

ILLUMINA

P

PIPELINE

by

ALEX

ILLUMINA

P

PIPELINE and

A

SSSEMBLY

OUTLINE:

project overview and dataflow

IGA output files

IGA quality scores

filtering for high quality reads

assembly parameters

visualization with Python Contig Viewer

Illumina GA2 high-throughput sequencing



Current performance:

10 to 20 million reads per lane

8 lanes per chip

single read - 85 to 125 nt

paired reads - 125 <-> 41+

80 to 160 million reads per slide

Conservative calculations: 100 M reads X 100 nt length = 10 Gb per run

(Illumina aiming at 90+ Gb per run - 10 Gb per lane - by the end of 2009)

Computational Facilities at the UCD Genome Center

Storage arrays - 48 Tb HD space

Computational servers with up to 128 and 512 Gb of RAM

Computational clusters with 100+ nodes

Illumina sequencing dataflow

**Illumina raw reads
on sequencing machine**

rsync/sftp
file transfer

**processed reads on
file server (storage array)**

QC and filtering for
high quality reads

**input files for downstream
assembly and analysis**



assembly using
CLC and/or Velvet

de novo assembled contigs



analysis and annotation
of assemblies

**data distribution via
sftp or web servers**

Project Goals



lettuce transcriptome sequencing

lettuce gene space sequencing

...

lettuce whole genome sequencing (~3 Mb)

Current Sequencing Scale Estimate

Cover and sequence transcriptome - 4 to 6 lanes sufficient
(1 lane by end of 2009 ?)

Gene space - 3 to 6 chips currently required
(0.5 chip - 4 lanes - by end of 2009 ?)

project overview and dataflow

IGA output files

IGA quality scores

filtering for high quality reads

assembly parameters

visualization with Python Contig Viewer

Figure 6 illustrates a typical Run Folder after IPAR image analysis and Pipeline base calling and alignment, or one containing only Pipeline analysis data.

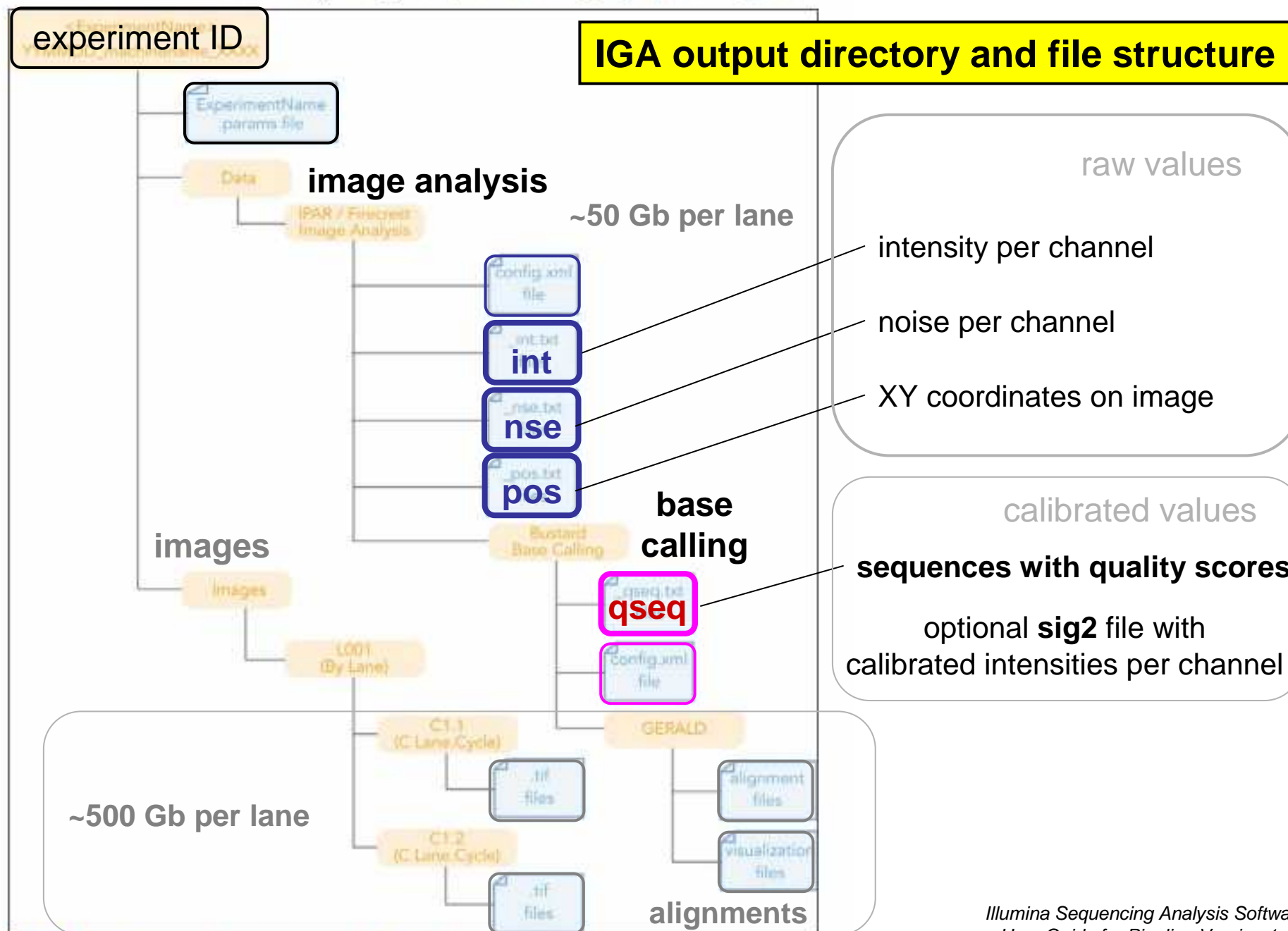


Figure 6 IPAR/Pipeline Run Folder Directory Structure

ILLUMINA Filtering Options for HQ reads

To remove the least reliable data from the analysis, the raw data can be filtered to remove any clusters that have “too much” intensity corresponding to bases other than the called base. By default, the purity of the signal from each cluster is examined over the first 25 cycles and chastity is calculated for each cycle:

$$\text{chastity} = \text{Highest_Intensity} / (\text{Highest_Intensity} + \text{Next_Highest_Intensity})$$

Chastity:

the ratio of the brightest intensity over the sum of the brightest and second brightest intensities

Filtering Options:

failed-chastity: the number of bases where the chastity is lower than the given threshold (1)

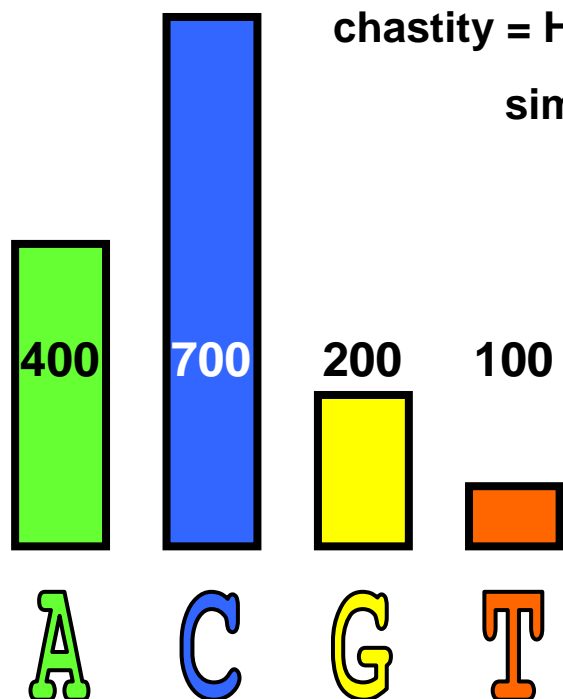
chastity: the minimum chastity (0.6)

purity: the minimum purity (25)

By default, the pipeline filters the clusters according to the relation "failed-chastity<=1", using a chastity threshold of 0.6, on the first 25 cycles. This removes all clusters with a chastity less than 0.6 on two or more bases among the first 25 bases.

```
--smt-filter failed-chastity --smt-relation le (less or equal)
--smt-threshold 1 --chastity-threshold 0.6
--pure-bases 25
```

[The new default filtering implemented in Bustard is that at most one cycle is less than the chastity threshold]



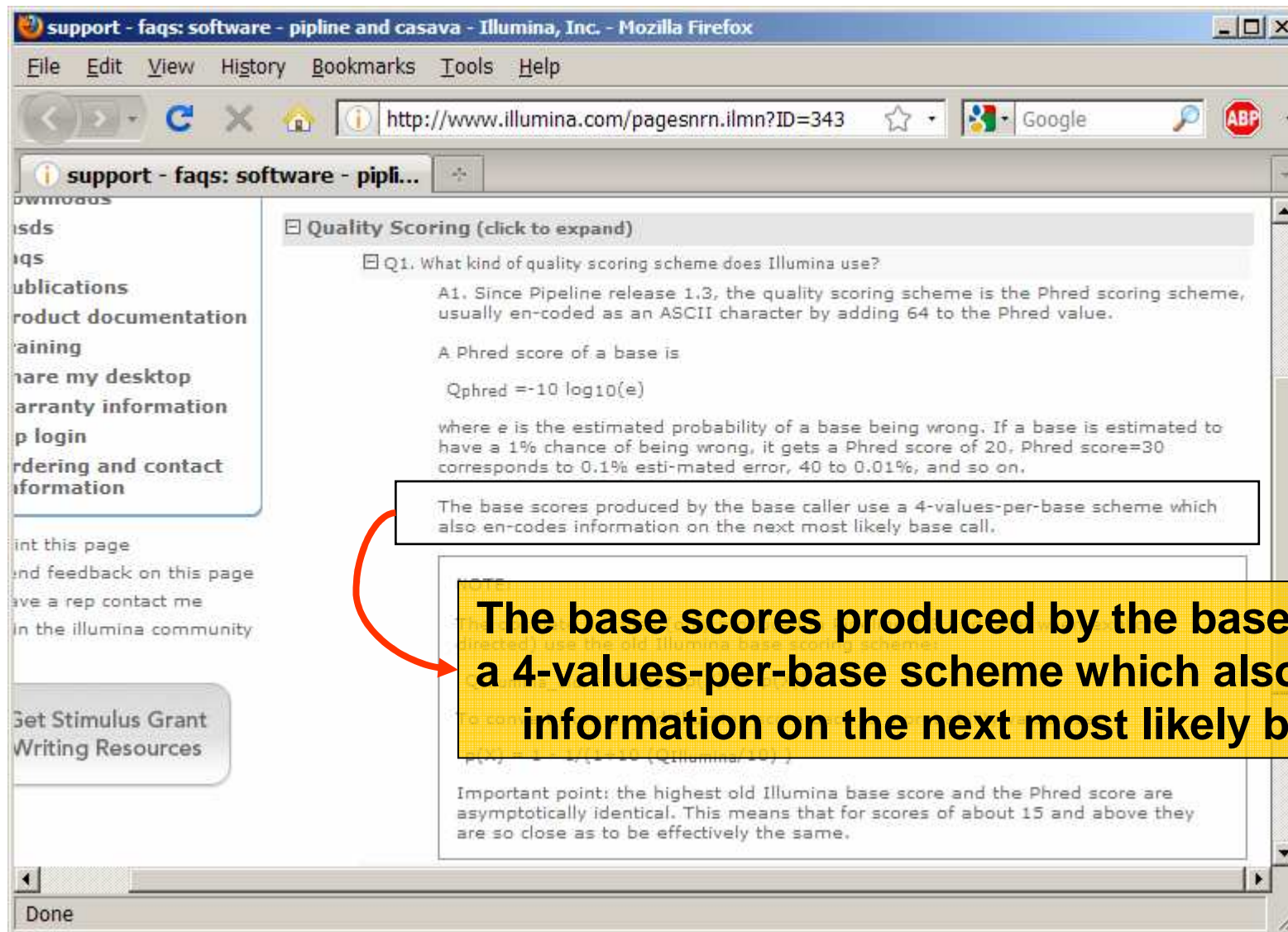
chastity = Highest_Intensity / (Highest_Intensity + Next_Highest_Intensity)

simple_ratio = Next_Highest_Intensity / Highest_Intensity

quality categories for simple ratio:

A	-	[0.0	<	Q	<=	0.2]
B	-	[0.2	<	Q	<=	0.4]
C	-	[0.4	<	Q	<=	0.6]
D	-	[0.6	<	Q	<=	0.8]
F	-	[0.8	<	Q	<=	1.0]

Intensity 1-st	1000	1000	1000	1000	1000	1000	1000
Intensity 2-nd	100	200	300	400	600	800	900
IGA chastity	0.91	0.83	0.77	0.71	0.63	0.56	0.53
(1.0 - ratio)	0.90	0.80	0.70	0.60	0.40	0.20	0.10
ratio	0.1	0.2	0.3	0.4	0.6	0.8	0.9
quality category	A (I)	A (I)	B (9)	B (9)	C (5)	D (0)	F (#)



A quality value Q is an integer mapping of p , the probability of the corresponding sequence letter being incorrect. It is calculated as follows:

$$Q_{phred} = -10 \log_{10} p$$

http://en.wikipedia.org/wiki/Phred_quality_score

Binary	Oct	Dec	Hex	Glyph
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1100	134	92	5C	\
101 1101	135	93	5D]
101 1110	136	94	5E	^
101 1111	137	95	5F	_
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b

http://en.wikipedia.org/wiki/FASTQ_format

A quality value Q is an integer mapping of p, the probability of the corresponding sequence letter being incorrect. It is calculated as follows:

$$Q_{\text{phred}} = -10 \log_{10} p$$

low quality

$$-10 \cdot \log_{10}(0.5) = -10 \cdot (-0.3) = 3$$

$$-10 \cdot \log_{10}(0.1) = -10 \cdot (-1) = 10$$

medium quality

$$-10 \cdot \log_{10}(0.01) = -10 \cdot (-2) = 20$$

<http://en.wikipedia.org/wiki/ASCII>

**Quality String - in symbolic ASCII format
(ASCII chracter code = quality value + 64)**

high quality

$$-10 \cdot \log_{10}(0.001) = -10 \cdot (-3) = 30$$

Illumina 1.3+ format can encode a Phred quality score from 0 to 62 using ASCII 64 to 126 (although in raw read data Phred scores from 0 to 40 only are expected).

