Sequence Alignment and Genome Assembly

Zemin Ning

The Wellcome Trust Sanger Institute

wellcome trust sanger institute

SCOTTISH BIOINFORMATICS FORUM

Outline of the Talk:

- *Global and Local Alignment*
- *Statistical* significance of alignment
- *Alignment method*

Biological Motivation
Why We Need Sequence Alignment

- Inference of Homology
  - Two genes are homologous if they share a common evolutionary history.
  - Evolutionary history can tell us a lot about properties of a given gene
  - Homology can be inferred from similarity between the genes
- Searching for Proteins with same or similar functions

Sequence Alignment

Global Alignment:

Goal: How similar are two sequences S_1 and S_2

Input: two sequences S_1, S_2 over the same alphabet

Output: two sequences S'_1, S'_2 of equal length

(S'_1, S'_2 are S_1, S_2 with possibly additional gaps)

Example:

- S_1= GCGCATGGATTGAGCGA
- S_2= TGCGCCATTGATGACC
- A possible alignment:

```
S'1= -GCGC-ATGGATTGAGCGA
S'2= TGCGCCATTGAT-GACC--
```

Sequence Alignment (cont)

Local Alignment:

Goal: Find the pair of substrings in two input sequences which have the highest similarity

Input: two sequences S_1, S_2 over the same alphabet
Output: two sequences S'_1, S'_2 of equal length
(S'_1, S'_2 are substrings of S_1, S_2 with possibly additional gaps)

Example:

- S_1= GCGCATGGATTGAGCGA
- S_2= TGCGCCATTGATGACC
- A possible alignment:

S'_1= ATTGA-G
S'_2= ATTGATG

Global vs. Local Alignment

- The Global Alignment Problem tries to find the longest path between vertices *(0,0)* and (*n*,*m*) in the edit graph.

- The Local Alignment Problem tries to find the longest path among paths between **arbitrary vertices** (*i*,*j*) and (*i'*, *j'*) in the edit graph.

Global vs. Local Alignment (cont'd)

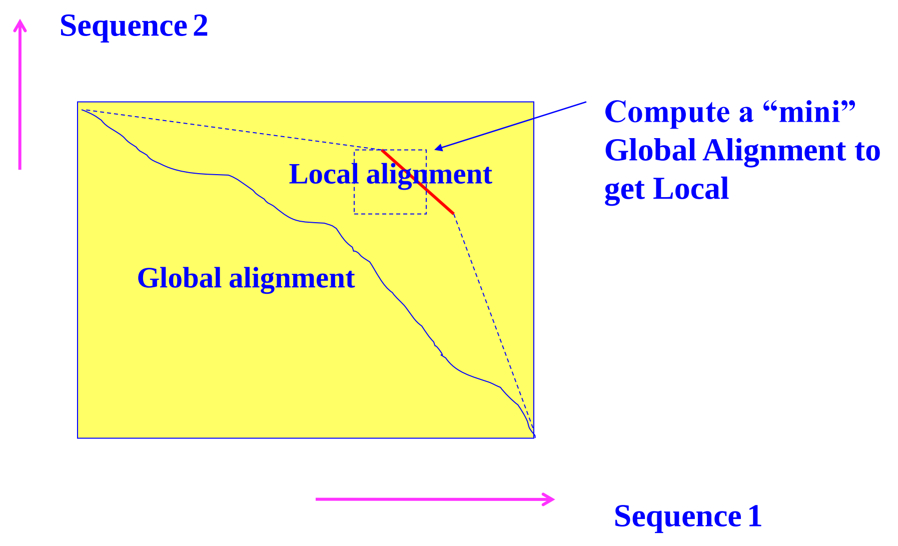
- **Global Alignment**

```
--T—-CC-C-AGT—-TATGT-CAGGGGACACG—A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || | | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG—T-CAGAT--C
```

- **Local Alignment—betten alignment to find conserved segment**

```
                  tccCAGTTATGTCAGgggacacgagcatgcagagac
                     ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

Local Alignment: Example



Statistic Significance of Alignment

We need to know how to evaluate the significance of the alignment. There are two scenarios:

First, the alignment indicates an evolutionary relationship between the sequences.

Second, the alignment is a chance occurrence. What answer is correct?

Here, the statistics are important to estimate of probability that the given alignment score might occur by chance.

E Value (E)

- **E value (E)** of an alignment score is the expected number of unrelated sequences in a database that would have a score at least as good.
- **Low E-values suggest that sequences are homologous.**

  If *E value* ≤ 0.02 sequence probably homologous

  If *E value* ≤ 1 homology cannot be ruled out

  If *E value* > 1 a match just by chance
- **Statistical significance depends on both the size of the alignments and the size of the sequence database**
  - Important consideration for comparing results across different searches
  - E-value increases as database gets bigger
  - E-value decreases as alignments get longer

E value Measuring Alignment Significance

P Value (P)

The *E-value* is not a probability; it's an expected value, i.e. the expected outcome.

Another criteria of the Alignment Significance is the probability that an alignment with this score could have arisen by chance - *p*-value:

$$E\text{-value}(S) = n \bullet p\text{-value}(S),$$

Here n is the number of sequences in the database, *S*.

The lower the p-value, the more likely it is that the alignment score is not by chance but was caused by alignment procedure.

For example, p = .01 means there is a 1 in 100 chance the result occurred by chance.

Methods of DNA Sequence Alignment

- Dot matrix analysis
- The dynamic programming (DP) algorithm
  - Needleman-Wunsch Algorithm
  - Smith-Waterman Algorithm
- Suffix tree
- Hash table based algorithm
- Short read alignment tools

Dot Matrix Analysis

• A dot matrix analysis is a method for comparing two sequences to look for possible alignment (Gibbs and McIntyre 1970)

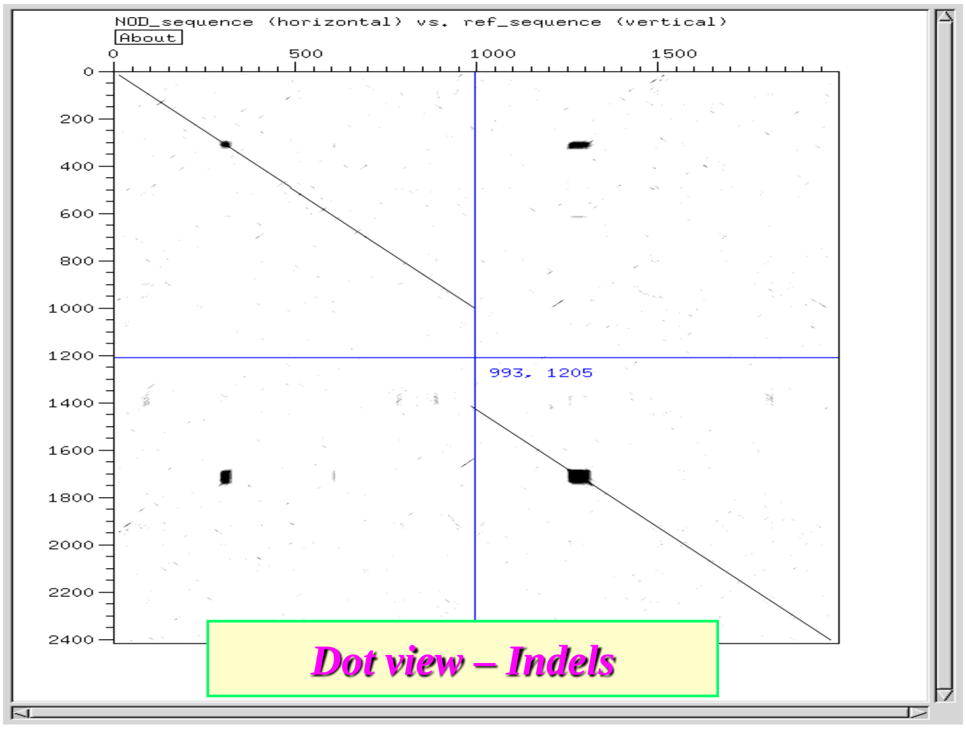
• One sequence (A) is listed across the top of the matrix and the other (B) is listed down the left side

• Starting from the first character in B, one moves across the page keeping in the first row and placing a dot in many column where the character in A is the same

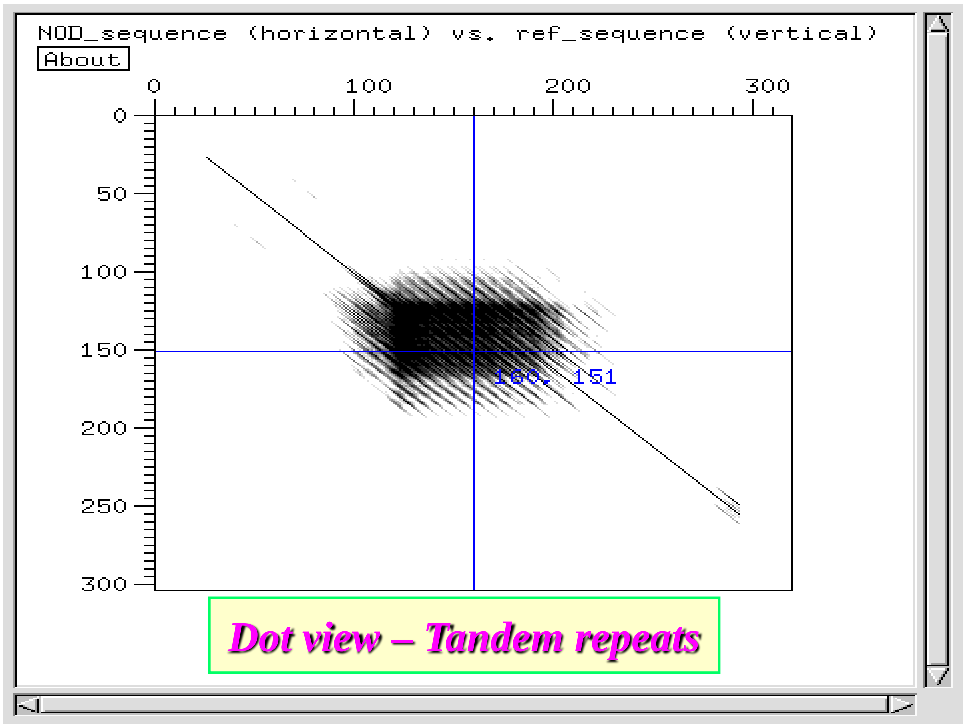
• The process is continued until all possible comparisons between A and B are made

• Any region of similarity is revealed by a diagonal row of dots

• Isolated dots not on diagonal represent random matches



Dot view – Indels

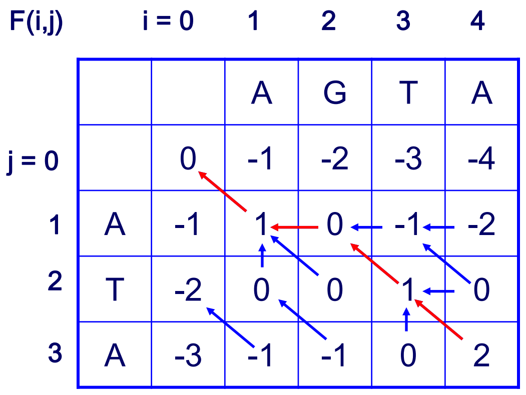


Dot view – Tandem repeats

The Needleman-Wunsch Algorithm

x = AGTA
y = ATA

m = 1
s = -1
d = -1



F(i,j)		i = 0	1	2	3	4
			A	G	T	A
j = 0		0	-1	-2	-3	-4
1	A	-1	1	0	-1	-2
2	T	-2	0	0	1	0
3	A	-3	-1	-1	0	2

Optimal Alignment:

F(4,3) = 2

AGTA
A - TA

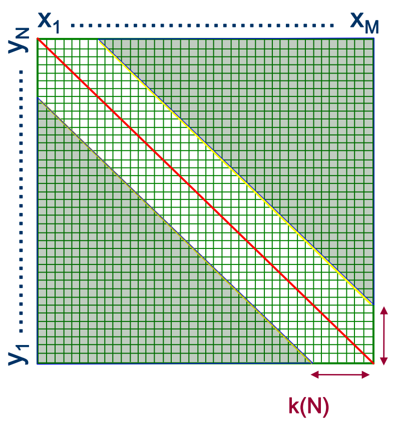
Smith-Waterman Algorithm

- Only works effectively when gap penalties are used
- In example shown
  - match = +1
  - mismatch = -1/3
  - gap = -1+1/3k (k=extent of gap)
- Start with all cell values = 0
- Looks in subcolumn and subrow shown and in direct diagonal for a score that is the highest when you take alignment score or gap penalty into account

	C	A	G	C	C	U	C	G	C	U	U	A	G
A	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
A	0.0	1.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.7
U	0.0	0.0	0.8	0.3	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.7
G	0.0	0.0	1.0	0.3	0.0	0.0	0.7	1.0	0.0	0.0	0.7	0.7	1.0
C	1.0	0.0	0.0	2.0	1.3	0.3	1.0	0.3	2.0	0.7	0.3	0.3	0.3
C	1.0	0.7	0.0	1.0	3.0	1.7	?						
A													
U													
U													
G													
A													
C													
G													
G													

$$H_{ij}=\max\{H_{i-1,\,j-1}+s(a_i,b_j),\ \max\{H_{i-k,j}-W_k\},\ \max\{H_{i,\,j-l}-W_l\},\ 0\}$$

Bounded Dynamic Programming



Initialization:

F(i,0), F(0,j) undefined for i, j > k

Iteration:

