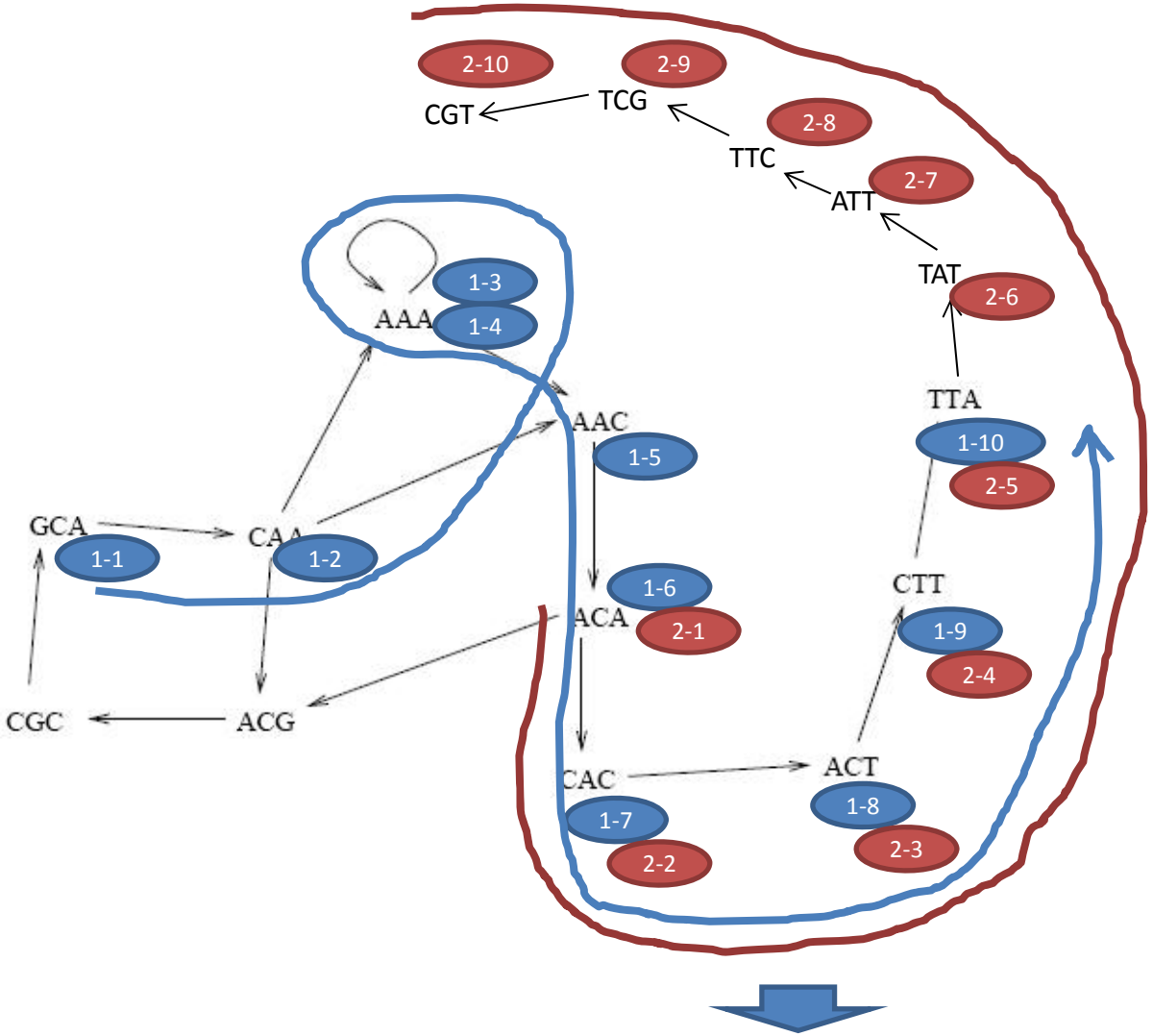
GCAAAACACTTA…

ACACTTATTCGT

K = 3



De brujin Graph Algorithm For Alignment (4)