The main output files in Bustard are the *_qseq files. They have the following format:

1. Machine name: (hopefully) unique identifier of the sequencer.
2. Run number: (hopefully) unique number to identify the run on the sequencer.
3. Lane number: positive integer (currently 1-8).
4. Tile number: positive integer.
5. X: x coordinate of the spot. Integer (can be negative).
6. Y: y coordinate of the spot. Integer (can be negative).
7. Index: positive integer. No indexing should have a value of 1.
8. Read Number: 1 for single reads; 1 or 2 for paired ends.
9. Sequence
10. Quality: the calibrated quality string.
11. Filter: Did the read pass filtering? 0 - No, 1 - Yes

```
===== qseq file =====
SOLEXA1 90707 2 lane 1 tile 6 88 0 1
TGTC AACAGAAAATTCAGAAATAGCTGACTTAAATCCTTTAGATTGTATTCCATTCGCATCGGCTAATTCAGAAATTTTATCTAG
ab_bbbb]_ ^`T`aab\ a`\ba]aaaaaa`a]S]a`aaa``aXaaa]_a`aa_____``^W^Q][^^^[[[ ^Y^] [][[]U[[YOX 1

SOLEXA1 90707 2 1 6 333 0 1
ATTCTTGATTGCCAAACGTTGATGATGGTGCCCAATTTTAGCGGGTGGTAAAGAATTTGGGGTCGCGGGTTGGTACGGGAGCCTT
aaaabaaQ^`Y__^J^`_Y`R[aG\YPIRTSGG]^_ \ZGXR`_XDTNWUBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB 0
=====
```

FASTA file:

```
>90707:SOLEXA1:2:1:6:88#0/1 891 LEN: 85
TGTC AACAGAAAATTCAGAAATAGCTGACTTAAATCCTTTAGATTGTATTCCATTCGCATCGGCTAATTCAGAAATTTTATCTAG
```

FASTQ file:

```
@90707:SOLEXA1:2:1:6:88#0/1
TGTC AACAGAAAATTCAGAAATAGCTGACTTAAATCCTTTAGATTGTATTCCATTCGCATCGGCTAATTCAGAAATTTTATCTAG
+90707:SOLEXA1:2:1:6:88#0/1
ab_bbbb]_ ^`T`aab\ a`\ba]aaaaaa`a]S]a`aaa``aXaaa]_a`aa_____``^W^Q][^^^[[[ ^Y^] [][[]U[[YOX
```

project overview and dataflow

IGA output files

IGA quality scores

filtering for high quality reads

assembly parameters

visualization with Python Contig Viewer

<http://code.google.com/p/atgc-illumina/>

The screenshot shows the 'atgc-illumina' project page on Google Code, viewed in Mozilla Firefox. The browser address bar shows the URL <http://code.google.com/p/atgc-illumina/source/browse/#svn/trunk>. The page header includes the project name 'atgc-illumina' with a logo and the tagline 'tools to process and manipulate Illumina sequences'. A search bar is present with the text 'Search projects'. Below the header, there are tabs for 'Project Home', 'Downloads', 'Wiki', 'Issues', 'Source', and 'Administer'. The 'Source' tab is active, showing a table of source files. The table has columns for 'Filename', 'Size', 'Rev', 'Date', and 'Author'. The 'trunk' directory is expanded, showing four files. Two red arrows point from the text annotations below to the 'tcl_illupa_rectificator_2009_05_19_Beta.tcl' and 'tcl_illupa_rectificator_2009_05_19_Beta.tcl' files. A red box highlights the last two files, 'tcl_sig2_base_caller_2009_05_14_Beta.tcl' and 'tcl_sig2_base_caller_2009_06_10_Beta.tcl', with a red arrow pointing from the text annotation below to the first of these files.

Filename	Size	Rev	Date	Author
tcl_illupa_rectificator_2009_05_19_Beta.tcl	10.0 KB	r6	Jun 10, 2009	akozik
tcl_illupa_rectificator_2009_05_19_Beta.tcl	10.0 KB	r6	Jun 10, 2009	akozik
tcl_sig2_base_caller_2009_05_14_Beta.tcl	12.6 KB	r5	Jun 09, 2009	akozik
tcl_sig2_base_caller_2009_06_10_Beta.tcl	12.7 KB	r7	Jul 15, 2009	akozik

sig2 basecalling / filtering (obsolete ?)

qseq parser - it takes into account IGA quality scores

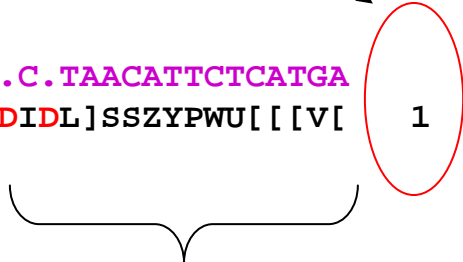
illupa rectificator - to transform raw intensities in vertical format into linear sig2 like order

qseq filtering approach summary:

qseq parser analyzes data from qseq files

filtering criteria '1'

```
SOLEXA1 90707      8      120      6      1512      0      1
AAAAACAGATGCTACTTTACTTTACTACCATATTTCCCAAACAAAGAGCTAACAGTAGAACATGACGC.C.TAACATTCTCATGA
a`aab_Z_V__aXaa\[aa`^a`[^]ab`aa^aa_a]SXZZ]`UXXS_Y`__^^S^^Y^S^GU]^TDIDL]SSZYPWU[[[V[ 1
```



sequence after first failed call is removed

>90707:SOLEXA1:8:120:6:1512#0/1 691 LEN: 68
AAAAACAGATGCTACTTTACTTTACTACCATATTTCCCAAACAAAGAGCTAACAGTAGAACATGACGC

future plans - to analyze string of quality scores regardless of IGA default filtering criteria

qseq parser output example - *.trim.fa file

```
bash-2.03$ tclsh8.3 tcl_solexa_qseq_parser_2009_07_16_Beta.tcl
Program usage:
qseq_file_to_process, output_file, tag_length, read_length
bash-2.03$
```

Without Tag:

```
>90707:SOLEXA1:8:120:6:1333#0/1    692  LEN: 68
GAATTCCTTCGCACATCATCATAAACCTCGGTTCTCCATTTGTCAAAAAGCAAACGGGGAATTTTTTG

>90707:SOLEXA1:8:120:6:1245#0/1    693  LEN: 68
GGGTGAATGTAAAGGAATATGAGGCTCTTCGACCTGACAAGCTTGCTGTGGATGGAGGCGAACAGAAT

>90707:SOLEXA1:8:120:6:236#0/1     694  LEN: 85
CGGCGATCTGATTCCGATGAGCAGTCAACCGCAAGCTCCACAGAGTAGGAGCCTGGTTATTACGGCTGGCGCAGGCGGAACCAAA

>90707:SOLEXA1:8:120:6:98#0/1     696  LEN: 85
ATCAGCTTTCCATGTAGCATAATAGTAACGCTGCCTAAGAAGTTGCAAATTCTCCCTGTCTAGAGAACTGGCTGAGAGCCTGGA
```

With Tag (bar code):

```
>PEFC42D7CAAXX:SOLEXA1:2:1:6:980#0/1  [ADPT: _GCTACAT_]    404  LEN: 54
ATAATAAAATACTTCATATCTATGTGTAAGATTAAAGGGACCATGCACTCACCT

>PEFC42D7CAAXX:SOLEXA1:2:1:6:509#0/1  [ADPT: _GCATAGT_]    405  LEN: 54
CATCATCTTGATCTTTTGCTCCACTTGGCCCTGCTCCATCATCTTGTGGATGCC

>PEFC42D7CAAXX:SOLEXA1:2:1:6:1127#0/1 [ADPT: _ACTAGCT_]    408  LEN: 54
TCCCTCCTTTAAACACCCCAAATTGCAATTAAACAATACTTGAGACTTTGGGC
```

```
#!/usr/bin/tcl
```

```
proc Process_Qseq {argv} {
```

```
    # set length_cut 60
    length_cut 40
```

```
    set f_in1  [open [lindex $argv 0] "r"]
    set f_out   [open [lindex $argv 1] "w"]
    ....
    set tag_1   [lindex $argv 2]
    set r_len   [lindex $argv 3]
```

sig2 file example

lane	tile	X coordinate	Y coordinate	calibrated intensity scores per channel											
				1-st cycle				2-nd cycle				3-d cycle			
				A	C	G	T	A	C	G	T	A	C	G	T
8	120	5	1105	-249	44	<u>523</u>	<u>315</u>	-248	<u>79</u>	25	<u>443</u>	-146	-138	<u>146</u>	<u>535</u>
8	120	5	942	-103	<u>164</u>	<u>486</u>	26	-25	<u>598</u>	-59	-34	-77	<u>564</u>	18	<u>43</u>
8	120	5	1392	-67	<u>97</u>	<u>501</u>	1	-101	<u>234</u>	-40	<u>154</u>	-128	-101	<u>306</u>	<u>238</u>
8	120	5	1114	<u>658</u>	<u>62</u>	61	52	<u>711</u>	<u>165</u>	43	-46	<u>646</u>	13	-1	<u>101</u>
8	120	5	1409	-124	<u>424</u>	<u>227</u>	-82	<u>548</u>	-92	-19	-99	<u>210</u>	25	<u>98</u>	26
8	120	5	327	-18	-69	<u>485</u>	-24	-39	<u>355</u>	-31	-72	-173	<u>450</u>	-14	-3
8	120	5	1017	-149	<u>587</u>	<u>144</u>	15	-259	<u>14</u>	-12	<u>697</u>	-67	<u>472</u>	1	<u>161</u>
8	120	5	413	<u>476</u>	-23	-44	<u>40</u>	-97	<u>88</u>	17	<u>429</u>	-95	<u>56</u>	-53	<u>493</u>
8	120	5	453	-99	<u>661</u>	-82	-37	<u>621</u>	<u>96</u>	9	-131	77	-252	<u>344</u>	<u>130</u>
					↑			↑						↑	
					C			A						G	

sig2 filtering approach summary:

sig2 basecaller analyzes calibrated peak intensities per channel

```
>SBO8_8_120_6_1512 | LINE: 692 [ FAIL: 69 ] :: LONG__Q: 24 :: CLEAN_LENGTH: 44  
[ ABCD_Q:56 F_QUAL:29 ] [ Q_FRACT:66 ]  
AAAAACAGATGCTACTTACTTTACTACCATATTTCCCAAACAANGNnCNAAcAGnANNAnAnNNCNC*N*NNnnNnnnnnNNnnN
```



low quality sequence after first failed call is removed

```
>SBO8_8_120_6_1512 | LINE: 692 [ LENGTH: 44 ] ( ____GC____: 13/44 )  
[ ABCD_Q:44 F_QUAL:0 ] [ Q_FRACT:100 ]  
AAAAACAGATGCTACTTACTTTACTACCATATTTCCCAAACAA
```

future plans - to find correlation between IGA quality scores and peak intensities

```
AAAAACAGATGCTACTTACTTTACTACCATATTTCCCAAACAANGNnCNAAcAGnANNAnAnNNCNC*N*NNnnNnnnnnNNnnN  
a`aab_Z_V__aXaa\[aa``^a`[^]ab`aa^aa_a]SXZZ]`UXXS_Y`__^^S^^Y^S^GU]^]TDIDL]SSZYPWU[[[V[
```

sig2 basecall / filtering


tcl_solexa_sig2_base_caller_2009_06_10_Beta.tcl - atgc-illumina - Project Hosting on Google Code - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://code.google.com/p/atgc-illumina/source/browse/trunk

Google

akozik@gmail.com | My favorites | Profile | Sign out

 **atgc-illumina**
tools to process and manipulate Illumina sequences

Search projects

Project Home Downloads Wiki Issues **Source** Administer

Checkout | Browse | Changes | Search Trunk | Request code review

Source path: svn/ trunk/ tcl_solexa_sig2_base_caller_2009_06_10_Beta.tcl < r6 r7

Show details

critical parameters

```
1 #!/usr/bin/tcl
2
3 proc Sig2_Table {argv} {
4
5     #### PARAMETERS ####
6     set upper_cut 180 ; # DONT CALL IF MAX VALUE BELOW THIS CUTOFF
7     set upper_dif 60 ; # ACCEPTABLE DIFF BETWEEN VALUES
8     set good_dif 100 ; # GOOD DIFF BETWEEN VALUES
9     set quality_upper_cut 1000 ; # QUALITY IN CAPITAL LETTERS ABOVE THIS VALUE
10    set quality_len_cut 20 ; # CUTOFF LENGTH OF ACCEPTABLE QUALITY SCORES
11    set good_length 24 ; # GOOD DNA LENGTH CUTOFF
12    set gc_upper 0.8 ; # UPPER GC CONTENT CUTOFF
13    set gc_lower 0.2 ; # LOWER GC CONTENT CUTOFF
14    set max_reads_debug 10000 ; # NUMBER OF SEQUENCES IN DEBUG FILE
15    set sig2_val_start 4 ; # COLUMN NUMBER WITH FIRST VALUES - QUADRO SET - COUNT FROM ZERO
16    ## END OF PARAM ##
17 }
```