```
For i = 1…M
  For j = max(1, i – k)…min(N, i+k)
```

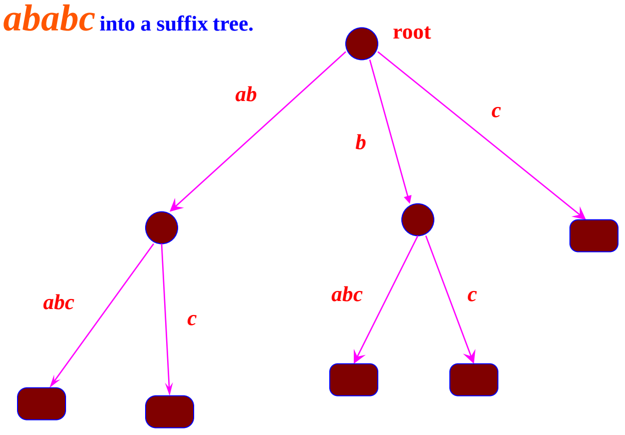
$$F(i,j)=\max\begin{cases}F(i-1,\,j-1)+s(x_i,y_j)\\ F(i,\,j-1)-d, \text{ if } j > i-k(N)\\ F(i-1,\,j)-d, \text{ if } j < i+k(N)\end{cases}$$

Termination: same

Easy to extend to the affine gap case

Suffix Tree Example

Mapping the string
ababc **into a suffix tree.**



Non-overlap Hashing v Overlap Hashing

ATGGCGTGCAGTCCATGTTCGGATCA
ATGGCGTGCAGT
TGGCGTGCAGTC
GGCGTGCAGTCC
GCGTGCAGTCCA
CGTGCAGTCCAT

Non-overlap hashing
W = N-k+1
(k = 12)

ATGGCGTGCAGTCCATGTTCGGATCATTACGTAAGC
ATGGGCAGATGT
CCATGTTCGGAT
CATTACGTAAGC

Overlap Hashing
W = N/k

Sequence Representation

Sequence S: $(s_1 s_2, \ldots, s_i, \ldots, s_m)$ $i = 1, 2, \ldots, m$

K-tuple: $(s_i s_{i+1} \ldots s_{i+k-1})$

Using two binary digits for each base, we may have the following representations:

"A" =00; "C" = 01; "G" = 10; "T" = 11

For any of the m/k no-overlapping k-tuples in the sequence, an integer may be used to represent the k-tuple in a unique way

$$E = \sum_{i=1}^{2k} \beta_i 2^{i-1} \quad \text{with } E_{\max} = 2^{2k} - 1$$

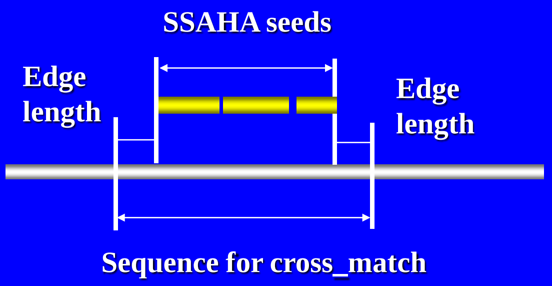
where β_i = 0 or 1, depending on the value of the sequence base and E_{max} is the maximum value of the possible E values.

Hash Table: *A 2-tuple hashing table of S1, S2 and S3*

E	*k*-tuple	N_i	Indices and Offsets						
0	AA	1	2, 19						
1	AC	3	1, 9	2, 5	2, 11				
2	AG	2	1, 15	2, 35					
3	AT	2	2, 13	3, 3					
4	CA	7	2, 3	2, 9	2, 21	2, 27	2, 33	3, 21	3, 23
5	CC	4	1, 21	2, 31	3, 5	3, 7			
6	CG	1	1, 5						
7	CT	6	1, 23	2, 39	2, 43	3, 13	3, 15	3, 17	
8	GA	4	1, 3	1, 17	2, 15	2, 25			
9	GC	0							
10	GG	5	1, 25	1, 31	2, 17	2, 29	3, 1		
11	GT	6	1, 1	1, 27	1, 29	2, 1	2, 37	3, 19	
12	TA	1	3, 25						
13	TC	6	1, 7	1, 11	1, 19	2, 23	2, 41	3, 11	
14	TG	3	1, 13	2, 7	3, 9				
15	TT								

S1=(GTGACGTCACTCTGAGGATCCCCTGGGTGTGG)
S2=(GTCAACTGCAACATGAGGAACATCGACAGGCCCAAGGTCTTCCT)
S3=(GGATCCCCTGTCCTCTCTGTCACATA)

SSAHA2 = SSAHA + Cross_Match



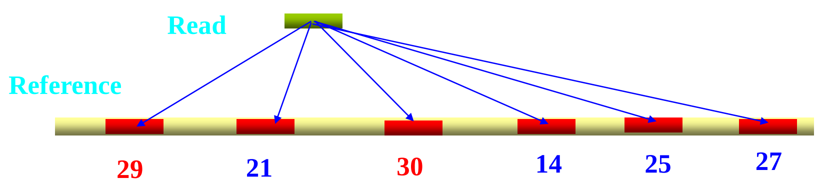
SSAHA for matching seeds, cross_match for sequence alignment.

Mapping Score in ssaha2

Read mapping score is used to assess the repetitive feature of the read in the genome. With a maximum mapping score 50, we have:

$$S_{map} = \begin{cases} 10*(30/R)(S_{\max} - S_{\max 2}) & if\,(S_{map} <= 50) \\ 50 & if\,(S_{map} > \ 50) \end{cases}$$

R = read length; S_{max} - maximum alignment score (smith-waterman) of the hits on genome; S_{max2} - second best alignment score of the hits on genome; Say you have one read of 30 bases which has a few hits on the genome: Best hit: exact match with S_{max} 30; Second best hit: one base mismatch with S_{max2} 29. The mapping score for this read is S_{map} = 10;

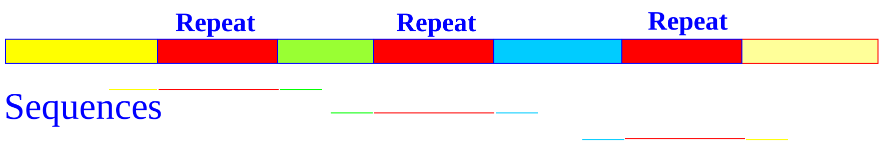


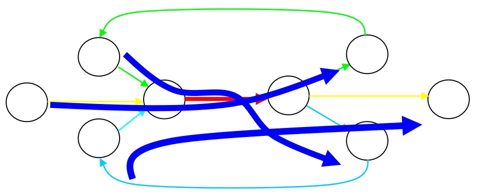
Genome Assembly using Solexa Short Reads Algorithms and Applications

Outline of the Talk:

- *Sequence Reconstruction and Euler Path*
- *Assembly strategy*
- *Sequence extension using read pairs, base qualities, fuzzy kmers or longer reads*
- *Repeat junctions*
- *Gap5 - visual inspection for mis-assembly errors*

Sequence Repeat Graph



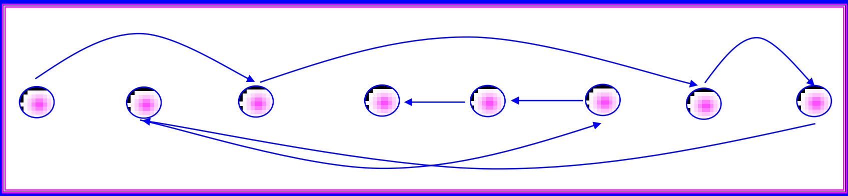


Sequence Reconstruction
- Hamiltonian path approach

S=(ATGCAGGTCC)

ATG -> TGC -> GCA -> CAG -> AGG -> GGT -> GTC -> TCC

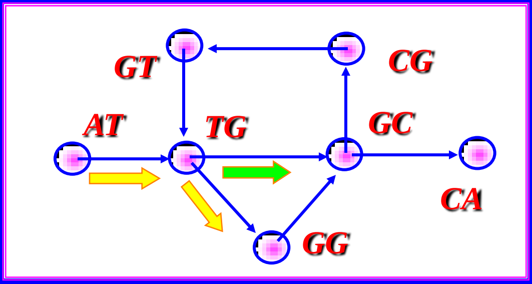
ATG AGG TGC TCC GTC GGT GCA CAG



<u>Vertices</u>: *k-tuples from the spectrum shown in red (8);*
<u>Edges</u>: *overlapping k-tuples (7);*
<u>Path</u>: *visiting all vertices corresponding to the sequence.*

Sequence Reconstruction - Euler path approach

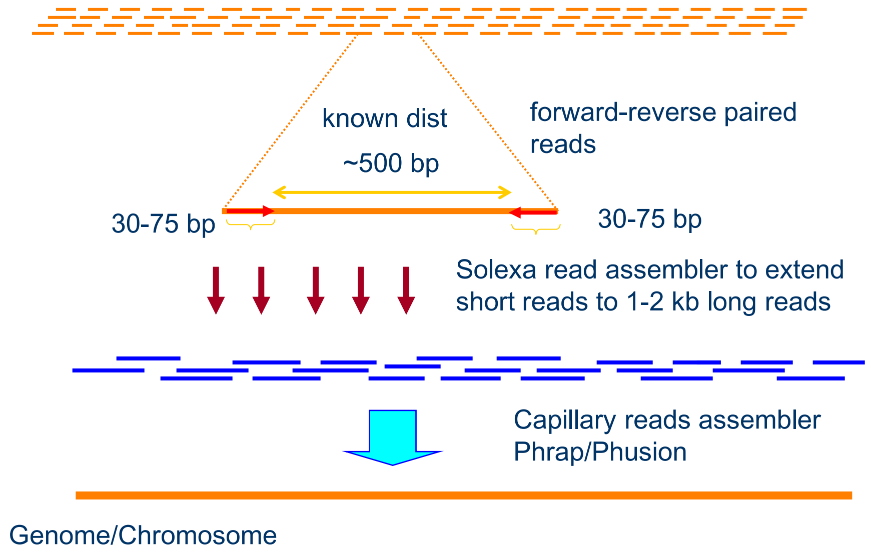
ATG -> TGG -> GGC -> GCG -> CGT -> GTG -> TGC -> GCA

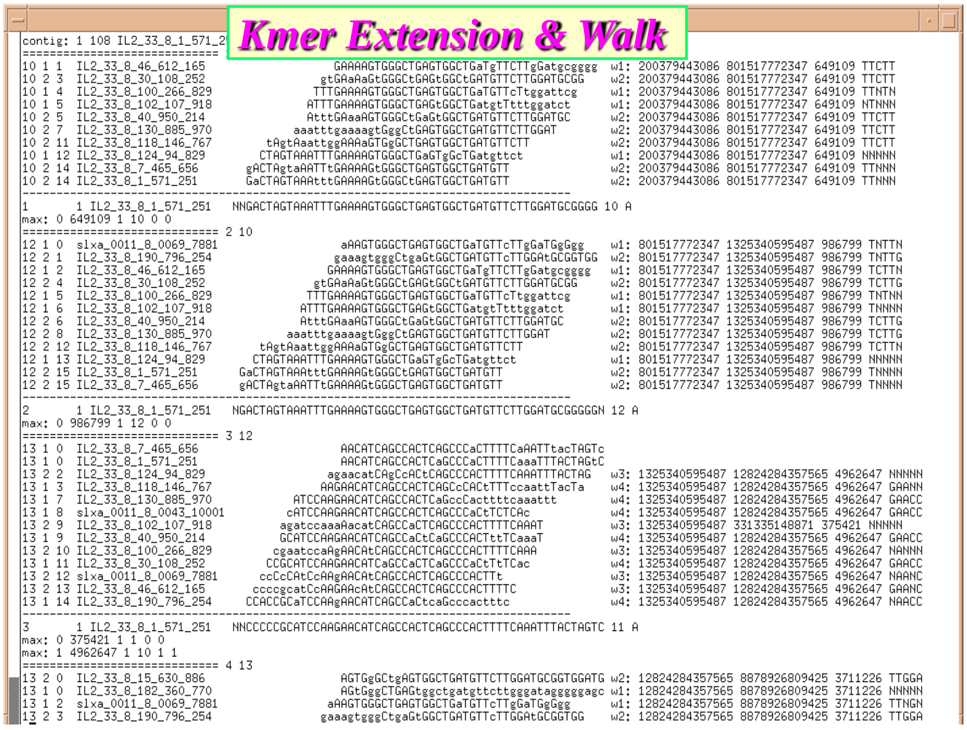


ATGGCGTGCA ATGCGTGGCA

Vertices:	correspond to (k-l)-tuples (7);
Edges:	correspond to k-tuples from the spectrum (8);