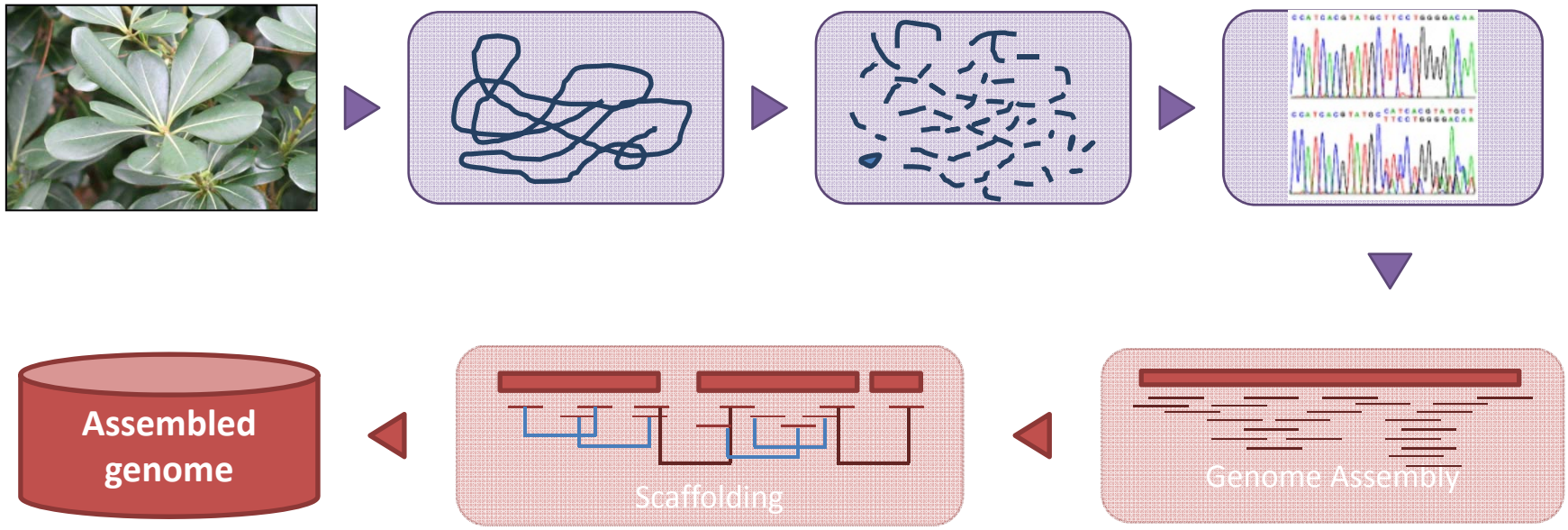
GCAAAACACTTATTCGT

How to Assembly Chloroplast Sequences

Revisit: Whole Shotgun Sequence Strategy

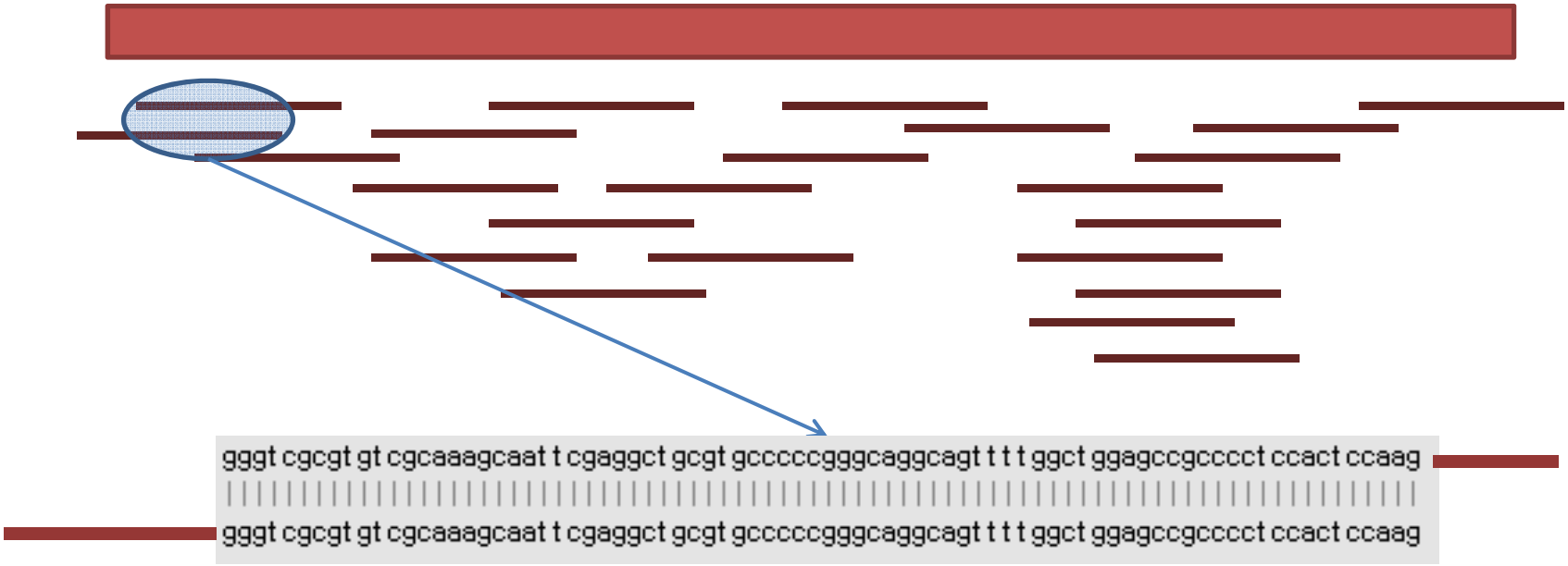
- Assembly process is essential for genome project because read length of each sequence is less than 1 kb.