Done

example of basecalling and filtering for sig2 files

	A	C	G	T		basecall			diff	ratio	qual
>SBO8_8_120_5_1105 COUNT: 636											
1:	A:-249	C:44	G:523	T:315	GOOD	BASE_CALL:->	G	FIRST_FAIL: -- MAX_DIFF: 208	RATIO: 0.60	QUAL: d	
2:	A:-248	C:79	G:25	T:443	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 364	RATIO: 0.17	QUAL: a	
3:	A:-146	C:-138	G:146	T:535	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 389	RATIO: 0.27	QUAL: b	
4:	A:-15	C:-177	G:159	T:372	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 213	RATIO: 0.42	QUAL: c	
5:	A:-241	C:-77	G:79	T:667	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 588	RATIO: 0.11	QUAL: a	
6:	A:2	C:-25	G:-10	T:558	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 556	RATIO: 0.00	QUAL: a	
7:	A:-54	C:-72	G:-1	T:561	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 562	RATIO: -0.00	QUAL: a	
8:	A:281	C:349	G:-13	T:-55	GOOD	BASE_CALL:->	c	FIRST_FAIL: -- MAX_DIFF: 68	RATIO: 0.81	QUAL: f	
9:	A:-67	C:-138	G:359	T:226	GOOD	BASE_CALL:->	G	FIRST_FAIL: -- MAX_DIFF: 133	RATIO: 0.62	QUAL: d	
10:	A:-160	C:-58	G:123	T:440	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 317	RATIO: 0.27	QUAL: b	
11:	A:-260	C:-21	G:6	T:668	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 662	RATIO: 0.00	QUAL: a	
12:	A:-17	C:-55	G:-31	T:613	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 630	RATIO: -0.02	QUAL: a	
13:	A:250	C:306	G:-35	T:-5	HOPE	BASE_CALL:->	x	FIRST_FAIL: -- MAX_DIFF: 56	RATIO: 0.81	QUAL: f	
14:	A:176	C:-13	G:-23	T:317	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 141	RATIO: 0.55	QUAL: c	
15:	A:318	C:7	G:32	T:17	GOOD	BASE_CALL:->	A	FIRST_FAIL: -- MAX_DIFF: 286	RATIO: 0.10	QUAL: a	
16:	A:194	C:303	G:69	T:-20	GOOD	BASE_CALL:->	C	FIRST_FAIL: -- MAX_DIFF: 109	RATIO: 0.64	QUAL: d	
17:	A:-86	C:13	G:-56	T:461	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 448	RATIO: 0.02	QUAL: a	
18:	A:353	C:-84	G:0	T:157	GOOD	BASE_CALL:->	A	FIRST_FAIL: -- MAX_DIFF: 196	RATIO: 0.44	QUAL: c	
19:	A:327	C:171	G:-19	T:54	GOOD	BASE_CALL:->	A	FIRST_FAIL: -- MAX_DIFF: 156	RATIO: 0.52	QUAL: c	
20:	A:149	C:-8	G:-18	T:288	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 139	RATIO: 0.51	QUAL: c	
21:	A:334	C:44	G:46	T:12	GOOD	BASE_CALL:->	A	FIRST_FAIL: -- MAX_DIFF: 288	RATIO: 0.13	QUAL: a	
22:	A:173	C:62	G:-11	T:218	HOPE	BASE_CALL:->	x	FIRST_FAIL: -- MAX_DIFF: 45	RATIO: 0.79	QUAL: d	
23:	A:130	C:-238	G:14	T:509	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 379	RATIO: 0.25	QUAL: b	
24:	A:89	C:80	G:-50	T:324	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 235	RATIO: 0.27	QUAL: b	
25:	A:-63	C:-70	G:361	T:74	GOOD	BASE_CALL:->	G	FIRST_FAIL: -- MAX_DIFF: 287	RATIO: 0.20	QUAL: b	
26:	A:41	C:-6	G:27	T:186	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 145	RATIO: 0.22	QUAL: b	
27:	A:86	C:36	G:-27	T:306	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 220	RATIO: 0.28	QUAL: b	
28:	A:-34	C:40	G:-11	T:346	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 306	RATIO: 0.11	QUAL: a	
29:	A:-147	C:80	G:-5	T:380	GOOD	BASE_CALL:->	T	FIRST_FAIL: -- MAX_DIFF: 300	RATIO: 0.21	QUAL: b	
30:	A:-101	C:-94	G:389	T:260	GOOD	BASE_CALL:->	G	FIRST_FAIL: -- MAX_DIFF: 129	RATIO: 0.66	QUAL: d	
31:	A:0	C:0	G:0	T:0	CRAP	BASE_CALL:->	*	FIRST_FAIL: 31 MAX_DIFF: 0	RATIO: X.XX	QUAL: x	
.....											
59:	A:74	C:54	G:115	T:83	NONE	BASE_CALL:->	n	FIRST_FAIL: 31 MAX_DIFF: 32	RATIO: 0.72	QUAL: x	
60:	A:173	C:-104	G:66	T:38	NONE	BASE_CALL:->	N	FIRST_FAIL: 31 MAX_DIFF: 107	RATIO: 0.38	QUAL: x	

>SBO8_8_120_5_1105 | LINE: 636 [FAIL: 31] :: SHORT_Q 22 ::
 CLEAN_LENGTH: 12 [ABCD_Q:56 F_QUAL:29] [Q_FRACT:66]
 GTTTTTTcGTTTtTACTAATAxTTGTTTTG*A*AGTGxgCCTCCCTxTGNGGATTCTTnN*GA*Gnn**t*G**nn**nNn*nnn

fasta output

sig2 basecalling and filtering - example of *.good.trim.fasta output file

```
>SB08_8_120_65_1105 | LINE: 9728 [ LENGTH: 44 ] ( ____GC____: 20/44 )
[ ABCD_Q:44 F_QUAL:0 ] [ Q_FRACT:100 ]
AACAAATGTCTCGAATCTTAACCGCACATCAAGCTGCAAGAGGG

>SB08_8_120_65_1326 | LINE: 9734 [ LENGTH: 46 ] ( ____GC____: 20/46 )
[ ABCD_Q:46 F_QUAL:0 ] [ Q_FRACT:100 ]
AGAACAAGGAGGATTATGCCAAGTTCTACGAGGCTTTCTCCAAGAA

>SB08_8_120_88_573 | LINE: 13230 [ LENGTH: 83 ] ( ____GC____: 22/83 )
[ ABCD_Q:83 F_QUAL:0 ] [ Q_FRACT:100 ]
CGAGAATATATATGAACAAGAGTTAGAACAAAAAGAAAGGATGAAAATGAAATCATAACAAAATTCTGAATCATTCTCTAACT

>SB08_8_120_88_899 | LINE: 13235 [ LENGTH: 63 ] ( ____GC____: 17/63 )
[ ABCD_Q:63 F_QUAL:0 ] [ Q_FRACT:100 ]
CTCATAAGTAATTCAAAGCTTTTTCAAACCTTCATTaCAAATAATTAACGTTTCATTTCATGG

>SB08_8_120_166_1586 | LINE: 25397 [ LENGTH: 48 ] ( LOWER_GC: 6/48 )
[ ABCD_Q:48 F_QUAL:0 ] [ Q_FRACT:100 ]
ATTCTGTTCACCTAATGGTTTATTTaTTTATTTATTTTTTTTTTTTATAA

>SB08_8_120_170_1139 | LINE: 26020 [ LENGTH: 30 ] ( UPPER_GC: 24/30 )
[ ABCD_Q:30 F_QUAL:0 ] [ Q_FRACT:100 ]
CGTCGATCCCCACCACGCCCGGCTCCGCCG
```

GC% content info

grep-able fasta header

sig2 basecalling and filtering of reads with adapter - example of *.good.trim.fastq

```
@SBN4_4_24_177_1635 _TCGTAT_ : 21
TTCAAGGATAACTTTTGCATATCGTAT
```

+

```
IIIII99IIIIIII9II599III909I9
```

```
@SBN4_4_24_178_1771 _TCGTAT_ : 19
TCCCAAATGTAGACAAAGCTCGTATGCCGTCTTCT
```

+

```
IIIIIIIII99I5IIIIII95959II95II599II99
```

```
@SBN4_4_24_179_1454 _NOT_FOUND_ : -1
AAATGAACTCAGCCGTAACGCGATTC
```

+

```
IIII9IIIIIII99I5II995I0I9II
```

I-40(73) 9-24(57) 5-20(53) 0-15(48) #-2(35)

Phred pseudo-quality scores (ASCII - 33)

```
@SBN4_4_24_181_1684 _NOT_FOUND_ : -1
AGTGGTCATGAGAAGAAAGATTGCT
```

+

```
I9II9IIII5I5II5IIII9II95II
```

```
@SBN4_4_24_182_1666 _TCGTAT_ : 23
GGGTGAGGGAGATAGATGGATTATCGTAT
```

+

```
999I9II59I9III9II95II99I959I9
```

```
@SBN4_4_24_183_1747 _TCGTAT_ : 24
AAGAGGGAAAAGGGCTATTAAGCTCGTAT
```

+

```
II9I9IIIIIII9999IIIIIII5III95II9
```

```
### TRIMMING WITH ADAPTER OPTIONS ###
set number_of_runs $cycle_last
### set trimming_with_adapter "FALSE"
set trimming_with_adapter "TRUE"
set adapter_seqs "TCGTAT"
set adapter_status "_NOT_FOUND_"
### END OF TRIMMING WITH ADAPTER ###
```

comment / uncomment options within source code

grep-able fasta header:

```
grep -A 1 " _TCGTAT_ "
```

[to extract all seqs with tag]

```
grep -A 1 " _TCGTAT_ : 21"
```

[to extract seqs with tag at pos 21]

illupa rectificator - re-format position, intensity, noise raw files into linear form

pos

-0.50	339.55
-0.47	29.30
-0.43	364.48
-0.37	653.60
-0.37	956.44
....	
1793.15	1684.18
1793.15	1756.25
1793.15	1790.75
1793.15	1850.09
1793.15	1911.18
1793.15	1955.35
1793.15	1982.12
1793.18	298.99
1793.46	1776.56
X	Y

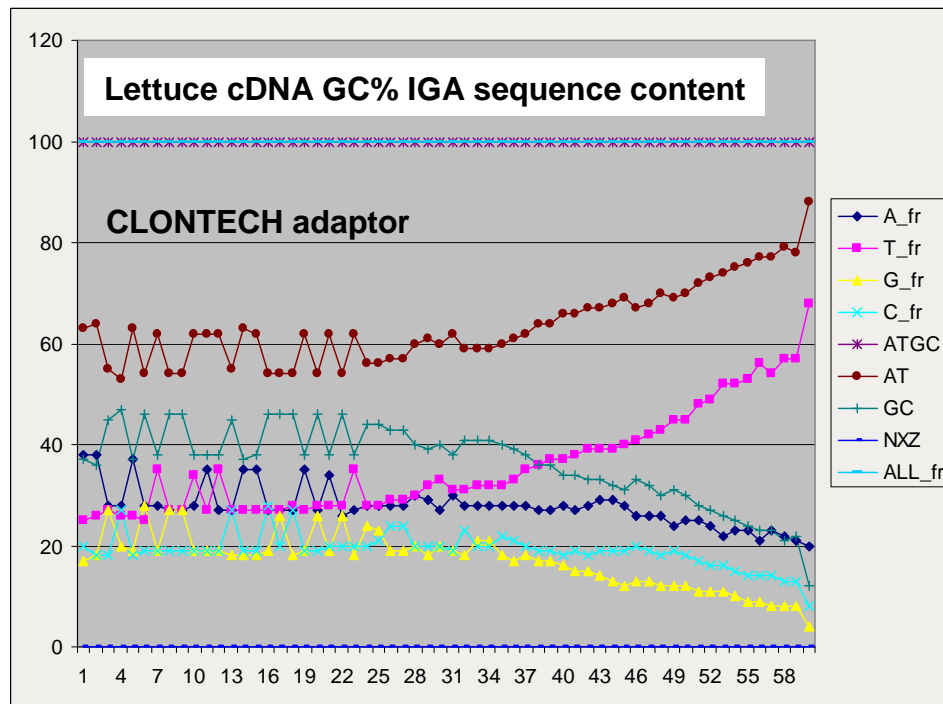
int

-25.0	-104.9	43.2	163.0
6.2	28.5	-145.9	502.6
-12.1	-16.1	-77.0	368.6
2.9	13.9	12.9	522.6
7.8	21.2	131.9	215.9
-1.2	9.3	233.4	342.9
-2.2	-4.3	-9.6	490.6
3.6	12.4	5.1	263.3
-1.8	7.4	2.0	82.2
2.9	-23.6	4.8	451.0
6.1	19.9	-2.5	511.6
5.8	14.7	60.9	140.3
0.8	10.2	-33.7	39.9
151.3	584.0	-199.8	-74.3
188.4	623.9	0.0	-49.1
#END CYCLE 1			

nse

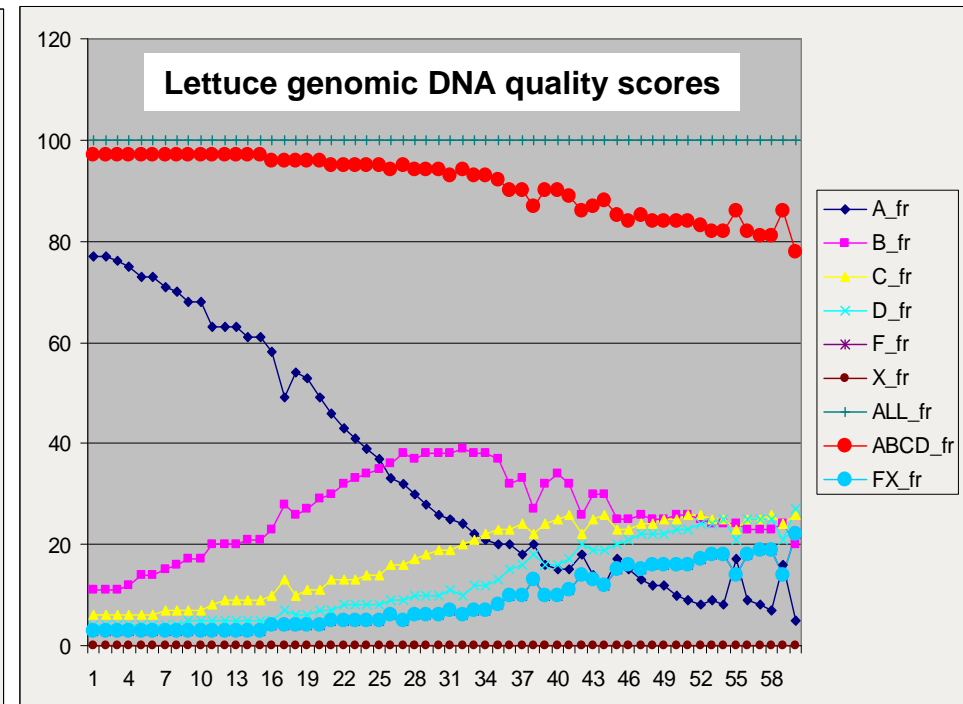
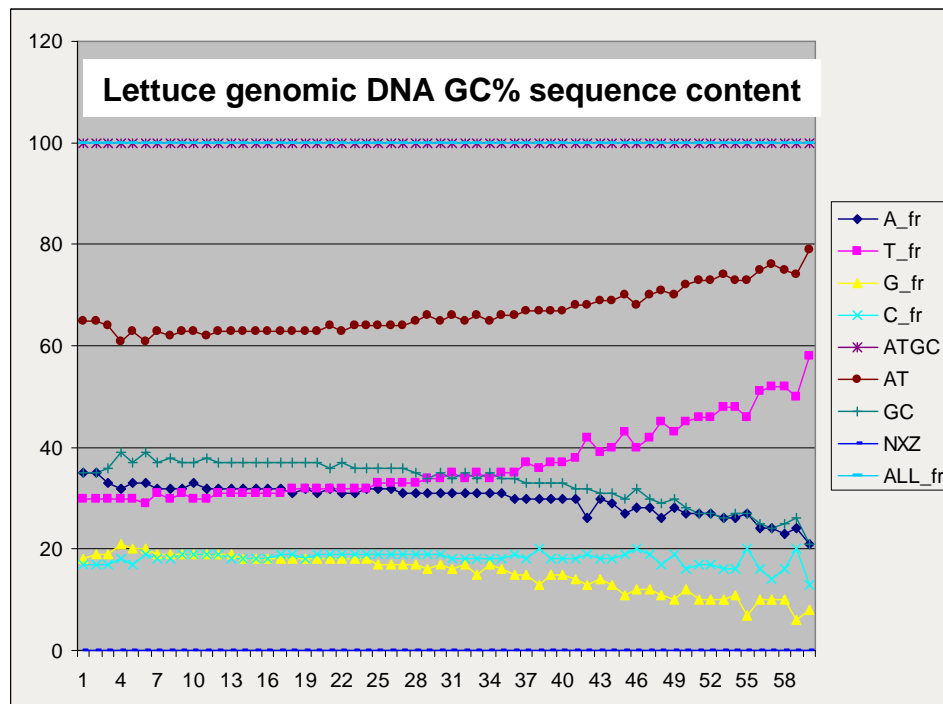
3.1	5.7	2.6	3.3
3.0	5.8	2.8	5.5
3.0	5.8	2.8	5.5
3.0	5.8	2.8	5.5
3.4	6.3	3.0	3.4
3.4	6.3	3.0	3.4
2.8	5.1	2.5	5.3
2.8	5.1	2.5	5.3
2.8	5.1	2.5	5.3
3.2	5.7	2.5	4.4
3.2	5.7	2.5	4.4
2.6	5.2	2.4	3.8
2.6	5.2	2.4	3.8
3.2	6.4	2.8	4.0
2.8	5.1	0.0	5.3
#END CYCLE 1			