Path:	visiting all EDGES corresponding to the sequence.

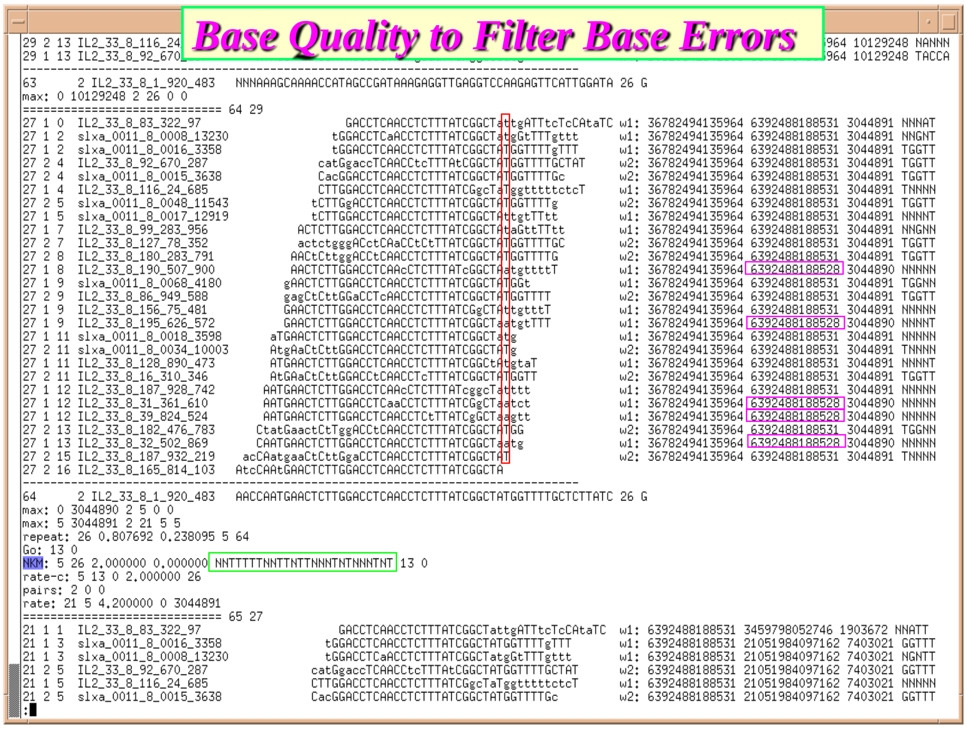
Assembly Strategy


known dist
~500 bp
forward-reverse paired reads
30-75 bp
30-75 bp
Solexa read assembler to extend short reads to 1-2 kb long reads
Capillary reads assembler Phrap/Phusion
Genome/Chromosome



Kmer Extension & Walk

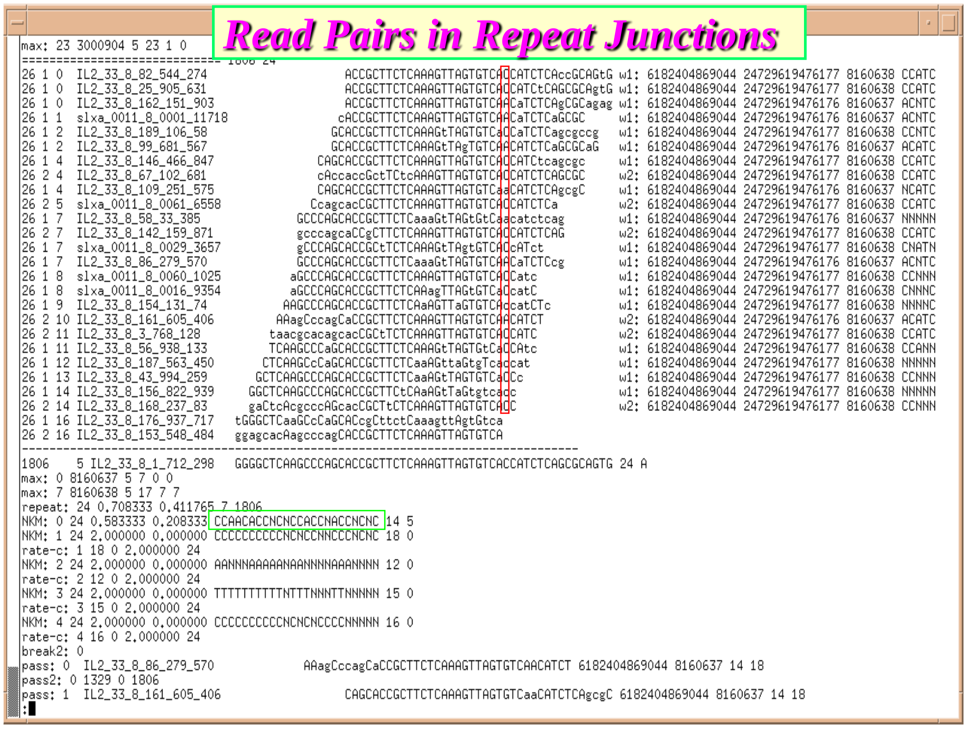
```
contig: 1 108 IL2_33_8_1_571_2
=============================
10 1 1  IL2_33_8_46_612_165                  GAAAAGTGGGCTGAGTGGCTGaTgTTCTTgGatgcgggg  w1: 200379443086 801517772347 649109 TTCTT
10 2 3  IL2_33_8_30_108_252                gtGAaAaGtGGGCtGAGtGGCtGATGTTCTTGGATGCGG    w2: 200379443086 801517772347 649109 TTCTT
10 1 4  IL2_33_8_100_266_829              TTTGAAAAGTGGGCTGAGTGGCTGaTGTTcTtggattcg     w1: 200379443086 801517772347 649109 TTNTN
10 1 5  IL2_33_8_102_107_918             ATTTGAAAAGTGGGCTGAGtGGCTGatgtTtttggatct      w1: 200379443086 801517772347 649109 NTNNN
10 2 5  IL2_33_8_40_950_214             AtttGAaaAGTGGGCtGaGtGGCTGATGTTCTTGGATGC      w2: 200379443086 801517772347 649109 TTCTT
10 2 7  IL2_33_8_130_885_970          aaatttgaaaagtGggCtGAGTGGCTGATGTTCTTGGAT        w2: 200379443086 801517772347 649109 TTCTT
10 2 11 IL2_33_8_118_146_767       tAgtAaattggAAAaGTGgGCTGAGTGGCTGATGTTCTT            w2: 200379443086 801517772347 649109 TTCTT
10 1 12 IL2_33_8_124_94_829       CTAGTAAATTTGAAAAGTGGGCTGaGTgGcTGatgttct             w1: 200379443086 801517772347 649109 NNNNN
10 2 14 IL2_33_8_7_465_656      gACTAgtaAATTtGAAAAGtGGGCTGAGTGGCTGATGTT               w2: 200379443086 801517772347 649109 TTNNN
10 2 14 IL2_33_8_1_571_251      GaCTAGTAAAtttGAAAAGtGGGCtGAGTGGCTGATGTT               w2: 200379443086 801517772347 649109 TTNNN
-------------------------------------------------------------------------------
1       1 IL2_33_8_1_571_251   NNGACTAGTAAATTTGAAAAGTGGGCTGAGTGGCTGATGTTCTTGGATGCGGGG 10 A
max: 0 649109 1 10 0 0
============================= 2 10
12 1 0  slxa_0011_8_0069_7881                 aAAGTGGGCTGAGTGGCTGaTGTTcTTgGaTGgGgg    w1: 801517772347 1325340595487 986799 TNTTN
12 2 1  IL2_33_8_190_796_254                 gaaagtgggCtgaGtGGCTGATGTTcTTGGAtGCGGTGG  w2: 801517772347 1325340595487 986799 TNTTG
12 1 2  IL2_33_8_46_612_165                 GAAAAGTGGGCTGAGTGGCTGaTgTTCTTgGatgcgggg  w1: 801517772347 1325340595487 986799 TCTTN
12 2 4  IL2_33_8_30_108_252               gtGAaAaGtGGGCtGAGtGGCtGATGTTCTTGGATGCGG    w2: 801517772347 1325340595487 986799 TCTTG
12 1 5  IL2_33_8_100_266_829             TTTGAAAAGTGGGCTGAGTGGCTGaTGTTcTtggattcg     w1: 801517772347 1325340595487 986799 TNTNN
12 1 6  IL2_33_8_102_107_918            ATTTGAAAAGTGGGCTGAGtGGCTGatgtTtttggatct      w1: 801517772347 1325340595487 986799 TNNNN
12 2 6  IL2_33_8_40_950_214             AtttGAaaAGTGGGCtGaGtGGCTGATGTTCTTGGATGC      w2: 801517772347 1325340595487 986799 TCTTG
12 2 8  IL2_33_8_130_885_970          aaatttgaaaagtGggCtGAGTGGCTGATGTTCTTGGAT        w2: 801517772347 1325340595487 986799 TCTTG
12 2 12 IL2_33_8_118_146_767      tAgtAaattggAAAaGTGgGCTGAGTGGCTGATGTTCTT            w2: 801517772347 1325340595487 986799 TCTTN
12 1 13 IL2_33_8_124_94_829      CTAGTAAATTTGAAAAGTGGGCTGaGTgGcTGatgttct             w1: 801517772347 1325340595487 986799 NNNNN
12 2 15 IL2_33_8_1_571_251     GaCTAGTAAAtttGAAAAGtGGGCtGAGTGGCTGATGTT               w2: 801517772347 1325340595487 986799 TNNNN
12 2 15 IL2_33_8_7_465_656     gACTAgtaAATTtGAAAAGtGGGCTGAGTGGCTGATGTT               w2: 801517772347 1325340595487 986799 TNNNN
-------------------------------------------------------------------------------
2       1 IL2_33_8_1_571_251   NGACTAGTAAATTTGAAAAGTGGGCTGAGTGGCTGATGTTCTTGGATGCGGGGGN 12 A
max: 0 986799 1 12 0 0
============================= 3 12
13 1 0  IL2_33_8_7_465_656                    AACATCAGCCACTCAGCCCaCTTTTCaAATTtacTAGTc