- Assembly process was conducted by several popular programs, such as phrap and PCAP3, for Sanger sequences.



Revisit: Genome Assembly Process

- We can perform genome assembly manually!



23 sequences should be **compared** with each other!

$_{23}C_2$ = 23*22 / 2 = 253 comparison!

Genome Assemblers For NGS Sequences (1)

Resource

Velvet: Algorithms for de novo short read assembly using de Bruijn graphs

Daniel R. Zerbino and Ewan Birney[1]

EMBL-European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, United Kingdom

We have developed a new set of algorithms, collectively called "Velvet," to manipulate de Bruijn graphs for genomic sequence assembly. A de Bruijn graph is a compact representation based on short words (*k*-mers) that is ideal for high coverage, very short read (25–50 bp) data sets. Applying Velvet to very short reads and paired-ends information only, one can produce contigs of significant length, up to 50-kb N50 length in simulations of prokaryotic data and 3-kb N50 on simulated mammalian BACs. When applied to real Solexa data sets without read pairs, Velvet generated contigs of ~8 kb in a prokaryote and 2 kb in a mammalian BAC, in close agreement with our simulated results without read-pair information. Velvet represents a new approach to assembly that can leverage very short reads in combination with read pairs to produce useful assemblies.

[Supplemental material is available online at www.genome.org. The code for Velvet is freely available, under the GNU Public License, at http://www.ebi.ac.uk/~zerbino/velvet.]

Velvet Assembler (1)