line	tile	X	Y	cycle 1				cycle 2				cycle 3				cycle 4			
				A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T
3	0012	108.61	1169.14	-10	20	22	630	9	21	894	479	13	25	756	411	182	539	2	20
3	0012	108.63	1586.81	570	685	1	9	43	37	770	545	38	45	540	394	770	529	-7	15
3	0012	108.67	1305.63	127	503	2	14	11	104	19	670	37	57	703	442	121	13	766	462
3	0012	108.72	1600.48	-98	-105	6	609	-56	528	-32	-18	676	489	-15	-32	651	407	-57	16
3	0012	108.75	1644.07	165	582	2	24	39	54	20	620	208	657	5	-13	868	574	4	-18
3	0012	108.75	1357.78	191	665	1	25	9	26	21	639	182	683	4	24	672	503	13	21
3	0012	108.76	1894.18	-26	-9	445	540	12	12	543	354	794	580	4	17	464	452	5	5
3	0012	108.78	1694.68	175	611	-7	1	-16	-29	18	569	187	633	-3	9	271	581	-16	
3	0012	108.82	1556.78	893	519	-4	18	182	545	4	3	15	17	614	343	555	374	8	12
3	0012	108.83	1365.97	148	645	2	27	908	740	3	8	773	635	6	25	694	538	2	24
3	0012	108.88	875.17	10	35	650	362	145	509	2	-43	135	453	0	21	464	364	3	14

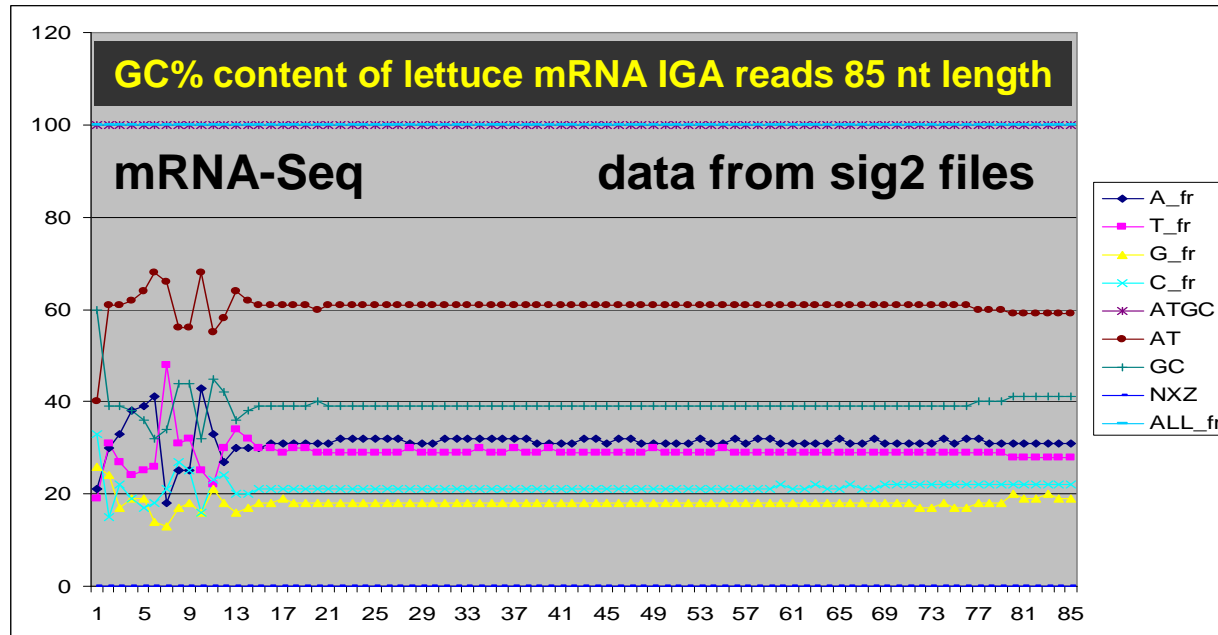


QC for GC% content and quality scores

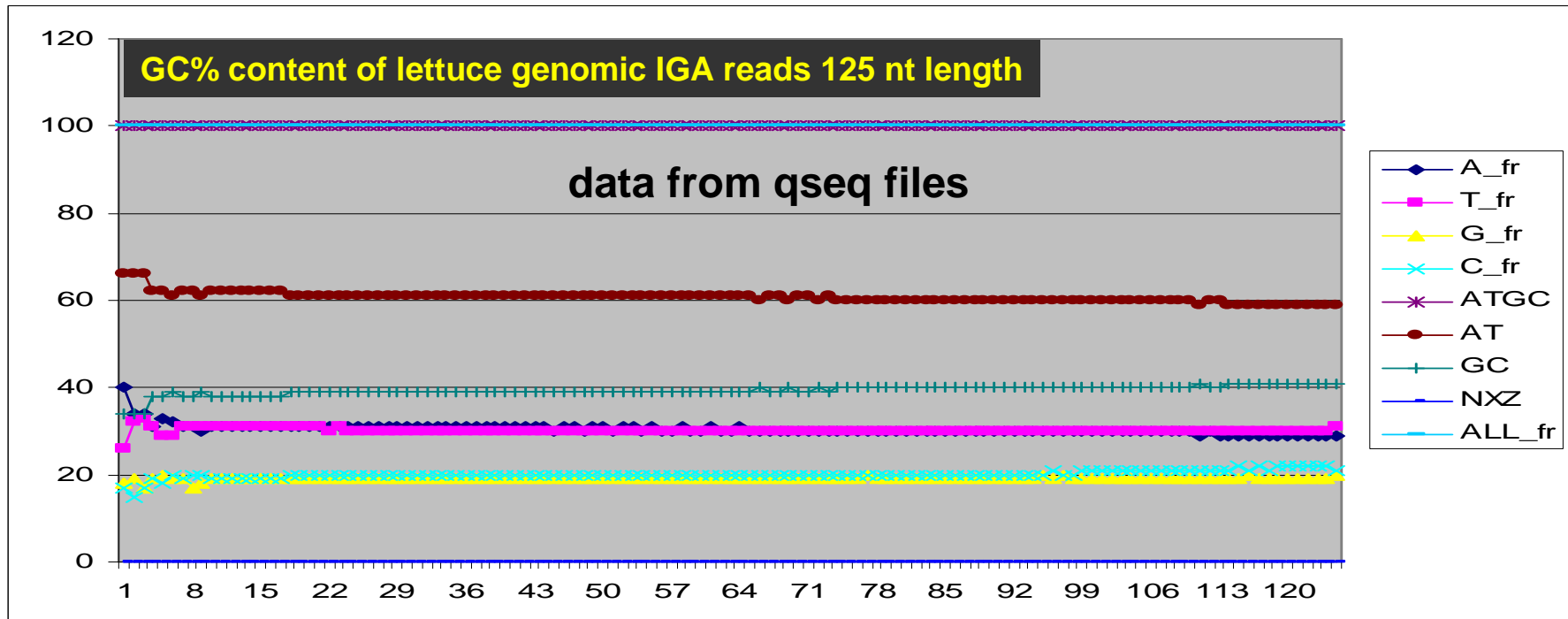
OLD CHEMISTRY

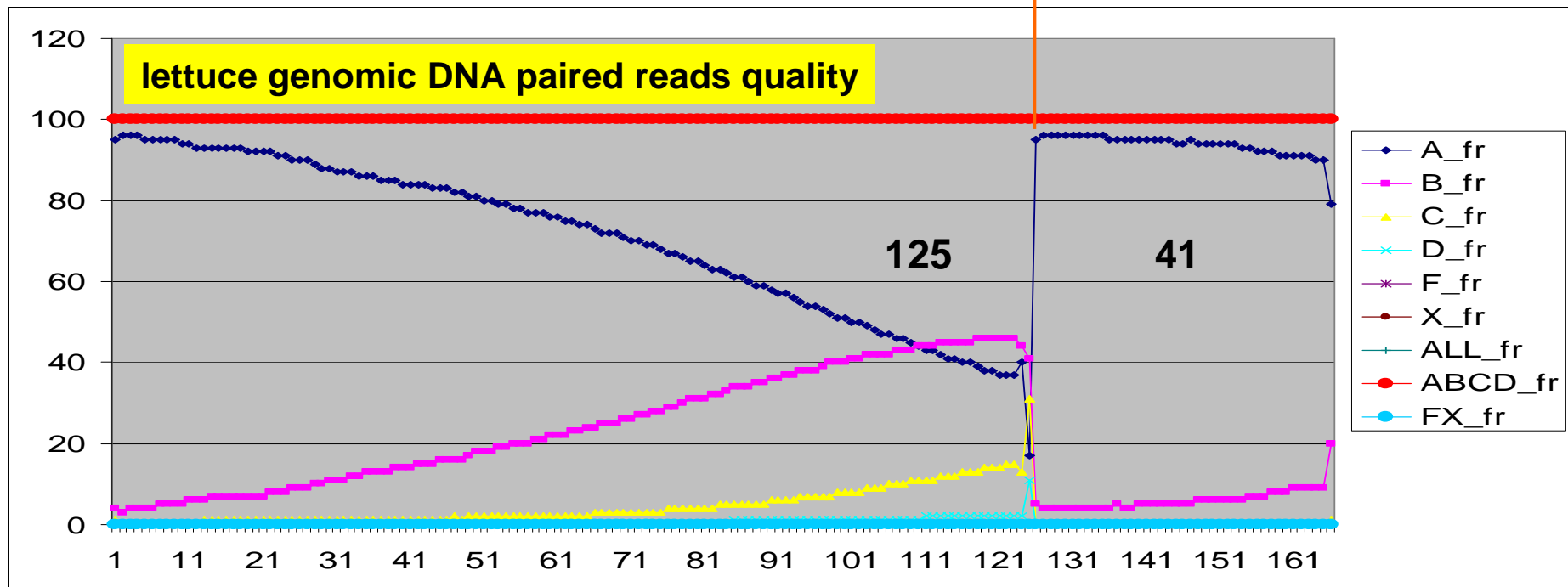
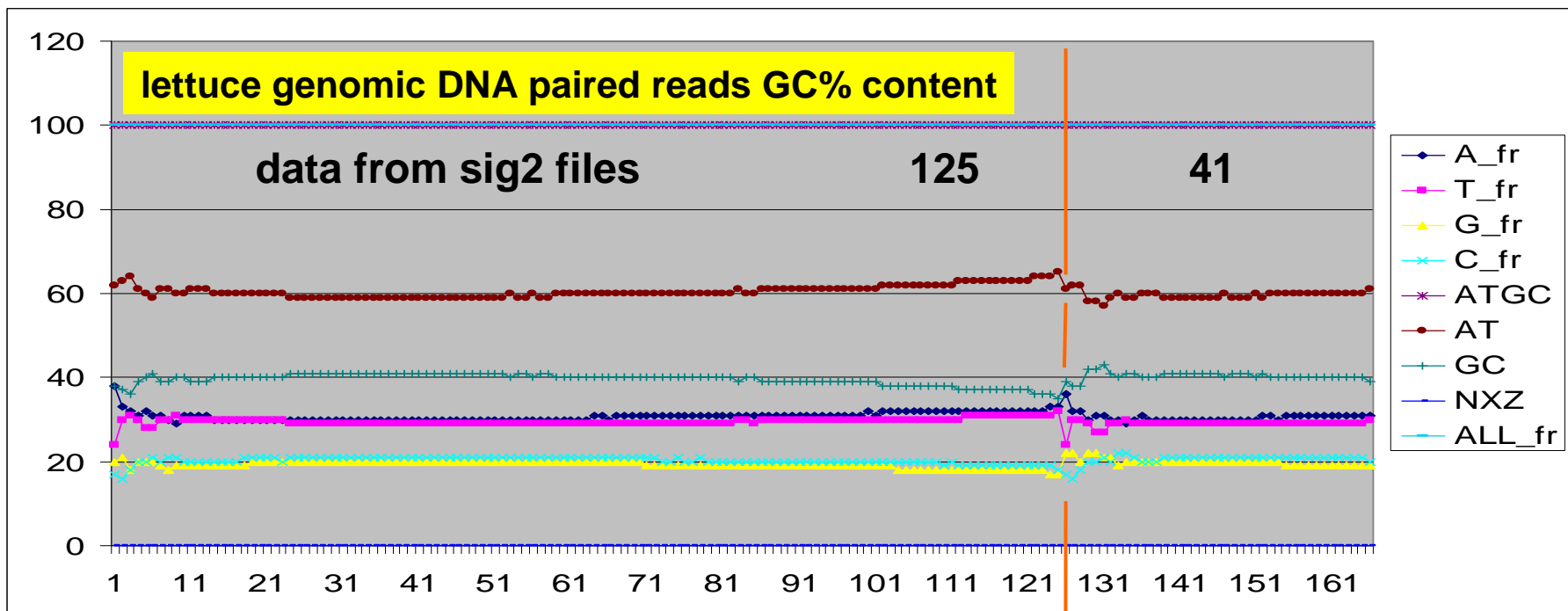