13 1 0  IL2_33_8_1_571_251                    AACATCAGCCACTCaGCCCaCTTTTCaaaTTTACTAGtC
13 2 2  IL2_33_8_124_94_829                 agaacatCAgCcACtCAGCCCACTTTTCAAATTTACTAG   w3: 1325340595487 12824284357565 4962647 NNNNN
13 1 3  IL2_33_8_118_146_767               AAGAACATCAGCCACTCAGCcCACtTTTccaattTacTa    w4: 1325340595487 12824284357565 4962647 GAANN
13 1 7  IL2_33_8_130_885_970           ATCCAAGAACATCAGCCACTCaGccCacttttcaaattt        w4: 1325340595487 12824284357565 4962647 GAACC
13 1 8  slxa_0011_8_0043_10001        cATCCAAGAACATCAGCCACTCAGCCCaCtTCTCAc            w4: 1325340595487 12824284357565 4962647 GAACC
13 2 9  IL2_33_8_102_107_918          agatccaaaAacatCAGCCaCTCAGCCCACTTTTCAAAT         w3: 1325340595487 331335148871 375421 NNNNN
13 1 9  IL2_33_8_40_950_214           GCATCCAAGAACATCAGCCaCtCaGCCCACTttTCaaaT         w4: 1325340595487 12824284357565 4962647 GAACC
13 2 10 IL2_33_8_100_266_829         cgaatccaAgAACAtCAGCCACTCAGCCCACTTTTCAAA          w3: 1325340595487 12824284357565 4962647 NANNN
13 1 11 IL2_33_8_30_108_252         CCGCATCCAAGAACATCaGCCaCTCaGCCCaCtTtTCac           w4: 1325340595487 12824284357565 4962647 GAACC
13 2 12 slxa_0011_8_0069_7881      ccCcCAtCcAAgAACAtCAGCCACTCAGCCCACTTt              w3: 1325340595487 12824284357565 4962647 NAANC
13 2 13 IL2_33_8_46_612_165       ccccgcatCcAAGAAcAtCAGCCACTCAGCCCACTTTTC            w3: 1325340595487 12824284357565 4962647 GAANC
13 1 14 IL2_33_8_190_796_254     CCACCGCaTCCAAgAACATCAGCCaCtcaGcccactttc             w4: 1325340595487 12824284357565 4962647 NAACC
-------------------------------------------------------------------------------
3       1 IL2_33_8_1_571_251   NNCCCCCGCATCCAAGAACATCAGCCACTCAGCCCACTTTTCAAATTTACTAGTC 11 A
max: 0 375421 1 1 0 0
max: 1 4962647 1 10 1 1
============================= 4 13
13 2 0  IL2_33_8_15_630_886                   AGTGgGCtgAGTGGCTGATGTTCTTGGATGCGGTGGATG w2: 12824284357565 8878926809425 3711226 TTGGA
13 1 0  IL2_33_8_182_360_770                  AGtGggCTGAGtggctgatgttcttgggatagggggagc w1: 12824284357565 8878926809425 3711226 NNNNN
13 1 2  slxa_0011_8_0069_7881               aAAGTGGGCTGAGTGGCTGaTGTTcTTgGaTGgGgg      w1: 12824284357565 8878926809425 3711226 TTNGN
13 2 3  IL2_33_8_190_796_254               gaaagtgggCtgaGtGGCTGATGTTcTTGGAtGCGGTGG    w2: 12824284357565 8878926809425 3711226 TTGGA
```



Base Quality to Filter Base Errors

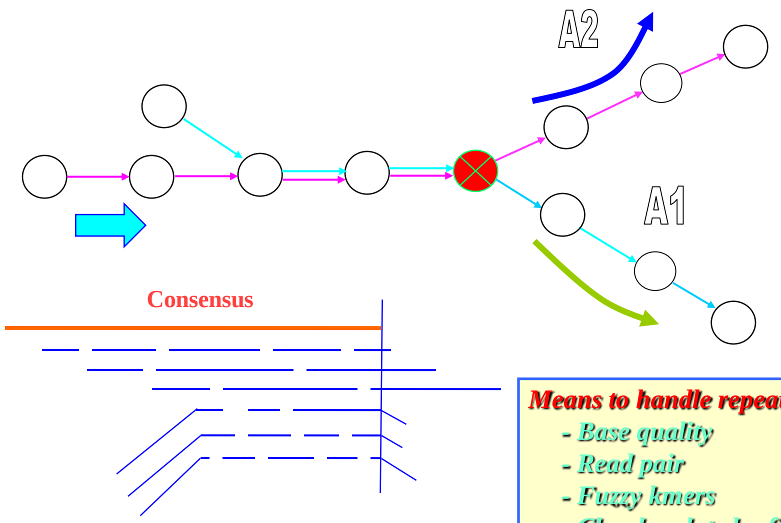
```
29 2 13 IL2_33_8_116_24                                                                          964 10129248 NANNN
29 1 13 IL2_33_8_92_670                                                                          964 10129248 TACCA
-------------------------------------------------------------------------------
63      2 IL2_33_8_1_920_483   NNNAAAGCAAAACCATAGCCGATAAAGAGGTTGAGGTCCAAGAGTTCATTGGATA 26 G
max: 0 10129248 2 26 0 0
============================= 64 29
27 1 0  IL2_33_8_83_322_97                     GACCTCAACCTCTTTATCGGCTattgATTtcTcCAtaTC w1: 36782494135964 6392488188531 3044891 NNNAT
27 1 2  slxa_0011_8_0008_13230               tGGACCTCaACCTCTTTATCGGCTatgGtTTTgttt        w1: 36782494135964 6392488188531 3044891 NNGNT
27 1 2  slxa_0011_8_0016_3358                tGGACCTCAACCTCTTTATCGGCTATGGTTTTgTTT        w1: 36782494135964 6392488188531 3044891 TGGTT
27 2 4  IL2_33_8_92_670_287                catGgaccTCAACCtcTTTAtCGGCTATGGTTTTGCTAT      w2: 36782494135964 6392488188531 3044891 TGGTT
27 2 4  slxa_0011_8_0015_3638              CacGGACCTCAACCTCTTTATCGGCTATGGTTTTGc         w2: 36782494135964 6392488188531 3044891 TGGTT
27 1 4  IL2_33_8_116_24_685                CTTGGACCTCAACCTCTTTATCGgcTaTggtttttctcT      w1: 36782494135964 6392488188531 3044891 TNNNN
27 2 5  slxa_0011_8_0048_11543            tCTTGgACCTCAACCTCTTTATCGGCTATGGTTTTg           w2: 36782494135964 6392488188531 3044891 TGGTT
27 1 5  slxa_0011_8_0017_12919            tCTTGGACCTCAACCTCTTTATCGGCTAttgtTTtt           w1: 36782494135964 6392488188531 3044891 NNNNT