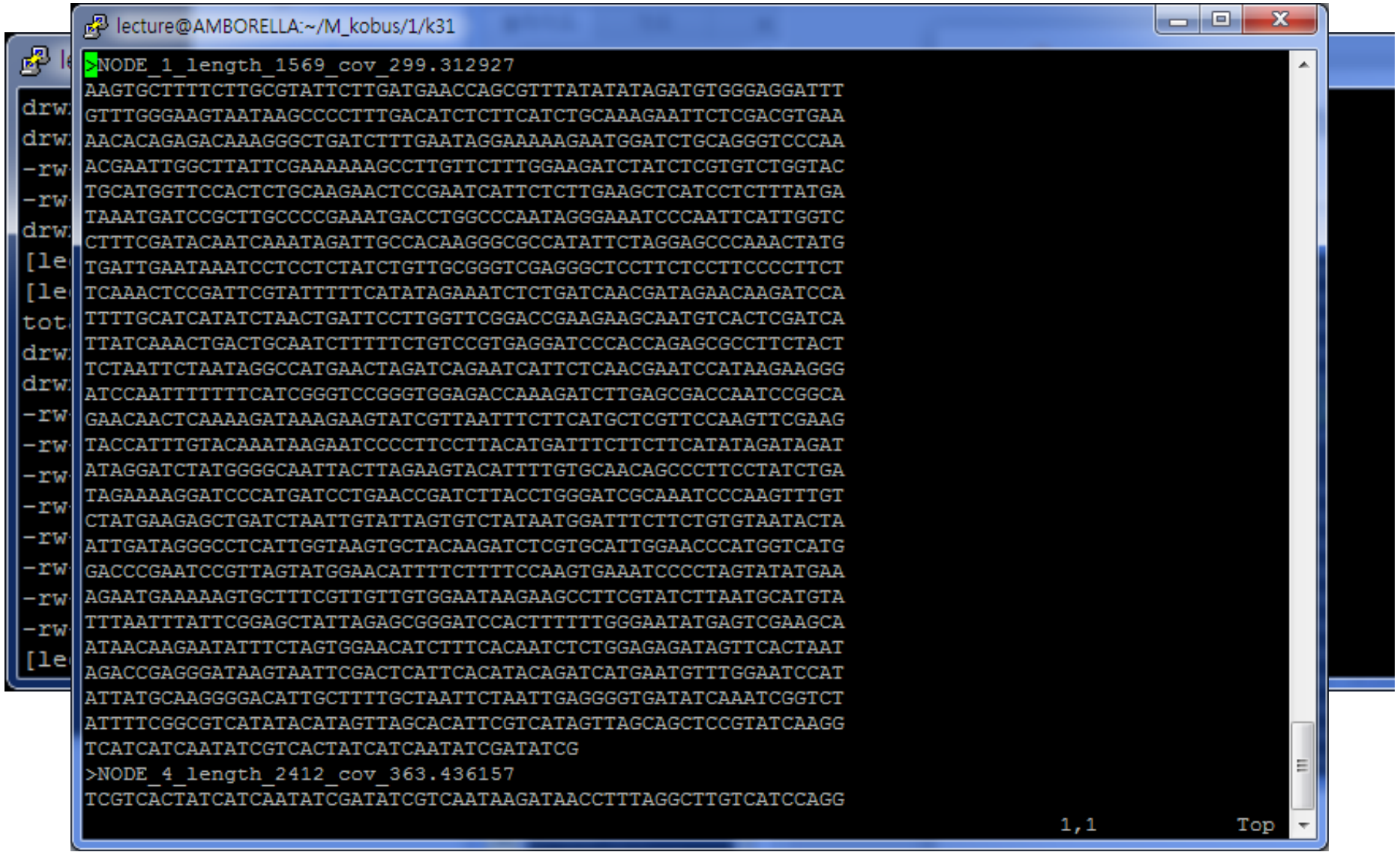
```
lecture@AMBORELLA:~/M_kobus/1
[lecture@AMBORELLA 1]$ ~exeman/velvet_1.1.03/velveth k31 31 -shortPaired -fastq 90000.0-1.fastq 90000.0-2.fastq
[0.000000] Reading FastQ file 90000.0-1.fastq
[0.193801] 90064 reads found.
[0.193810] Done
[0.193821] Reading FastQ file 90000.0-2.fastq
[0.388996] 90053 reads found.
[0.389005] Done
[0.389037] Reading read set file k31/Sequences;
[0.427236] 180117 sequences found
[0.595392] Done
[0.595403] 180117 sequences in total.
[0.595502] Writing into roadmap file k31/Roadmaps...
[0.673878] Inputting sequences...
[0.673890] Inputting sequence 0 / 180117
[3.079403]  === Sequences loaded in 2.405528 s
[3.079434] Done inputting sequences
[3.079438] Destroying splay table
[3.091199] Splay table destroyed
[lecture@AMBORELLA 1]$
```