QC examples for GC% content



NEW CHEMISTRY





project overview and dataflow

IGA output files

IGA quality scores

filtering for high quality reads

assembly parameters

visualization with Python Contig Viewer

de novo assembly software -

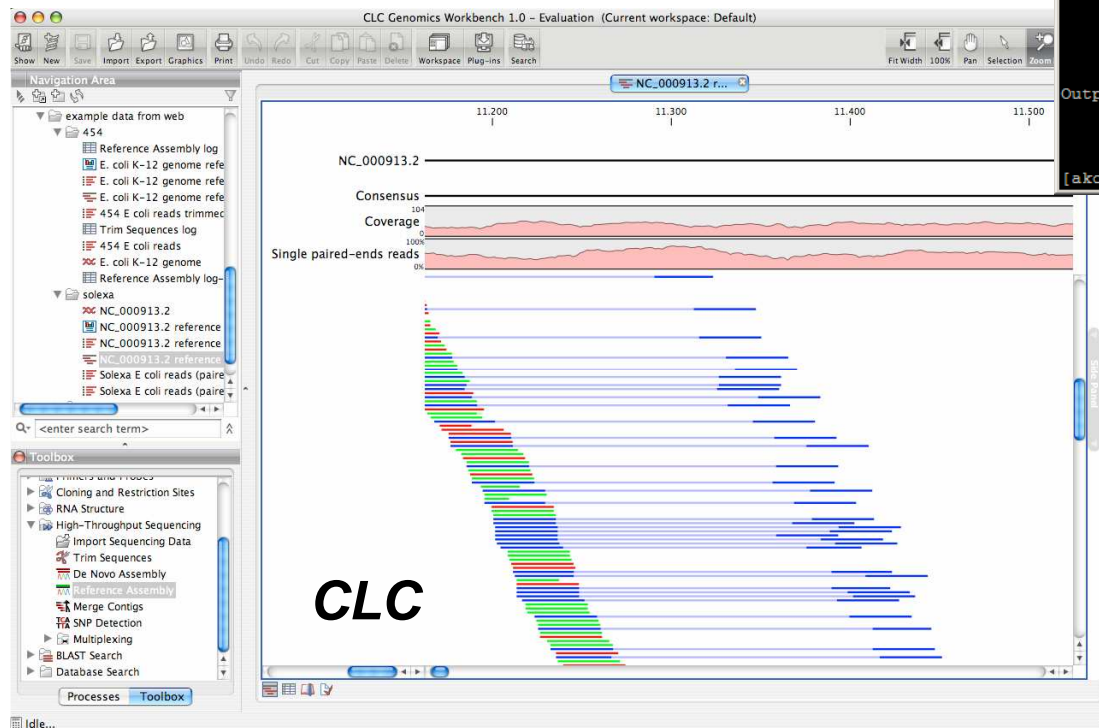
Velvet - publicly available

<http://www.ebi.ac.uk/~zerbino/velvet/>

CLC Genomics workbench - commercial

<http://www.clcbio.com/>

different algorithms - different assemblies



```
akozik@aristotle:~/_velvet_0.7.49
[akozik@aristotle _velvet_0.7.49]$ ./velveth
velveth - simple hashing program
Version 0.7.49

Copyright 2007, 2008 Daniel Zerbino (zerbino@ebi.ac.uk)
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Compilation settings:
CATEGORIES = 2
MAXKMERLENGTH = 31

Usage:
./velveth directory hash_length {[-file_format][-read_type] filename}

        directory      : directory name for output files
        hash length     : odd integer (if even, it will be decremented)
        <= 31 (if above, will be reduced)
        filename        : path to sequence file or - for standard input

File format options:
-fastq
-fastq.gz
-eland
-gerald

Read type options:
-short
-shortPaired
-short2
-shortPaired2
-long
-longPaired

Output:
directory/Roadmaps
directory/Sequences
[Both files are picked up by graph, so please leave them there]
[akozik@aristotle _velvet_0.7.49]$
```

Velvet

Velvet -

- large input 120+ million reads
- low number of chimeric contigs

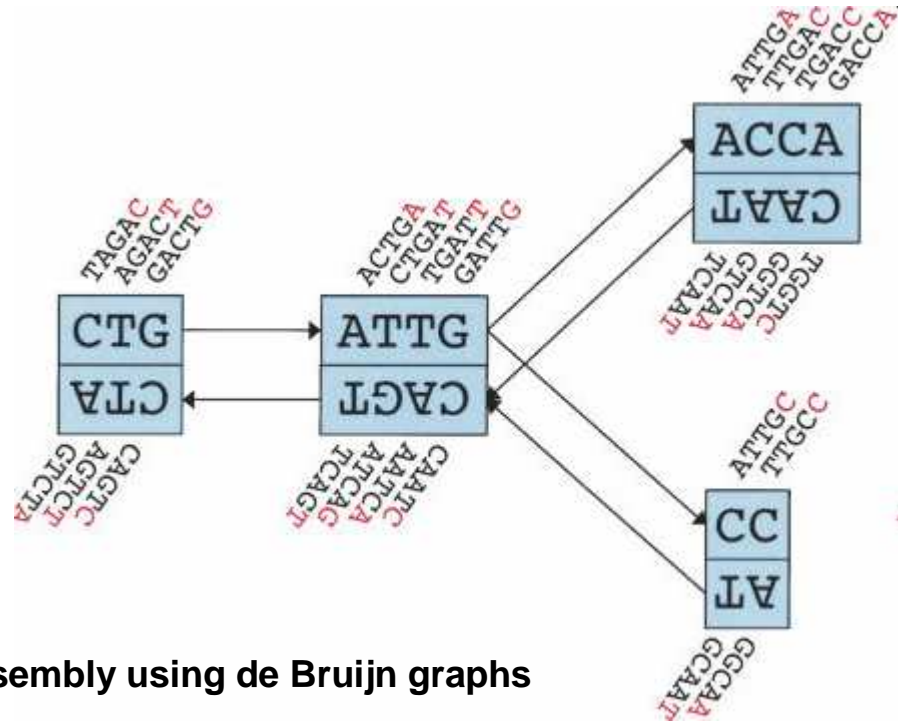
CLC -

- 20 million reads input limit
- higher number of chimeras

CLC contigs usually longer than Velvet contigs

Velvet

critical parameter K-mer value



Velvet: Algorithms for de novo short read assembly using de Bruijn graphs Daniel R. Zerbino and Ewan Birney

Figure 1.

Schematic representation of our implementation of the de Bruijn graph. Each node, represented by a single rectangle, represents a series of overlapping k-mers (in this case, $k = 5$), listed directly above or below. (Red) The last nucleotide of each k-mer. The sequence of those final nucleotides, copied in large letters in the rectangle, is the sequence of the node. The twin node, directly attached to the node, either below or above, represents the reverse series of reverse complement k-mers. Arcs are represented as arrows between nodes. The last k-mer of an arc's origin overlaps with the first of its destination. Each arc has a symmetric arc. Note that the two nodes on the left could be merged into one without loss of information, because they form a chain.

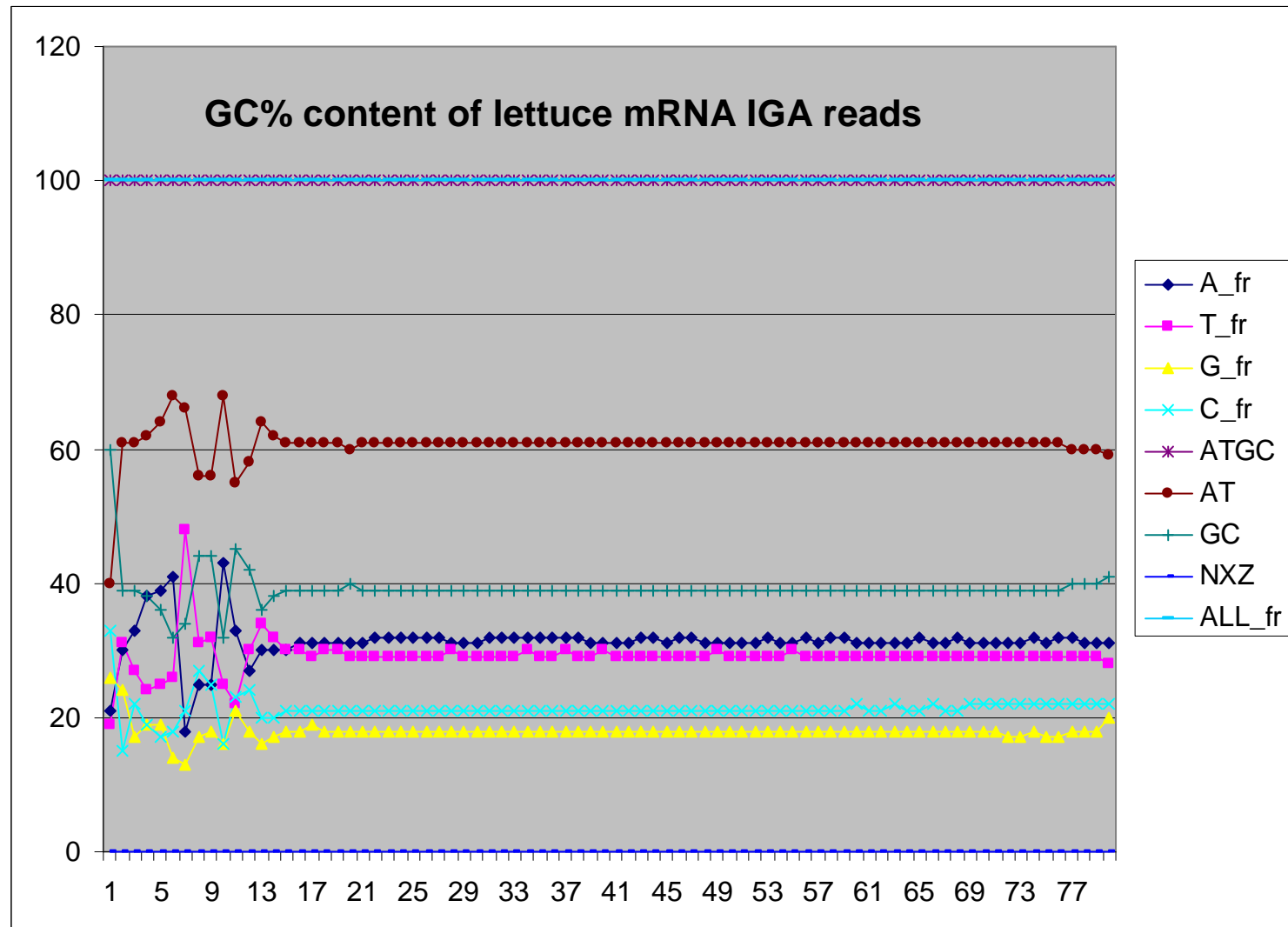
Genome Res. 2008 May; 18(5): 821–829.

doi: 10.1101/gr.074492.107.

Copyright © 2008, Cold Spring Harbor Laboratory Press

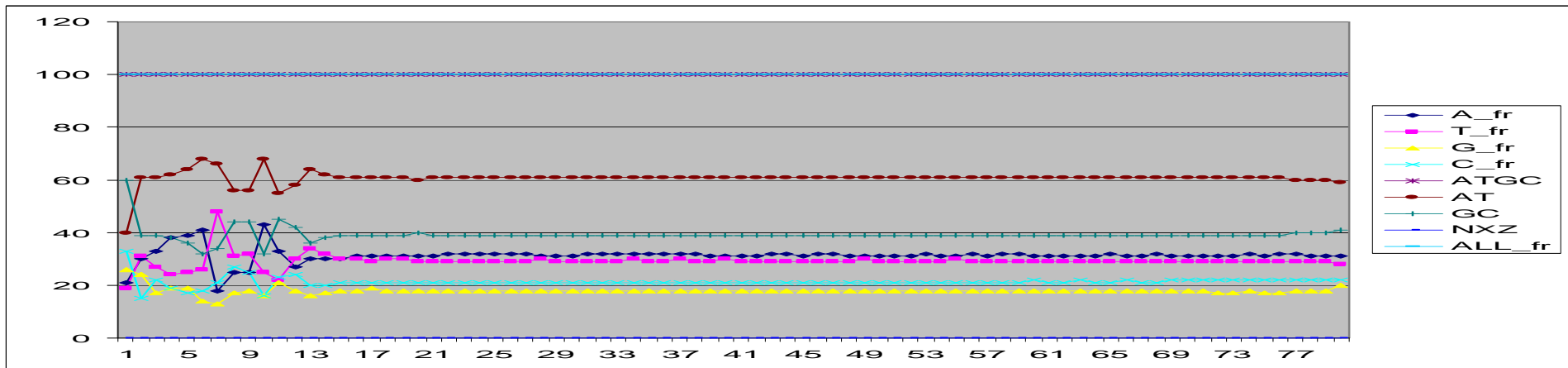
<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2336801>

case study - assembly of 22.62 million mRNA reads



QC trimming and assembly with 22.62 million reads

four subsets: 80 nt, 70 nt, 65 nt and 60 nt



```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_80.fa
>SB08_8_120_1651_1715
CAACAAGGCAAACCATCTTGGACAAAACGTTTGTGCTTGATGACAAACTATTATGCAAAAAGATCCGGGAAGAAGCAGA
```

```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_70_A.fa
>SB08_8_120_1651_1715
AACCATCTTGGACAAAACGTTTGTGCTTGATGACAAACTATTATGCAAAAAGATCCGGGAAGAAGCAGA
```

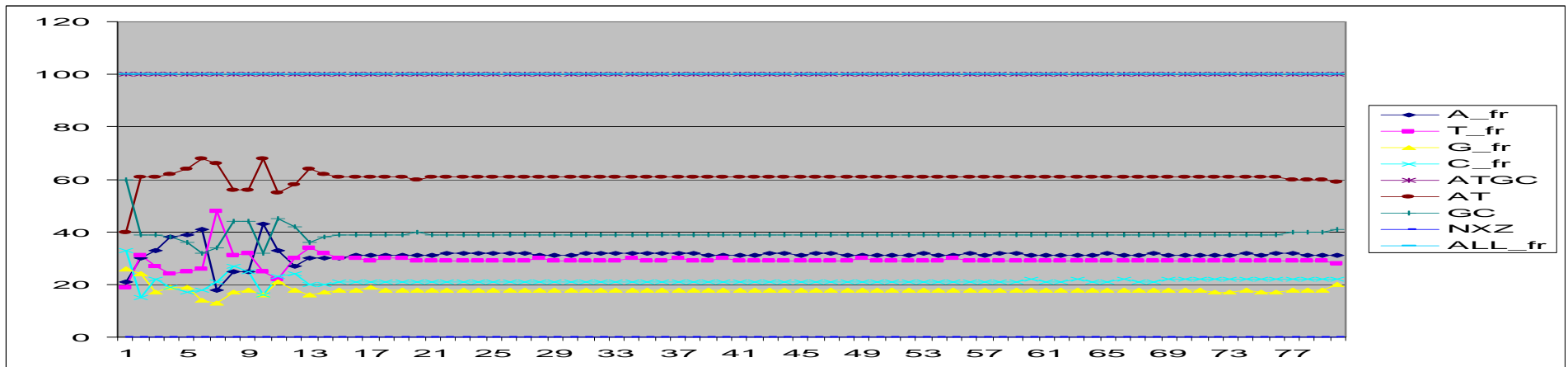
A

```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_70_D.fa
>SB08_8_120_1651_1715
CAACAAGGCAAACCATCTTGGACAAAACGTTTGTGCTTGATGACAAACTATTATGCAAAAAGATCCGGG
```