27 1 7  IL2_33_8_99_283_956             ACTCTTGGACCTCAACCTCTTTATCGGCTAtaGttTTtt         w1: 36782494135964 6392488188531 3044891 NNGNN
27 2 7  IL2_33_8_127_78_352             actctgggACctCAaCCtCtTTATCGGCTATGGTTTTGC         w2: 36782494135964 6392488188531 3044891 TGGTT
27 2 8  IL2_33_8_180_283_791            AACtCttggACCtCAACCTCTTTATCGGCTATGGTTTTG          w2: 36782494135964 6392488188531 3044891 TGGTT
27 1 8  IL2_33_8_190_507_900            AACTCTTGGACCTCAAcCTCTTTATcGGCTAatgttttT          w1: 36782494135964 6392488188528 3044890 NNNNN
27 1 9  slxa_0011_8_0068_4180          gAACTCTTGGACCTCAACCTCTTTATCGGCTATGGt               w1: 36782494135964 6392488188531 3044891 TGGNN
27 2 9  IL2_33_8_86_949_588            gagCtCttGGaCCTcAACCTCTTTATCGGCTATGGTTTT           w2: 36782494135964 6392488188531 3044891 TGGTT
27 1 9  IL2_33_8_156_75_481            GAACTCTTGGACCTCAACCTCTTTATCGgCTAttgtttT           w1: 36782494135964 6392488188531 3044891 NNNNN
27 1 9  IL2_33_8_195_626_572           GAACTCTTGGACCTCAACCTCTTTATCGGCTaatgtTTT           w1: 36782494135964 6392488188528 3044890 NNNNT
27 1 11 slxa_0011_8_0018_3598         aTGAACTCTTGGACCTCAACCTCTTTATCGGCTatg                w1: 36782494135964 6392488188531 3044891 NNNNN
27 2 11 slxa_0011_8_0034_10003        AtgAaCtCttGGACCTCAACCTCTTTATCGGCTATg                w2: 36782494135964 6392488188531 3044891 TNNNN
27 1 11 IL2_33_8_128_890_473          ATGAACTCTTGGACCTCAACCTCTTTATCGGCtAtgtaT             w1: 36782494135964 6392488188531 3044891 NNNNT
27 2 11 IL2_33_8_16_310_346           AtGAaCtCttGGACCtCAACCTcTTTATCGGCTATGGTT             w2: 36782494135964 6392488188531 3044891 TGGTT
27 1 12 IL2_33_8_187_928_742         AATGAACTCTTGGACCTCAAcCTCTTTATcggcTatttt              w1: 36782494135964 6392488188531 3044891 NNNNN
27 1 12 IL2_33_8_31_361_610          AATGAACTCTTGGACCTCaaCCTCTTTATCGgCTaatct              w1: 36782494135964 6392488188528 3044890 NNNNN
27 1 12 IL2_33_8_39_824_524          AATGAACTCTTGGACCTCAACCTCtTTATCgGCTaagtt              w1: 36782494135964 6392488188528 3044890 NNNNN
27 2 13 IL2_33_8_182_476_783        CtatGaactCtTggACCtCAACCTCTTTATCGGCTATGG               w2: 36782494135964 6392488188531 3044891 TGGNN
27 1 13 IL2_33_8_32_502_869         CAATGAACTCTTGGACCTCAACCTCTTTATCGGCTaatg               w1: 36782494135964 6392488188528 3044890 NNNNN
27 2 15 IL2_33_8_187_932_219      acCAatgaaCtCttGgaCCTCAACCTCTTTATCGGCTAT                 w2: 36782494135964 6392488188531 3044891 TNNNN
27 2 16 IL2_33_8_165_814_103     AtcCAAtGAACTCTTGGACCTCAACCTCTTTATCGGCTA
-------------------------------------------------------------------------------
64      2 IL2_33_8_1_920_483   AACCAATGAACTCTTGGACCTCAACCTCTTTATCGGCTATGGTTTTGCTCTTATC 26 G
max: 0 3044890 2 5 0 0
max: 5 3044891 2 21 5 5
repeat: 26 0.807692 0.238095 5 64
Go: 13 0
NKM: 5 26 2.000000 0.000000 NNTTTTTNNTTNTTNNNTNTNNNTNT 13 0
rate-c: 5 13 0 2.000000 26
pairs: 2 0 0
rate: 21 5 4.200000 0 3044891
============================= 65 27
21 1 1  IL2_33_8_83_322_97                    GACCTCAACCTCTTTATCGGCTattgATTtcTcCAtaTC  w1: 6392488188531 3459798052746 1903672 NNATT
21 1 3  slxa_0011_8_0016_3358               tGGACCTCAACCTCTTTATCGGCTATGGTTTTgTTT         w1: 6392488188531 21051984097162 7403021 GGTTT
21 1 3  slxa_0011_8_0008_13230              tGGACCTCaACCTCTTTATCGGCTatgGtTTTgttt         w1: 6392488188531 21051984097162 7403021 NGNTT
21 2 5  IL2_33_8_92_670_287               catGgaccTCAACCtcTTTAtCGGCTATGGTTTTGCTAT       w2: 6392488188531 21051984097162 7403021 GGTTT
21 1 5  IL2_33_8_116_24_685               CTTGGACCTCAACCTCTTTATCGgcTaTggtttttctcT       w1: 6392488188531 21051984097162 7403021 NNNNN
21 2 5  slxa_0011_8_0015_3638             CacGGACCTCAACCTCTTTATCGGCTATGGTTTTGc          w2: 6392488188531 21051984097162 7403021 GGTTT
:
```

Read Pairs in Repeat Junctions