~ex tq

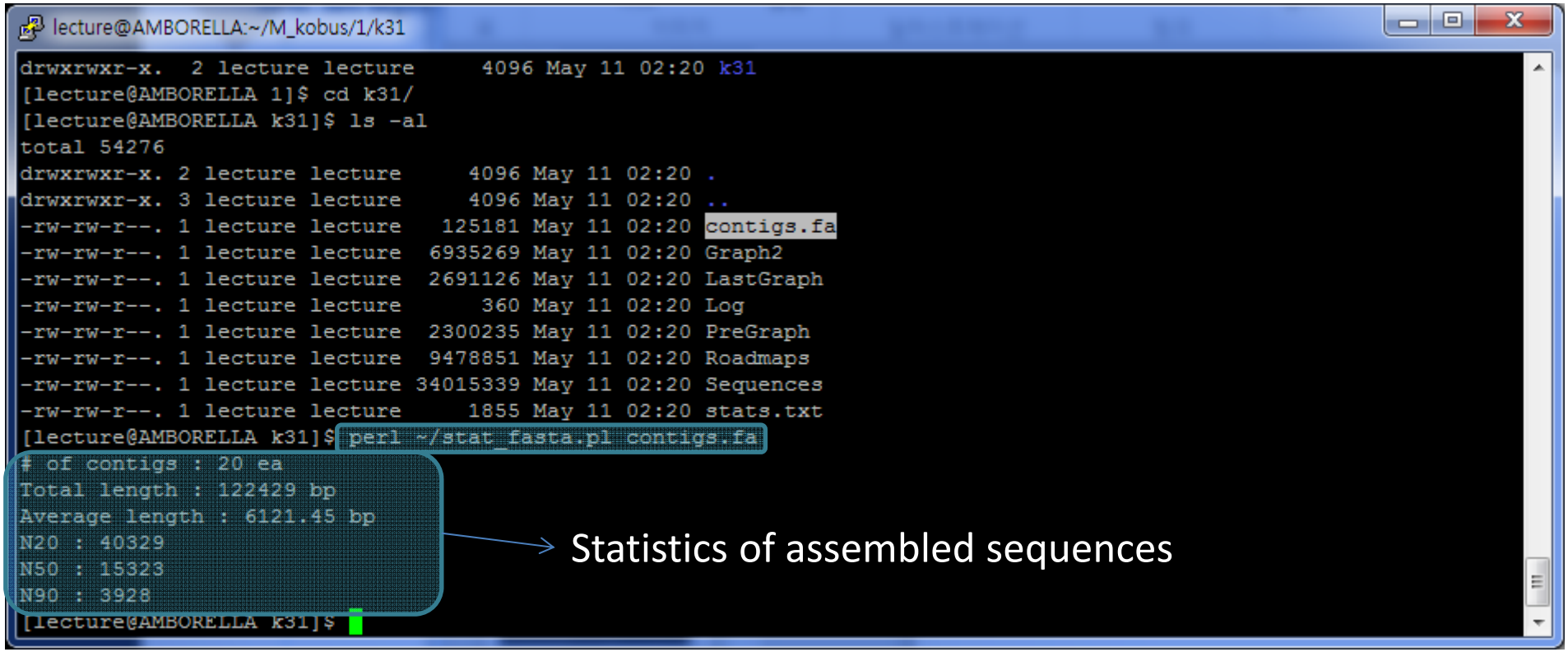
```
lecture@AMBORELLA:~/M_kobus/1
[0.595392] Done
[0.595403] 180117 sequences in total.
[0.595502] Writing into roadmap file k31/Roadmaps...
[0.673878] Inputting sequences...
[0.673890] Inputting sequence 0 / 180117
[3.079403]  === Sequences loaded in 2.405528 s
[3.079434] Done inputting sequences
[3.079438] Destroying splay table
[3.091199] Splay table destroyed
[lecture@AMBORELLA 1]$ ~exeman/velvet_1.1.03/velvetg k31 -ins_length 500 -cov_cutoff auto -exp_cov auto
[0.000000] Reading roadmap file k31/Roadmaps
[0.288275] 180117 roadmaps read
[0.288487] Creating insertion markers
[0.309452] Ordering insertion markers
[0.470370] Counting preNodes
[0.487079] 147937 preNodes counted, creating them now
[1.053024] Adjusting marker info...
[1.071163] Connecting preNodes
```

~exeman/velvet_1.1.03/velvetg k31 -ins_length 500 -cov_cutoff auto -exp_cov auto

Velvet Assembler (2)



Velvet Assembler (3)



N50

- Sort all contigs from largest to smallest
- Determining the minimum set of contigs whose sizes total 50% of the entire genome
- The N50 is distinct from the mean and the median!

- 3 3 4 6 7 8 8 9 9 9 10 11 13 25
- Mean = 125/14= 8.93 (sum=125)
- Median = (3+25)/2= 14
- 125/2= 62.5 sum of contig lengths reach to 62.5
  (from the largest to the smallest)
- 25+13+11+10=59
- 25+13+11+10+9=68
- Therefore, N50 = 9

N50 계산의 예

Contig 번호	각각의 contig 길이	누적값		
1	33407	33407		
7	15243	48650		
6	14172	62822	52860.5	N50= 14172
8	9250	72072		
2	8275	80347		
10	5714	86061		
3	5406	91467		
13	5227	96694		
4	3683	100377		
11	2849	103226		
9	1251	104477		
14	663	105140		
5	442	105582		
12	136	105718		
16	2	105720		
15	1	105721		
합계	105721			
합계 /2	52860.5			