D

set 'A' - trim left part of original sequence
set 'D' - trim right part of original sequence

10 nt trimming



```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_80.fa
>SB08_8_120_1651_1715
CAACAAGGCAAACCATCTTGGACAAAACGTTTGTGCTTGATGACAACTATTATGCAAAAAGATCCGGGAAGAAGCAGA
```

```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_65_A.fa
>SB08_8_120_1651_1715
TCTTGGACAAAACGTTTGTGCTTGATGACAACTATTATGCAAAAAGATCCGGGAAGAAGCAGA
```

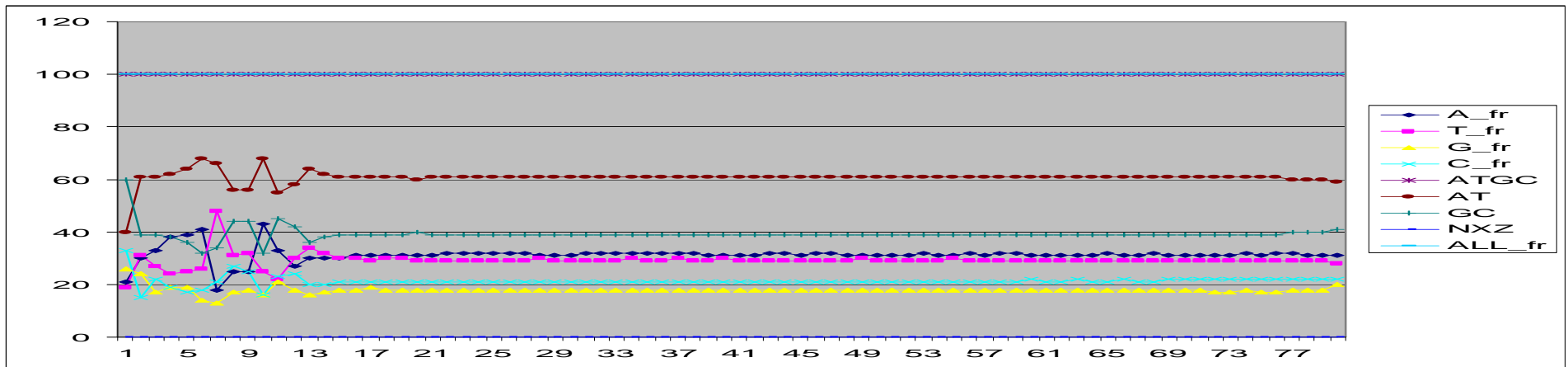
A

```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_65_B.fa
>SB08_8_120_1651_1715
AACCATCTTGGACAAAACGTTTGTGCTTGATGACAACTATTATGCAAAAAGATCCGGGAAGAA
```

C

```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_65_D.fa
>SB08_8_120_1651_1715
CAACAAGGCAAACCATCTTGGACAAAACGTTTGTGCTTGATGACAACTATTATGCAAAAAGAT
```

D 15 nt trimming



```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_80.fa
>SB08_8_120_1651_1715
CAACAAGGCAAACCATCTTGGACAAAACGTTTGGTTGCTTGATGACAAACTATTATGCAAAAAGATCCGGGAAGAAGCAGA
```

```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_60_A.fa
>SB08_8_120_1651_1715
GACAAAACGTTTGGTTGCTTGATGACAAACTATTATGCAAAAAGATCCGGGAAGAAGCAGA
```

A

```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_60_B.fa
>SB08_8_120_1651_1715
TCTTGGACAAAACGTTTGGTTGCTTGATGACAAACTATTATGCAAAAAGATCCGGGAAGAA
```

B

```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_60_C.fa
>SB08_8_120_1651_1715
AACCATCTTGGACAAAACGTTTGGTTGCTTGATGACAAACTATTATGCAAAAAGATCCGGG
```

C

```
bash-2.03$ tail -2 090707_SOLEXA1_FC42EK3AAXX_lettuce_cDNA.Clean.GC.Q_60_D.fa
>SB08_8_120_1651_1715
CAACAAGGCAAACCATCTTGGACAAAACGTTTGGTTGCTTGATGACAAACTATTATGCAAA
```

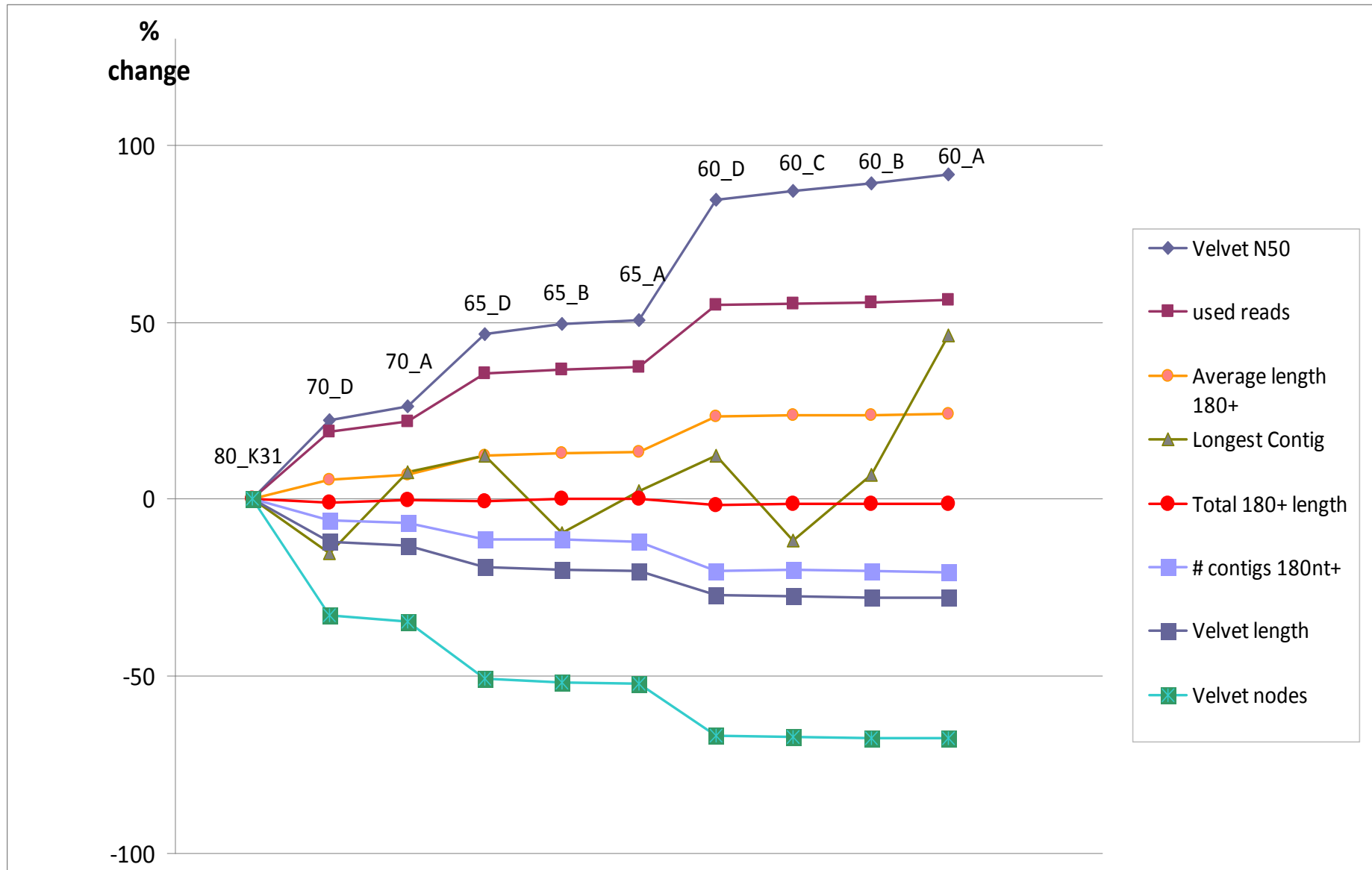
D

20 nt trimming

Trimming effect on assemblies (transcriptome)

sample	# of reads (million)	used reads (million)	velvet N50	velvet # nodes	velvet length million nt	longest	median 180 nt +	average 180 nt +	contigs 180nt +	total 180+ length million nt
80_K27	22.62	8.91	126	968,002	53.08	2428	260	312	79,889	24.94
80_K29	22.62	9.78	141	829,641	50.38	2448	262	318	82,619	26.28
80_K31	22.62	10.73	156	719,028	48.97	4417	267	326	85,123	27.76
70_D	22.62	12.78	191	483,450	43.01	3752	278	344	80,045	27.53
70_A	22.62	13.08	197	469,949	42.62	4752	279	349	79,578	27.74
65_D	22.62	14.53	229	353,683	39.51	4964	290	366	75,602	27.64
65_B	22.62	14.68	233	347,461	39.28	3995	289	368	75,387	27.76
65_A	22.62	14.75	235	343,477	39.11	4523	290	370	75,038	27.76
60_D	22.62	16.61	288	239,076	35.71	4964	307	402	67,942	27.29
60_C	22.62	16.67	292	237,207	35.57	3900	305	403	68,112	27.41
60_B	22.62	16.71	295	235,140	35.44	4720	306	404	67,836	27.41
60_A	22.62	16.76	299	233,365	35.33	6459	306	405	67,744	27.43

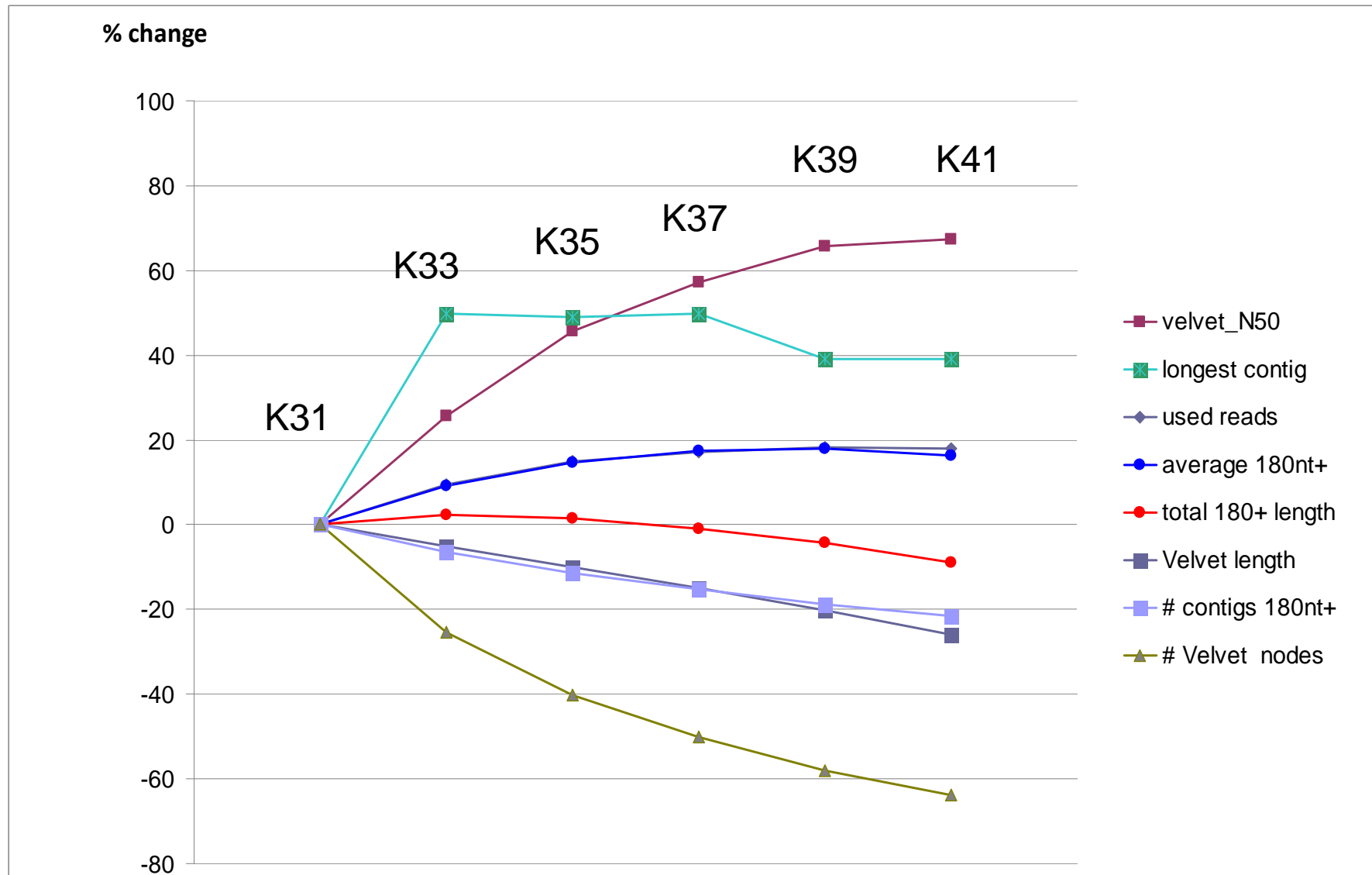
Trimming effect on assemblies (transcriptome)



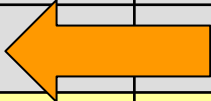
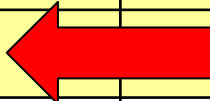

Velvet K-mer effect on assemblies (transcriptome)

sample	# of reads (million)	used reads (million)	velvet N50	velvet # nodes	velvet length million nt	longest	median 180 nt +	average 180 nt +	contigs 180nt +	total 180+ length million nt
80_QC K31	47.50	17.03	126	1,160,217	60.85	4272	261	318	93,175	29.67
60_QC K31	47.50	33.44	303	325,378	42.83	5131	313	422	76,994	32.55
60_QC K33	47.50	36.64	381	242,908	40.68	7682	328	461	72,092	33.26
60_QC K35	47.50	38.39	441	194,847	38.58	7640	338	484	68,241	33.03
60_QC K37	47.50	39.17	476	162,163	36.46	7682	342	495	65,230	32.26
60_QC K39	47.50	39.58	502	136,282	34.20	7141	343	498	62,557	31.16
60_QC K41	47.50	39.47	507	117,446	31.76	7141	338	491	60,446	29.68

Velvet K-mer effect on assemblies (transcriptome):
same input 47.50 million reads of 60 nt length - different K-mer values