```
max: 23 3000904 5 23 1 0
============================= 1806 24
26 1 0  IL2_33_8_82_544_274                    ACCGCTTCTCAAAGTTAGTGTCACCATCTCAccGCAGtG w1: 6182404869044 24729619476177 8160638 CCATC
26 1 0  IL2_33_8_25_905_631                    ACCGCTTCTCAAAGTTAGTGTCACCATCtCAGCGCAgtG w1: 6182404869044 24729619476177 8160638 CCATC
26 1 0  IL2_33_8_162_151_903                   ACCGCTTCTCAAAGTTAGTGTCAACaTCTCAgCGCagag w1: 6182404869044 24729619476176 8160637 ACNTC
26 1 1  slxa_0011_8_0001_11718                cACCGCTTCTCAAAGTTAGTGTCAACaTCTCaGCGC     w1: 6182404869044 24729619476176 8160637 ACNTC
26 1 2  IL2_33_8_189_106_58                  GCACCGCTTCTCAAAGtTAGTGTCaCCaTCTCagcgccg   w1: 6182404869044 24729619476177 8160638 CCNTC
26 1 2  IL2_33_8_99_681_567                  GCACCGCTTCTCAAAGtTAgTGTCAACATCTCaGCGCaG   w1: 6182404869044 24729619476176 8160637 ACATC
26 1 4  IL2_33_8_146_466_847               CAGCACCGCTTCTCAAAGTTAGTGTCACCATCtcagcgc     w1: 6182404869044 24729619476177 8160638 CCATC
26 2 4  IL2_33_8_67_102_681                cAccaccGctTCtcAAAGTTAGTGTCACCATCTCAGCGC     w2: 6182404869044 24729619476177 8160638 CCATC
26 1 4  IL2_33_8_109_251_575               CAGCACCGCTTCTCAAAGTTAGTGTCaaCATCTCAgcgC     w1: 6182404869044 24729619476176 8160637 NCATC
26 2 5  slxa_0011_8_0061_6558             CcagcacCGCTTCTCAAAGTTAGTGTCACCATCTCa         w2: 6182404869044 24729619476177 8160638 CCATC
26 1 7  IL2_33_8_58_33_385              GCCCAGCACCGCTTCTCaaaGtTAGtGtCaacatctcag        w1: 6182404869044 24729619476176 8160637 NNNNN
26 2 7  IL2_33_8_142_159_871            gcccagcaCCgCTTCTCAAAGTTAGTGTCACCATCTCAG        w2: 6182404869044 24729619476177 8160638 CCATC
26 1 7  slxa_0011_8_0029_3657           gCCCAGCACCGCtTCTCAAAGtTAgtGTCACcATct           w1: 6182404869044 24729619476177 8160638 CNATN
26 1 7  IL2_33_8_86_279_570             GCCCAGCACCGCTTCTCaaaGtTAGTGTCAACaTCTCcg        w1: 6182404869044 24729619476176 8160637 ACNTC
26 1 8  slxa_0011_8_0060_1025          aGCCCAGCACCGCTTCTCAAAGTTAGTGTCACCatc            w1: 6182404869044 24729619476177 8160638 CCNNN
26 1 8  slxa_0011_8_0016_9354          aGCCCAGCACCGCTTCTCAAagTTAGtGTCaCcatC            w1: 6182404869044 24729619476177 8160638 CNNNC
26 1 9  IL2_33_8_154_131_74           AAGCCCAGCACCGCTTCTCAaAGTTaGTGTCAccatCTc          w1: 6182404869044 24729619476177 8160638 NNNNC
26 2 10 IL2_33_8_161_605_406         AAagCccagCaCCGCTTCTCAAAGTTAGTGTCAACATCT           w2: 6182404869044 24729619476176 8160637 ACATC
26 2 11 IL2_33_8_3_768_128          taacgcacagcacCGCtTCTCAAAGTTAGTGTCACCATC            w2: 6182404869044 24729619476177 8160638 CCATC
26 1 11 IL2_33_8_56_938_133         TCAAGCCCaGCACCGCTTCTCAAAGtTAGTGtCaCCAtc            w1: 6182404869044 24729619476177 8160638 CCANN
26 1 12 IL2_33_8_187_563_450       CTCAAGCcCaGCACCGCTTCTCaaAGttaGtgTcaccat             w1: 6182404869044 24729619476177 8160638 NNNNN
26 1 13 IL2_33_8_43_994_259       GCTCAAGCCCAGCACCGCTTCTCaaAGtTAGTGTCaCCc              w1: 6182404869044 24729619476177 8160638 CCNNN
26 1 14 IL2_33_8_156_822_939     GGCTCAAGCCCAGCACCGCTTCtCAaAGtTaGtgtcacc               w1: 6182404869044 24729619476177 8160638 NNNNN
26 2 14 IL2_33_8_168_237_83      gaCtcAcgcccAGcacCGCTtCTCAAAGTTAGTGTCACC               w2: 6182404869044 24729619476177 8160638 CCNNN
26 1 16 IL2_33_8_176_937_717   tGGGCTCaaGCcCaGCACcgCttctCaaagttAgtGtca
26 2 16 IL2_33_8_153_548_484   ggagcacAagcccagCACCGCTTCTCAAAGTTAGTGTCA
---------------------------------------------------------------------------------
1806    5 IL2_33_8_1_712_298   GGGGCTCAAGCCCAGCACCGCTTCTCAAAGTTAGTGTCACCATCTCAGCGCAGTG 24 A
max: 0 8160637 5 7 0 0
max: 7 8160638 5 17 7 7
repeat: 24 0.708333 0.411765 7 1806
NKM: 0 24 0.583333 0.208333 CCAACACCNCNCCACCNACCNCNC 14 5
NKM: 1 24 2.000000 0.000000 CCCCCCCCCCNCNCCNNCCCNCNC 18 0
rate-c: 1 18 0 2.000000 24
NKM: 2 24 2.000000 0.000000 AANNNAAAAANAANNNNAAANNNN 12 0
rate-c: 2 12 0 2.000000 24
NKM: 3 24 2.000000 0.000000 TTTTTTTTTTNTTTNNNTTNNNNN 15 0
rate-c: 3 15 0 2.000000 24
NKM: 4 24 2.000000 0.000000 CCCCCCCCCCNCNCNCCCCNNNNN 16 0
rate-c: 4 16 0 2.000000 24
break2: 0
pass: 0  IL2_33_8_86_279_570             AAagCccagCaCCGCTTCTCAAAGTTAGTGTCAACATCT 6182404869044 8160637 14 18
pass2: 0 1329 0 1806
pass: 1  IL2_33_8_161_605_406                  CAGCACCGCTTCTCAAAGTTAGTGTCaaCATCTCAgcgC 6182404869044 8160637 14 18
:
```

Kmer Extension & Repeat Junctions

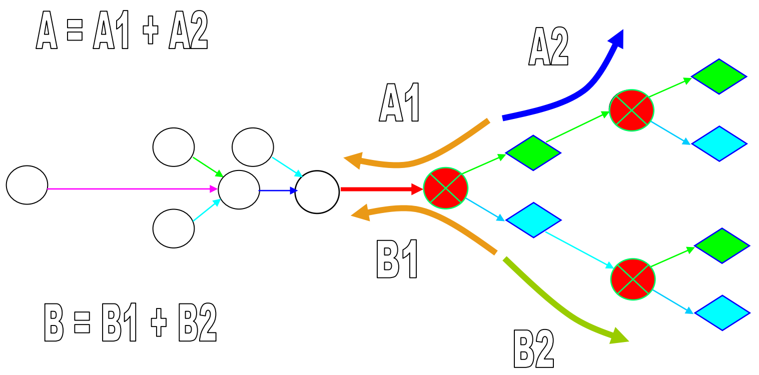


Pileup of other reads like 454, Sanger etc at a repeat junction

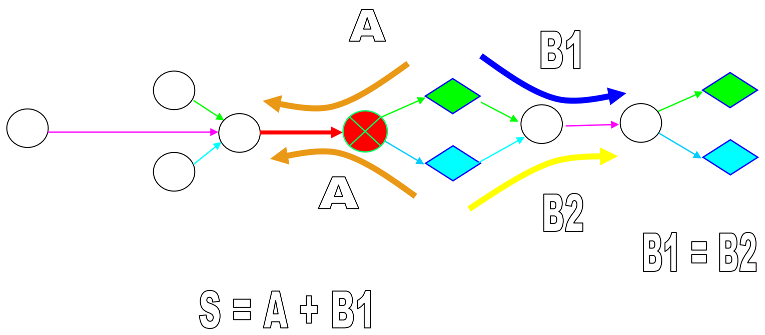
Means to handle repeats:

- *Base quality*
- *Read pair*
- *Fuzzy kmers*
- *Closely related reference*
- *454 or Sanger reads*

Handling of Repeat Junctions



Handling of Single Base Variations



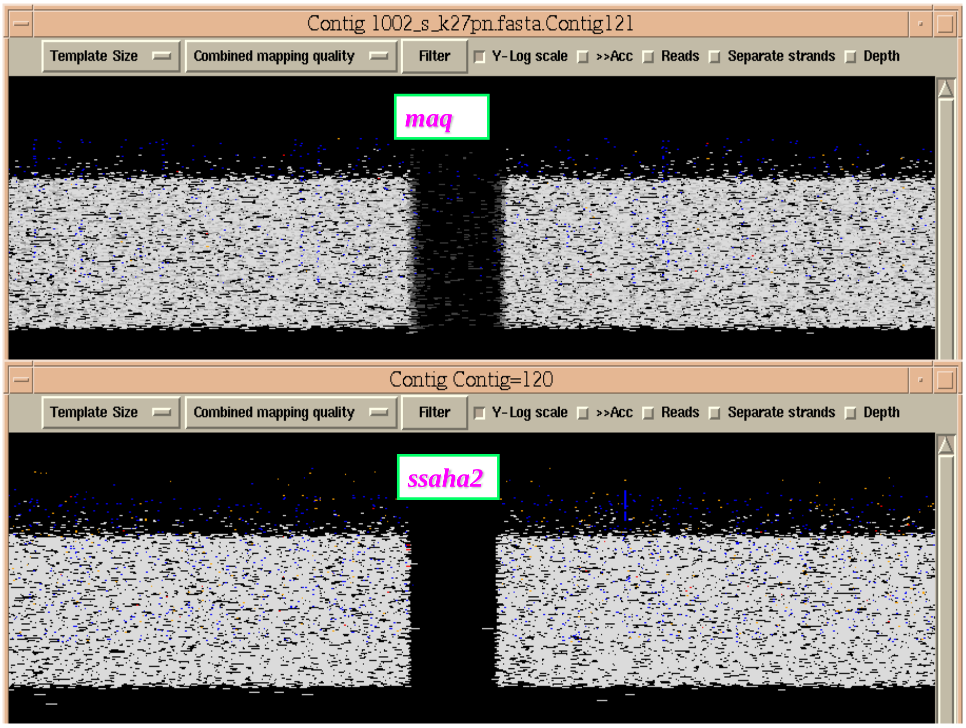
Salmonella seftenberg Solexa Assembly from Pair-End Reads

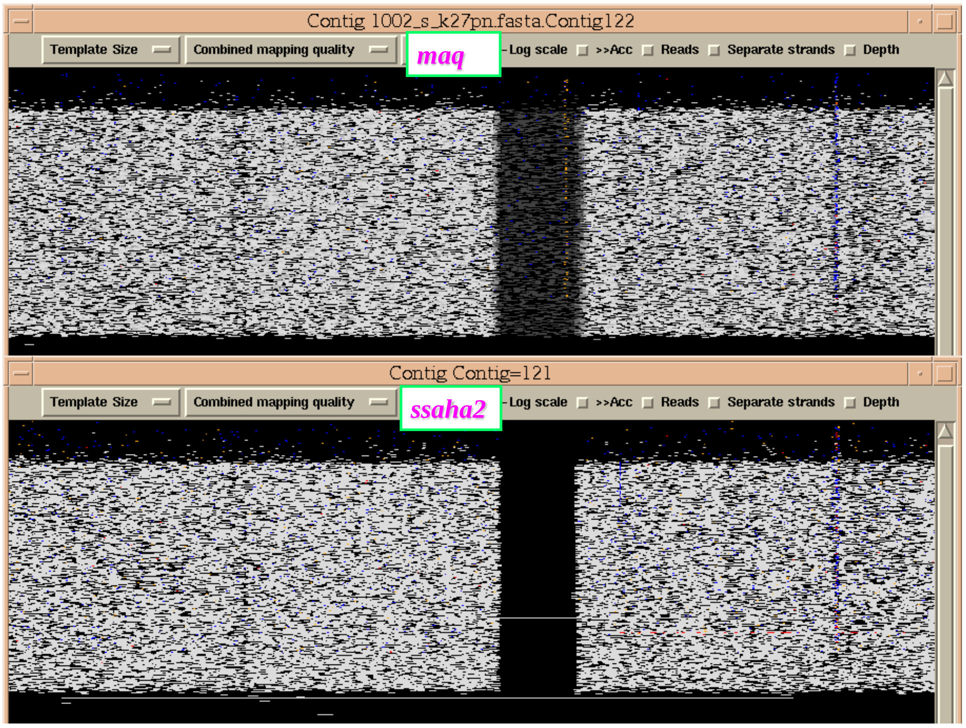
Solexa reads:

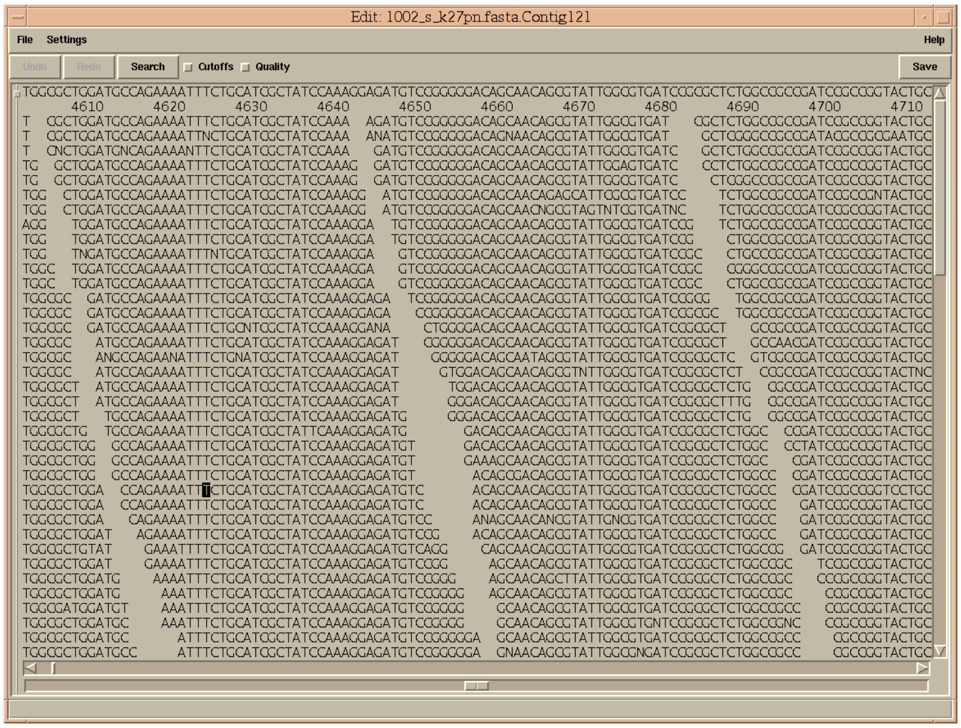
Number of reads:	6,000,000;
Finished genome size:	~4.8 Mbp;
Read length:	2x37 bp;
Estimated read coverage:	~92.5 X;
Insert size:	170/50-300 bp;

Assembly features: - contig stats

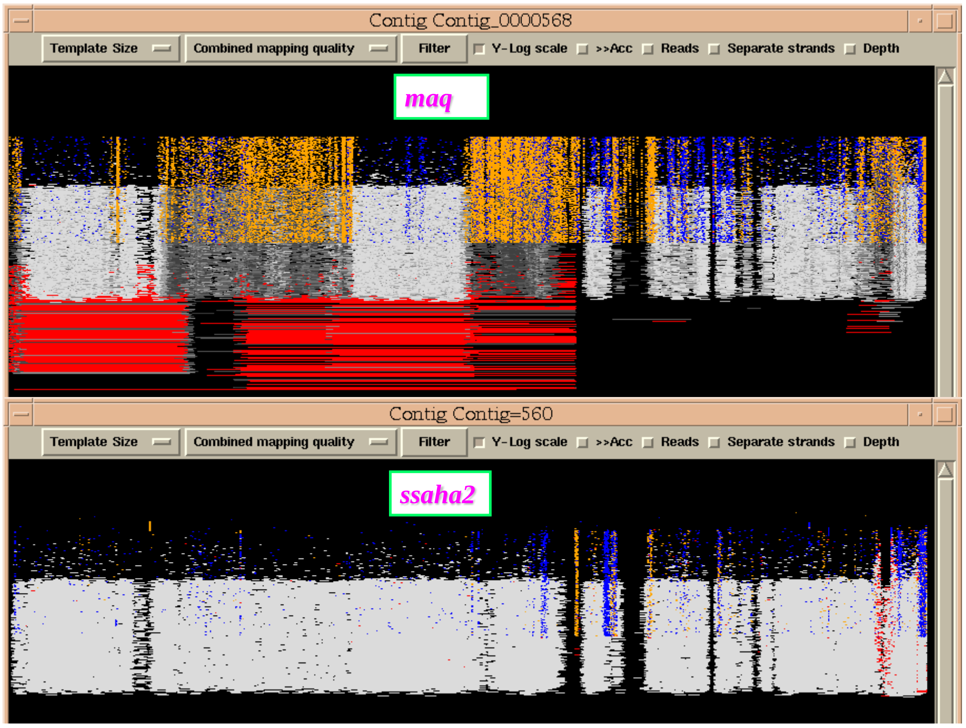
	Solexa	454
Total number of contigs:	75;	390
Total bases of contigs:	4.80 Mbp	4.77 Mb
N50 contig size:	139,353	25,702
Largest contig:	395,600	62,040
Averaged contig size:	63,969	12,224
Contig coverage on genome:	~99.8 %	99.4%
Contig extension errors:	0	
Mis-assembly errors:	0	4

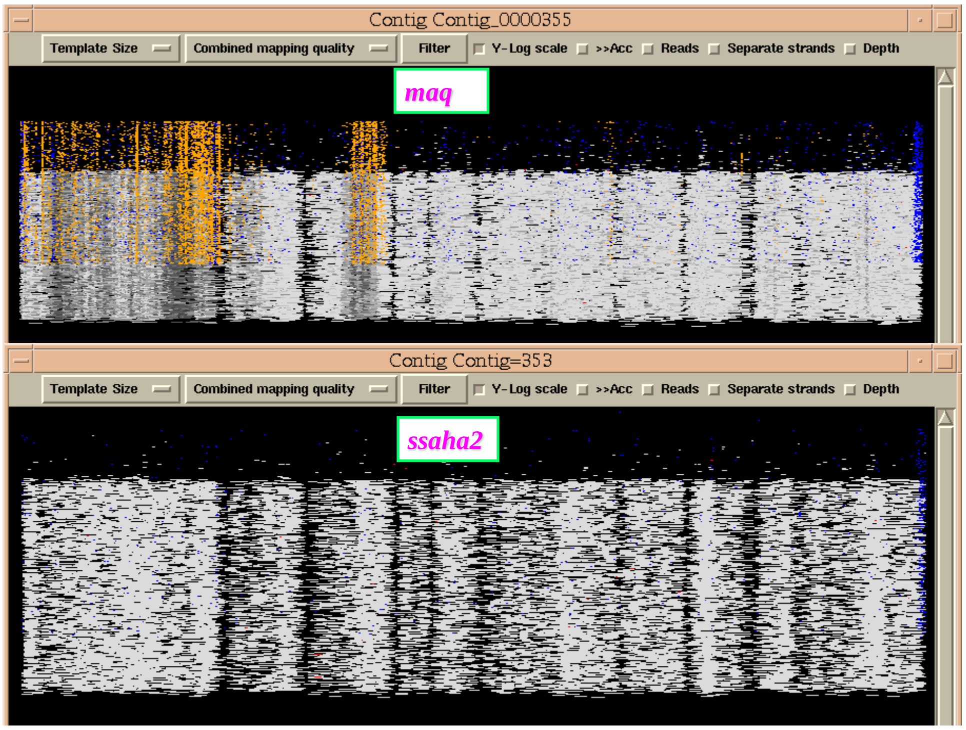


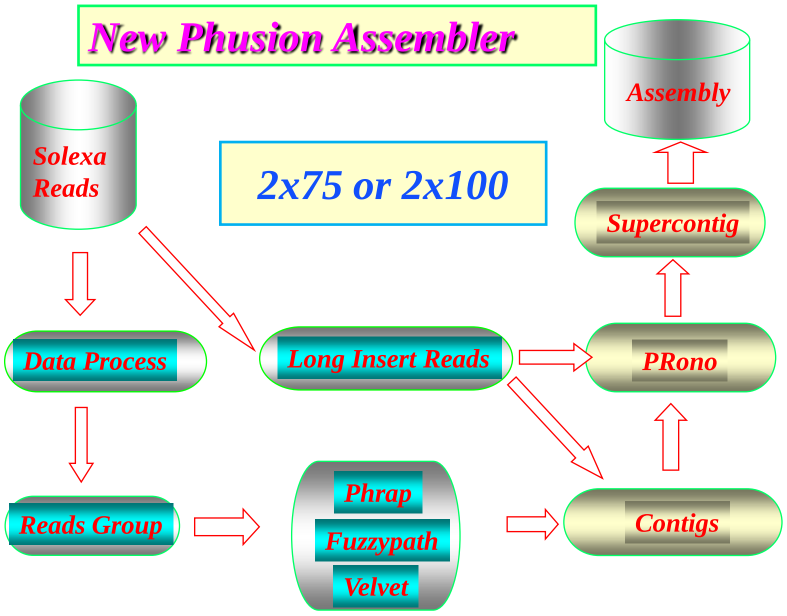