Velvet K-mer effect on assemblies (gene space)

sample	# of reads (million)	used reads (million)	velvet N50	velvet # nodes	velvet length million nt	longest	median 180 nt +	average 180 nt +	contigs 180nt +	total 180+ length million nt
050_K31	55.41	NA	71	2,412,943	52.68	5610				
075_K31	55.41	NA	86	15,931,390	433.43	3680				
100_K31	55.41	NA	83	28,268,865	723.08	3906				
125_K31	55.41	NA	80	37,395,673	890.06	4247				
050_K37	55.41	2.20	78	378,970	8.56	21703				
075_K37	55.41	9.58	110	7,368,125	312.01	4386				
100_K37	55.41	17.83	122	17,113,337	654.16	3520				
125_K37	55.41	21.00	118	24,634,813	872.36	4769				
125_K47	55.41	26.78	189	11,692,947	753.67	6265				
125_K53	55.41	28.11	217	7,359,161	662.23	4809				

assembly of the assemblies

1 step

Velvet and/or CLC with different parameters and samples

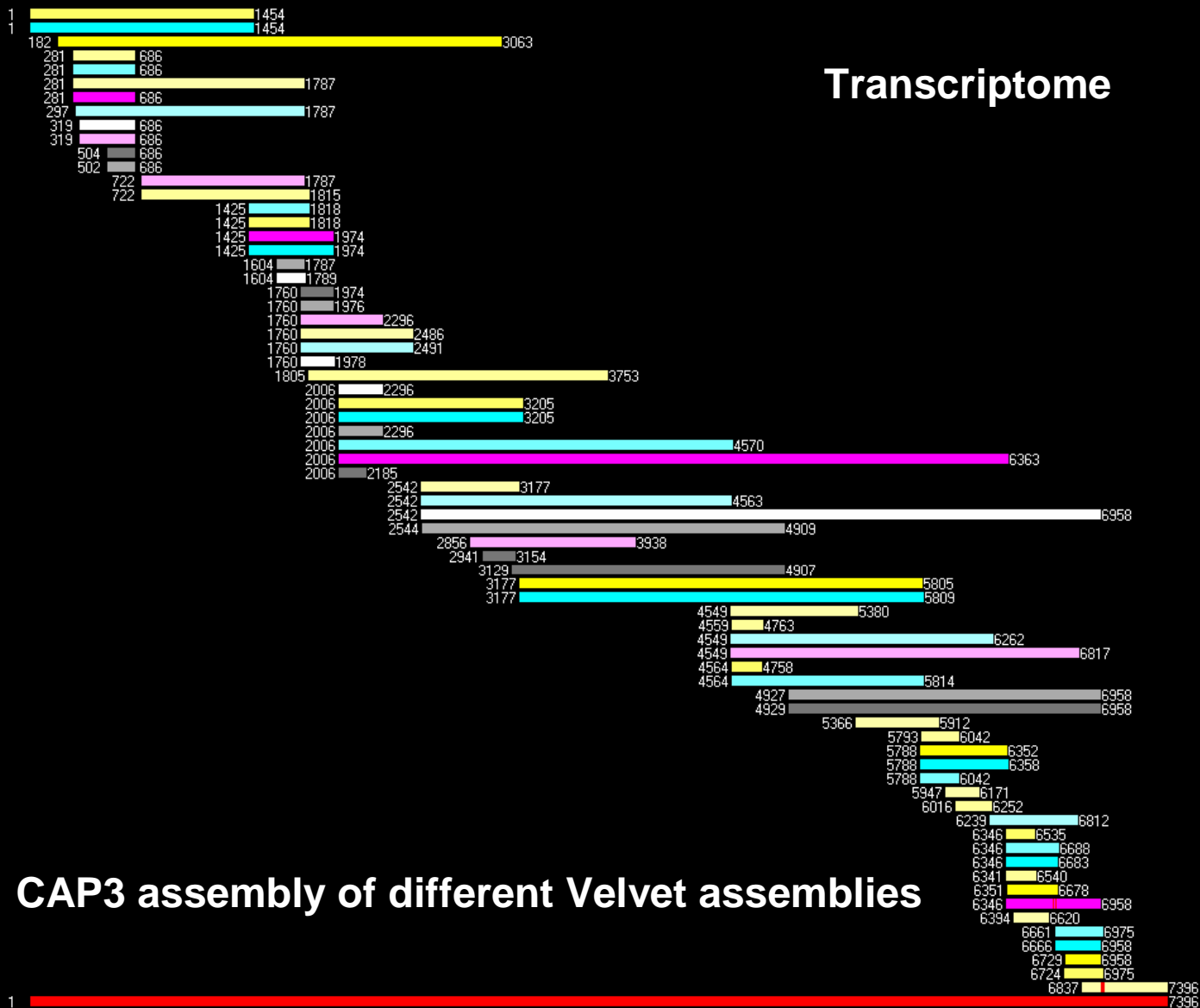
2 step

CAP3 assembly with Velvet and/or CLC contigs

3 step

Visualization and validation

Transcriptome



SBOX_V07_399
SBOX_V07_398
SBOX_V07_397
SBOX_V07_396
SBOX_V07_395
SBOX_V07_394
SBOX_V07_393
SBOX_V07_392
SBOX_V07_391
SBOX_V07_390
SBOX_V07_389
SBOX_V07_388
SBOX_V07_387
SBOX_V07_386
SBOX_V07_385
SBOX_V07_384
SBOX_V07_383
SBOX_V07_382
SBOX_V07_381
SBOX_V07_380
SBOX_V07_379
SBOX_V07_378
SBOX_V07_377
SBOX_V07_376
SBOX_V07_375
SBOX_V07_374
SBOX_V07_373
SBOX_V07_372
SBOX_V07_371
SBOX_V07_370
SBOX_V07_369
SBOX_V07_368
SBOX_V07_367
SBOX_V07_366
SBOX_V07_365
SBOX_V07_364
SBOX_V07_363
SBOX_V07_362
SBOX_V07_361
SBOX_V07_360
SBOX_V07_359
SBOX_V07_358
SBOX_V07_357
SBOX_V07_356
SBOX_V07_355
SBOX_V07_354
SBOX_V07_353
SBOX_V07_352
SBOX_V07_351
SBOX_V07_350
SBOX_V07_349
SBOX_V07_348
SBOX_V07_347
SBOX_V07_346
SBOX_V07_345
SBOX_V07_344
SBOX_V07_343
SBOX_V07_342
SBOX_V07_341
SBOX_V07_340
SBOX_V07_339
SBOX_V07_338
SBOX_V07_337
SBOX_V07_336
SBOX_V07_335
SBOX_V07_334
SBOX_V07_333
SBOX_V07_332
SBOX_V07_331
SBOX_V07_330
SBOX_V07_329
SBOX_V07_328
SBOX_V07_327
SBOX_V07_326
SBOX_V07_325
SBOX_V07_324
SBOX_V07_323
SBOX_V07_322
SBOX_V07_321
SBOX_V07_320
SBOX_V07_319
SBOX_V07_318
SBOX_V07_317
SBOX_V07_316
SBOX_V07_315
SBOX_V07_314
SBOX_V07_313
SBOX_V07_312
SBOX_V07_311
SBOX_V07_310
SBOX_V07_309
SBOX_V07_308
SBOX_V07_307
SBOX_V07_306
SBOX_V07_305
SBOX_V07_304
SBOX_V07_303
SBOX_V07_302
SBOX_V07_301
SBOX_V07_300
SBOX_V07_299
SBOX_V07_298
SBOX_V07_297
SBOX_V07_296
SBOX_V07_295
SBOX_V07_294
SBOX_V07_293
SBOX_V07_292
SBOX_V07_291
SBOX_V07_290
SBOX_V07_289
SBOX_V07_288
SBOX_V07_287
SBOX_V07_286
SBOX_V07_285
SBOX_V07_284
SBOX_V07_283
SBOX_V07_282
SBOX_V07_281
SBOX_V07_280
SBOX_V07_279
SBOX_V07_278
SBOX_V07_277
SBOX_V07_276
SBOX_V07_275
SBOX_V07_274
SBOX_V07_273
SBOX_V07_272
SBOX_V07_271
SBOX_V07_270
SBOX_V07_269
SBOX_V07_268
SBOX_V07_267
SBOX_V07_266
SBOX_V07_265
SBOX_V07_264
SBOX_V07_263
SBOX_V07_262
SBOX_V07_261
SBOX_V07_260
SBOX_V07_259
SBOX_V07_258
SBOX_V07_257
SBOX_V07_256
SBOX_V07_255
SBOX_V07_254
SBOX_V07_253
SBOX_V07_252
SBOX_V07_251
SBOX_V07_250
SBOX_V07_249
SBOX_V07_248
SBOX_V07_247
SBOX_V07_246
SBOX_V07_245
SBOX_V07_244
SBOX_V07_243
SBOX_V07_242
SBOX_V07_241
SBOX_V07_240
SBOX_V07_239
SBOX_V07_238
SBOX_V07_237
SBOX_V07_236
SBOX_V07_235
SBOX_V07_234
SBOX_V07_233
SBOX_V07_232
SBOX_V07_231
SBOX_V07_230
SBOX_V07_229
SBOX_V07_228
SBOX_V07_227
SBOX_V07_226
SBOX_V07_225
SBOX_V07_224
SBOX_V07_223
SBOX_V07_222
SBOX_V07_221
SBOX_V07_220
SBOX_V07_219
SBOX_V07_218
SBOX_V07_217
SBOX_V07_216
SBOX_V07_215
SBOX_V07_214
SBOX_V07_213
SBOX_V07_212
SBOX_V07_211
SBOX_V07_210
SBOX_V07_209
SBOX_V07_208
SBOX_V07_207
SBOX_V07_206
SBOX_V07_205
SBOX_V07_204
SBOX_V07_203
SBOX_V07_202
SBOX_V07_201
SBOX_V07_200
SBOX_V07_199
SBOX_V07_198
SBOX_V07_197
SBOX_V07_196
SBOX_V07_195
SBOX_V07_194
SBOX_V07_193
SBOX_V07_192
SBOX_V07_191
SBOX_V07_190
SBOX_V07_189
SBOX_V07_188
SBOX_V07_187
SBOX_V07_186
SBOX_V07_185
SBOX_V07_184
SBOX_V07_183
SBOX_V07_182
SBOX_V07_181
SBOX_V07_180
SBOX_V07_179
SBOX_V07_178
SBOX_V07_177
SBOX_V07_176
SBOX_V07_175
SBOX_V07_174
SBOX_V07_173
SBOX_V07_172
SBOX_V07_171
SBOX_V07_170
SBOX_V07_169
SBOX_V07_168
SBOX_V07_167
SBOX_V07_166
SBOX_V07_165
SBOX_V07_164
SBOX_V07_163
SBOX_V07_162
SBOX_V07_161
SBOX_V07_160
SBOX_V07_159
SBOX_V07_158
SBOX_V07_157
SBOX_V07_156
SBOX_V07_155
SBOX_V07_154
SBOX_V07_153
SBOX_V07_152
SBOX_V07_151
SBOX_V07_150
SBOX_V07_149
SBOX_V07_148
SBOX_V07_147
SBOX_V07_146
SBOX_V07_145
SBOX_V07_144
SBOX_V07_143
SBOX_V07_142
SBOX_V07_141
SBOX_V07_140
SBOX_V07_139
SBOX_V07_138
SBOX_V07_137
SBOX_V07_136
SBOX_V07_135
SBOX_V07_134
SBOX_V07_133
SBOX_V07_132
SBOX_V07_131
SBOX_V07_130
SBOX_V07_129
SBOX_V07_128
SBOX_V07_127
SBOX_V07_126
SBOX_V07_125
SBOX_V07_124
SBOX_V07_123
SBOX_V07_122
SBOX_V07_121
SBOX_V07_120
SBOX_V07_119
SBOX_V07_118
SBOX_V07_117
SBOX_V07_116
SBOX_V07_115
SBOX_V07_114
SBOX_V07_113
SBOX_V07_112
SBOX_V07_111
SBOX_V07_110
SBOX_V07_109
SBOX_V07_108
SBOX_V07_107
SBOX_V07_106
SBOX_V07_105
SBOX_V07_104
SBOX_V07_103
SBOX_V07_102
SBOX_V07_101
SBOX_V07_100
SBOX_V07_99
SBOX_V07_98
SBOX_V07_97
SBOX_V07_96
SBOX_V07_95
SBOX_V07_94
SBOX_V07_93
SBOX_V07_92
SBOX_V07_91
SBOX_V07_90
SBOX_V07_89
SBOX_V07_88
SBOX_V07_87
SBOX_V07_86
SBOX_V07_85
SBOX_V07_84
SBOX_V07_83
SBOX_V07_82
SBOX_V07_81
SBOX_V07_80
SBOX_V07_79
SBOX_V07_78
SBOX_V07_77
SBOX_V07_76
SBOX_V07_75
SBOX_V07_74
SBOX_V07_73
SBOX_V07_72
SBOX_V07_71
SBOX_V07_70
SBOX_V07_69
SBOX_V07_68
SBOX_V07_67
SBOX_V07_66
SBOX_V07_65
SBOX_V07_64
SBOX_V07_63
SBOX_V07_62
SBOX_V07_61
SBOX_V07_60
SBOX_V07_59
SBOX_V07_58
SBOX_V07_57
SBOX_V07_56
SBOX_V07_55
SBOX_V07_54
SBOX_V07_53
SBOX_V07_52
SBOX_V07_51
SBOX_V07_50
SBOX_V07_49
SBOX_V07_48
SBOX_V07_47
SBOX_V07_46
SBOX_V07_45
SBOX_V07_44
SBOX_V07_43
SBOX_V07_42
SBOX_V07_41
SBOX_V07_40
SBOX_V07_39
SBOX_V07_38
SBOX_V07_37
SBOX_V07_36
SBOX_V07_35
SBOX_V07_34
SBOX_V07_33
SBOX_V07_32
SBOX_V07_31
SBOX_V07_30
SBOX_V07_29
SBOX_V07_28
SBOX_V07_27
SBOX_V07_26
SBOX_V07_25
SBOX_V07_24
SBOX_V07_23
SBOX_V07_22
SBOX_V07_21
SBOX_V07_20
SBOX_V07_19
SBOX_V07_18
SBOX_V07_17
SBOX_V07_16
SBOX_V07_15
SBOX_V07_14
SBOX_V07_13
SBOX_V07_12
SBOX_V07_11
SBOX_V07_10
SBOX_V07_9
SBOX_V07_8
SBOX_V07_7
SBOX_V07_6
SBOX_V07_5
SBOX_V07_4
SBOX_V07_3
SBOX_V07_2
SBOX_V07_1

CAP3 assembly of different Velvet assemblies

GC% content

Query= SBOX_V07_399 L:7396
(7396 letters)

Database: PlantRefSeq_GB_2009_07_07.fasta
246,809 sequences; 93,580,655 total letters

>Arab_thal.NP_190607 (NP_190607.1 GI:15229738) { unknown protein } [Arabidopsis
thaliana]
Length = 3071

Score = 1771 bits (4588), Expect = 0.0
Identities = 1050/2534 (41%), Positives = 1510/2534 (59%), Gaps = 121/2534 (4%)
Frame = -2

Query: 7395 KCCLAMGQLDFFFLGYSSALSLTLLLRQIQNAFSLEKAQTSTPTYDTPLVRVWDCDSSIAE 7216
KC + +G+LD YSS SL LL+ QI+ A L + + + LV D +A
Sbjct: 652 KCSMVLGKLDIVFEYSSLFSLALLIWQIEWAQKLLVDDYTGEVHSSSLV-TGGVDPEMAS 710

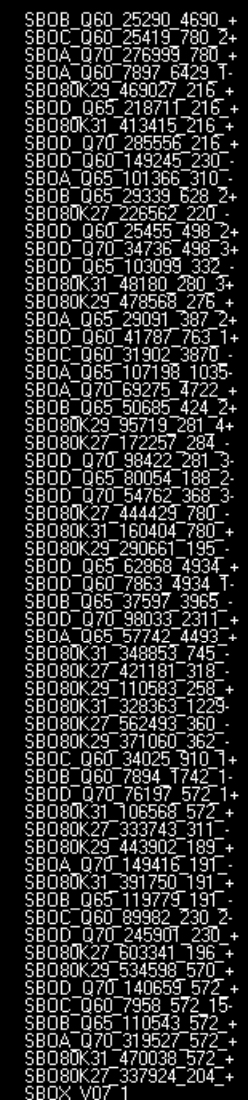
Query: 7215 -----MEKELHKVIEPEKLIIEVGVIYVVGPRIRVSLRK-DSLHKAVD--DIHLSFDC 7075
+E LH+V PE+ I+VG+ + GP+I++ + K + ++ + DI L FD
Sbjct: 711 YDEYGIYRRSIELSLHRVHPERQIQVGILLGGPQIKLLVEKAEVNTLIGKKDI-LLFDF 769

.....