```
TGGCGCTGGATGCCAGAAAATTTCTGCATCGCTATCCAAAGGAGATGTCCGGGGGACAGCAACAGCGTATTGGCGTGATCCGCGCTCTGGCCGCCGATCCGCCGGTACTGC
         4610      4620      4630      4640      4650      4660      4670      4680      4690      4700      4710
T  CGCTGGATGCCAGAAAATTTCTGCATCGCTATCCAAA  AGATGTCCGGGGGACAGCAACAGCGTATTGGCGTGAT   CGCTCTGGCCGCCGATCCGCCGGTACTGC
T  CGCTGGATGCCAGAAAATTNCTGCATCGCTATCCAAA  ANATGTCCGGGGGACAGNAACAGCGTATTGGCGTGAT    GCTCGGGCCGCCGATACGCCGCGAATGC
T  CNCTGGATGNCAGAAAANTTCTGCATCGCTATCCAAA   GATGTCCGGGGGACAGCAACAGCGTATTGGCGTGATC   GCTCTGGCCGCCGATCCGCCGGTACTGC
TG  GCTGGATGCCAGAAAATTTCTGCATCGCTATCCAAAG  GATGTCCGGGGGACAGCAACAGCGTATTGGAGTGATC   CCTCTGGCCGCCGATCCGCCGGTACTGC
TG  GCTGGATGCCAGAAAATTTCTGCATCGCTATCCAAAG  GATGTCCGGGGGACAGCAACAGCGTATTGGCGTGATC    CTCGGCCCGCCGATCCGCCGGTACTGC
TGG  CTGGATGCCAGAAAATTTCTGCATCGCTATCCAAAGG  ATGTCCGGGGGACAGCAACAGAGCATTCGCGTGATCC    TCTGGCCGCCGATCCGCCGNTACTGC
TGG  CTGGATGCCAGAAAATTTCTGCATCGCTATCCAAAGG  ATGTCCGGGGGACAGCAACNGCGTAGTNTCGTGATNC    TCTGGCCGCCGATCCGCCGGTACTGC
AGG   TGGATGCCAGAAAATTTCTGCATCGCTATCCAAAGGA  TGTCCGGGGGACAGCAACAGCGTATTGGCGTGATCCG   TCTGGCCGCCGATCCGCCGGTACTGC
TGG   TGGATGCCAGAAAATTTCTGCATCGCTATCCAAAGGA  TGTCCGGGGGACAGCAACAGCGTATTGGCGTGATCCG    CTGGCCGCCGATCCGCCGGTACTGC
TGG   TNGATGCCAGAAAATTTNTGCATCGCTATCCAAAGGA   GTCCGGGGGACAGCAACAGCGTATTGGCGTGATCCGC   CTGCCCGCCGATCCGCCGGTACTGC
TGGC  TGGATGCCAGAAAATTTCTGCATCGCTATCCAAAGGA   GTCCGGGGGACAGCAACAGCGTATTGGCGTGATCCGC   CGGGCCGCCGATCCGCCGGTACTGC
TGGC  TGGATGCCAGAAAATTTCTGCATCGCTATCCAAAGGA   GTCCGGGGGACAGCAACAGCGTATTGGCGTGATCCGC   CTGGCCGCCGATCCGCCGGTACTGC
TGGCGC  GATGCCAGAAAATTTCTGCATCGCTATCCAAAGGAGA  TCCGGGGGACAGCAACAGCGTATTGGCGTGATCCGCG    TGGCCGCCGATCCGCCGGTACTGC
TGGCGC  GATGCCAGAAAATTTCTGCATCGCTATCCAAAGGAGA   CCGGGGGACAGCAACAGCGTATTGGCGTGATCCGCGC   TGGCCGCCGATCCGCCGGTACTGC
TGGCGC  GATGCCAGAAAATTTCTGCNTCGCTATCCAAAGGANA    CTGGGGACAGCAACAGCGTATTGGCGTGATCCGCGCT    GCCGCCGATCCGCCGGTACTGC
TGGCGC   ATGCCAGAAAATTTCTGCATCGCTATCCAAAGGAGAT   CGGGGGACAGCAACAGCGTATTGGCGTGATCCGCGCT    GCCAACGATCCGCCGGTACTGC
TGGCGC   ANGCCAGAANATTTCTGNATCGCTATCCAAAGGAGAT    GGGGGACAGCAATAGCGTATTGGCGTGATCCGCGCTC   GTCGCCGATCCGCCGGTACTGC
TGGCGC   ATGCCAGAAAATTTCTGCATCGCTATCCAAAGGAGAT     GTGGACAGCAACAGCGTNTTGGCGTGATCCGCGCTCT   CCGCCGATCCGCCGGTACTNC
TGGCGCT  ATGCCAGAAAATTTCTGCATCGCTATCCAAAGGAGAT      TGGACAGCAACAGCGTATTGGCGTGATCCGCGCTCTG   CGCCGATCCGCCGGTACTGC
TGGCGCT  ATGCCAGAAAATTTCTGCATCGCTATCCAAAGGAGAT     GGGACAGCAACAGCGTATTGGCGTGATCCGCGCTTTG   CGCCGATCCGCCGGTACTGC
TGGCGCT   TGCCAGAAAATTTCTGCATCGCTATCCAAAGGAGATG    GGGACAGCAACAGCGTATTGGCGTGATCCGCGCTCTG   CGCCGATCCGCCGGTACTGC
TGGCGCTG  TGCCAGAAAATTTCTGCATCGCTATTCAAAGGAGATG      GACAGCAACAGCGTATTGGCGTGATCCGCGCTCTGGC   CCGATCCGCCGGTACTGC
TGGCGCTGG  GCCAGAAAATTTCTGCATCGCTATCCAAAGGAGATGT     GACAGCAACAGCGTATTGGCGTGATCCGCGCTCTGGC   CCTATCCGCCGGTACTGC
TGGCGCTGG  GCCAGAAAATTTCTGCATCGCTATCCAAAGGAGATGT     GAAAGCAACAGCGTATTGGCGTGATCCGCGCTCTGGC    CGATCCGCCGGTACTGC
TGGCGCTGG  GCCAGAAAATTTCTGCATCGCTATCCAAAGGAGATGT      ACAGCGACAGCGTATTGGCGTGATCCGCGCTCTGGCC   CGATCCGCCGGTACTGC
TGGCGCTGGA  CCAGAAAATTTCTGCATCGCTATCCAAAGGAGATGTC     ACAGCAACAGCGTATTGGCGTGATCCGCGCTCTGGCC   CGATCCGCCGGTCCTGC
TGGCGCTGGA  CCAGAAAATTTCTGCATCGCTATCCAAAGGAGATGTC     ACAGCAACAGCGTATTGGCGTGATCCGCGCTCTGGCC    GATCCGCCGGTACTGC
TGGCGCTGGA   CAGAAAATTTCTGCATCGCTATCCAAAGGAGATGTCC    ANAGCAACANCGTATTGNCGTGATCCGCGCTCTGGCC    GATCCGCCGGTACTGC
TGGCGCTGGAT   AGAAAATTTCTGCATCGCTATCCAAAGGAGATGTCCG    ACAGCAACAGCGTATTGGCGTGATCCGCGCTCTGGCC    GATCCGCCGGTACTGC
TGGCGCTGTAT    GAAATTTTCTGCATCGCTATCCAAAGGAGATGTCAGG    CAGCAACAGCGTATTGGCGTGATCCGCGCTCTGGCCG   GATCCGCCGGTACTGC
TGGCGCTGGAT    GAAAATTTCTGCATCGCTATCCAAAGGAGATGTCCGG    AGCAACAGCGTATTGGCGTGATCCGCGCTCTGGCCGC    TCCGCCGGTACTGC
TGGCGCTGGATG    AAAATTTCTGCATCGCTATCCAAAGGAGATGTCCGGG   AGCAACAGCTTATTGGCGTGATCCGCGCTCTGGCCGC    CCCGCCGGTACTGC
TGGCGCTGGATG     AAATTTCTGCATCGCTATCCAAAGGAGATGTCCGGGG  AGCAACAGCGTATTGGCGTGATCCGCGCTCTGGCCGC     CCGCCGGTACTGC
TGGCGATGGATGT    AAATTTCTGCATCGCTATCCAAAGGAGATGTCCGGGG   GCAACAGCGTATTGGCGTGATCCGCGCTCTGGCCGCC    CCGCCGGTACTGC
TGGCGCTGGATGC    AAATTTCTGCATCGCTATCCAAAGGAGATGTCCGGGG   GCAACAGCGTATTGGCGTGNTCCGCGCTCTGGCCGNC    CCGCCGGTACTGC
TGGCGCTGGATGC      ATTTCTGCATCGCTATCCAAAGGAGATGTCCGGGGGA  GCAACAGCGTATTGGCGTGATCCGCGCTCTGGCCGCC     CGCCGGTACTGC
TGGCGCTGGATGCC     ATTTCTGCATCGCTATCCAAAGGAGATGTCCGGGGGA  GNAACAGCGTATTGGCGNGATCCGCGCTCTGGCCGCC     CGCCGGTACTGC
```


Contig Contig_0000568
Template Size
Combined mapping quality
Filter
Y-Log scale
>>Acc
Reads
Separate strands
Depth
maq
Contig Contig=560
Template Size
Combined mapping quality
Filter
Y-Log scale
>>Acc
Reads
Separate strands
Depth
ssaha2


Contig Contig_0000355
Template Size
Combined mapping quality
Filter
Y-Log scale
>>Acc
Reads
Separate strands
Depth
maq
Contig Contig=353
Template Size
Combined mapping quality
Filter
Y-Log scale
>>Acc
Reads
Separate strands
Depth
ssaha2



Genome Assembly - Normal Cell

Solexa reads:

Number of reads:	557 Million;
Finished genome size:	3.0 GB;
Read length:	2x75bp;
Estimated read coverage:	~25X;
Insert size:	190/50-300 bp;
Number of reads clustered:	458 Million

Assembly features: - contig stats

Total number of contigs:	1,020,346;
Total bases of contigs:	2.713 Gb
N50 contig size:	8,344;
Largest contig:	107,613
Averaged contig size:	2,659;
Contig coverage over the genome:	~90 %;
Mis-assembly errors:	?

Acknowledgements:

- *Jim Mullikin*
- *Yong Gu*
- *Hannes Ponstingl*
- *James Bonfield*
- *Heng Li*
- *Daniel Zerbino (EBI)*
- *Tony Cox*
- *Richard Durbin*

Solexa