Query: 528 GDSSFHSHPLHHLRGQNEWIRIGPTILTLCHEHLFVNFAIGWLREQAGDL-----TAKI 376
GD S S P L G EW+IGPT+LTLCEHLFV+FAI L++ A A+
Sbjct: 2938 GDPSSSRPRRRLHGDKEWKIGPTVLTLCHEHLFVSFAIRILKQHATKAITSRLPKKEEAEA 2997

Query: 375 NWGDRFKGDPPKEIVKEEESKMSVL-KWGVGRFVFAGMVAYIDGRLCRCIPNPVARRIV 199
D +V + + KM + K G+G FV +G+VAYIDGRLCR IPNP+ARRIV
Sbjct: 2998 ETSDSGSNTAMVPVSDNKKKKMKFMWKAGIGNFVASGIVAYIDGRLCRQIPNPIARRIV 3057

Query: 198 SGFVLSFLDKTRDK 157
SGF+LSFLDK+ ++
Sbjct: 3058 SGFLLSFLDKSSEQ 3071



Query= SBOX_V07_1 L:6481
(6481 letters)

Database: PlantRefSeq_GB_2009_07_07.fasta
246,809 sequences; 93,580,655 total letters

>Viti_vini.XP_002267871 (XP_002267871.1 GI:225430631) { PREDICTED: hypothetical protein } [
Vitis vinifera]
Length = 2543

Score = 3073 bits (7968), Expect = 0.0
Identities = 1577/2092 (75%), Positives = 1801/2092 (86%), Gaps = 2/2092 (0%)
Frame = -3

Query: 6272 EPWVSLASSIPTSSSTKKRIRIFRNEIPSILLNSEMSDSTSQVLVDLIFTTLYIYDDRGR 6093
E ++A S+ T STKKR+RIFR+EIP IL NSEMS++ S LVD+IF TLYIYDD GSR
Sbjct: 6 ESLAAIAGSVSTVSTKKRVRIFRDEIPPILTNSEMSAELASLLVDIIFNTLYIYDDHGSR 65

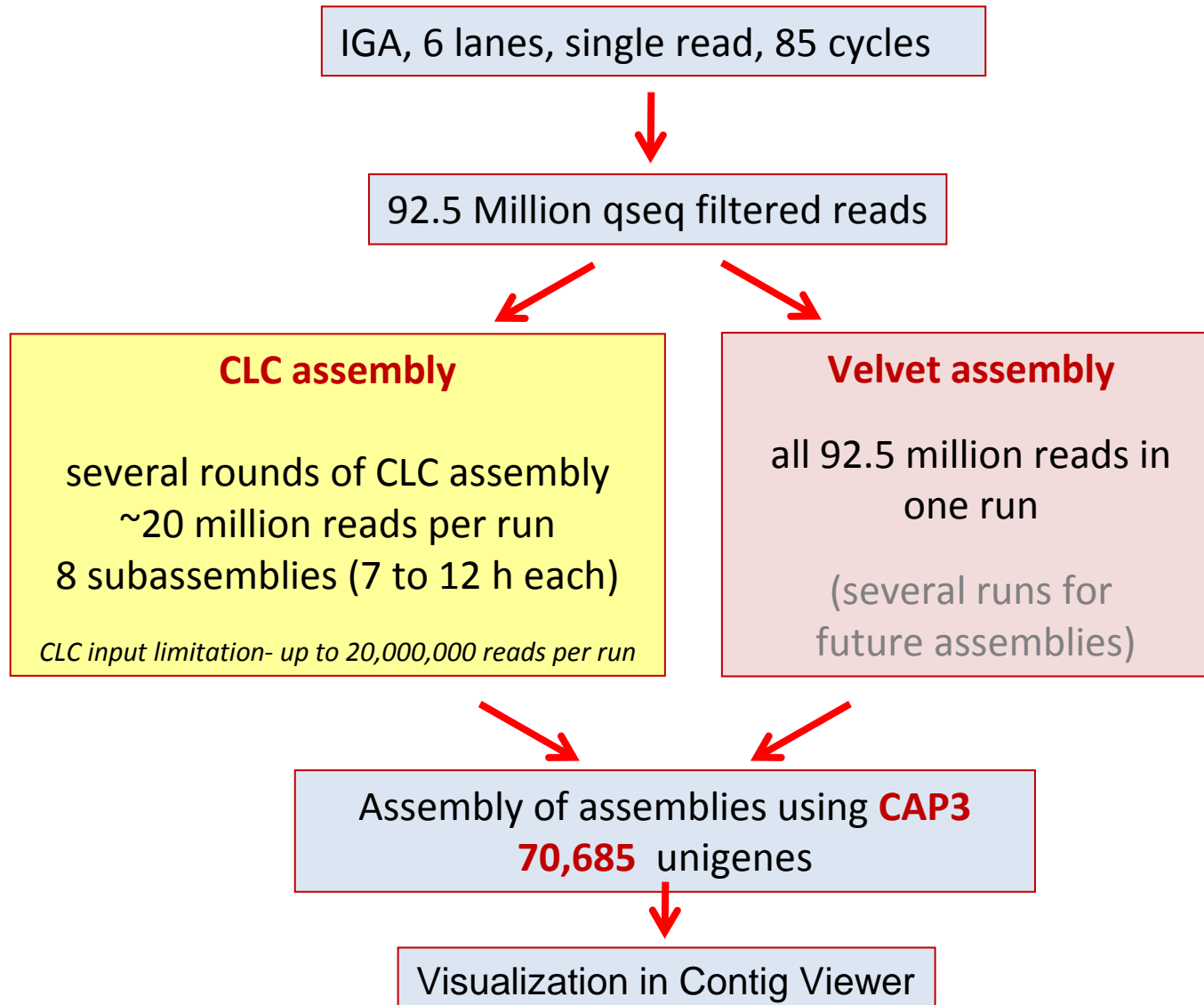
Query: 6092 KAVDDLIIKSLSEVVFMTFAAALVQVMDKQLKVQSHVGCSRLMSWSCILLCKTQFISAS 5913
KAVDD+I K+L EV+FMK+FAA LVQ M+KQ K QS++GC RL+ WSC+LL K++F S S
Sbjct: 66 KAVDDVISKALGEVIFMKSFAATLVQFMEKQSKFQSNIGCYRLKWSCLLLSKSRFASVS 125

.....

Query: 338 LAFSTLYKSAGMQAIDEIVPTLLRALEDDDMADTALDGLKQILSVRTAAVLPHILPKLVH 159
LAFSTLYKSAGMQAIDEIVPTLL +LEDD +DTALDGLKQILSVRT AVLPHILPKLVH
Sbjct: 1901 LAFSTLYKSAGMQAIDEIVPTLLHSLEDDQTSDTALDGLKQILSVRTTAVLPHILPKLVH 1960

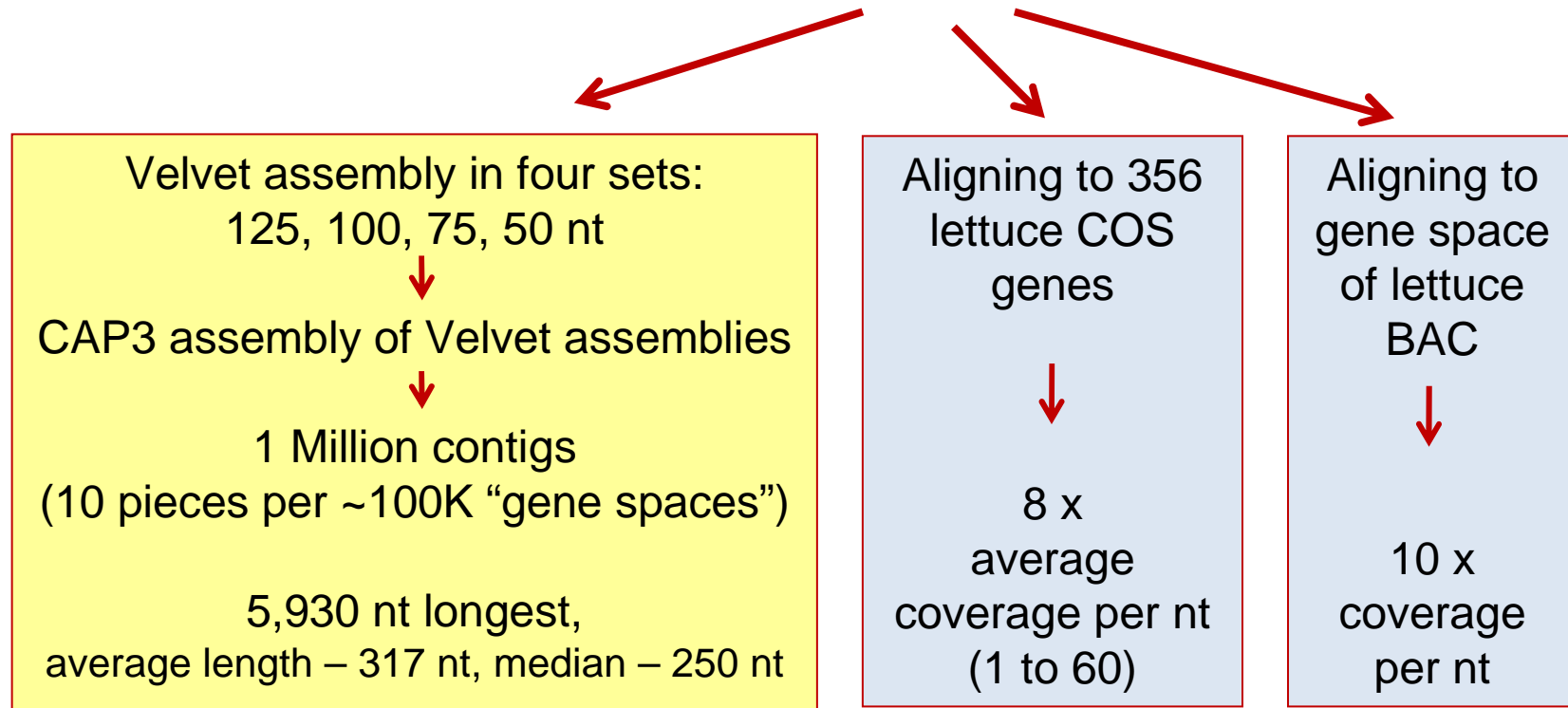
Query: 158 LPLSAFNAHAIGAVAEVAGAGLVHLSTVLPALLSAMGDDDHAEVQNLA KKA 3
LPL+AFNAHA+GA+AEVAG GLN HL VLPALLSAM DDD +VQ LAKKA
Sbjct: 1961 LPLTAFNAHALGALAEVAGPGLNFHLGIVLPALLSAMSDDD-TDVQKLAKKA 2011

Lettuce Transcriptome Sequencing and Assembly Pipeline



Lettuce Gene Space Sequencing

55 million, 125 nt single reads
6 GB (= 2x coverage w/o repeat reduction)



QC Analysis of Transcriptome Assembly





BLASTX versus Plant RefSeq Database: out of 8,300 longest contigs only 83 (1%) have no hits (similarity) to known protein sequences (expectation cutoff 1e-10)

QUERY	SUBJECT	HIT	Query_length_nt	Hit_Alignment_Length_aa	
LSat_DN_X01_992	Arab_thal.NP_186875	(NP_186875.2 GI:30678519) { BIG (BIG); binding / ubiquitin-protein ligase/ zinc ion binding } [Arabidopsis thaliana]	15382	5096	
LSat_DN_X01_25623	Arab_thal.NP_001154245	(NP_001154245.1 GI:238480786) { unknown protein } [Arabidopsis thaliana]	13037	4214	
LSat_DN_X01_20163	Viti_vini.XP_002274761	(XP_002274761.1 GI:225437162) { PREDICTED: hypothetical protein } [Vitis vinifera]	12279	3562	
LSat_DN_X01_15972	Popu_tric.XP_002327756	(XP_002327756.1 GI:224134102) { predicted protein } [Populus trichocarpa]	12155	3881	
LSat_DN_X01_9498	Viti_vini.XP_002267079	(XP_002267079.1 GI:225425575) { PREDICTED: hypothetical protein } [Vitis vinifera]	12112	508	chimeric?
LSat_DN_X01_1	Viti_vini.XP_002268896	(XP_002268896.1 GI:225461945) { PREDICTED: similar to E3 ubiquitin-protein ligase UPL1 } [Vitis vinifera]	11800	3754	
LSat_DN_X01_2	Viti_vini.XP_002283711	(XP_002283711.1 GI:225459044) { PREDICTED: hypothetical protein } [Vitis vinifera]	11743	1948	
LSat_DN_X01_6	Viti_vini.XP_002274451	(XP_002274451.1 GI:225432116) { PREDICTED: similar to vacuolar protein sorting 13C protein-like } [Vitis vinifera]	11434	3718	
LSat_DN_X01_11175	Popu_tric.XP_002327931	(XP_002327931.1 GI:224132994) { predicted protein } [Populus trichocarpa]	11311	3060	
LSat_DN_X01_35703	Viti_vini.XP_002283711	(XP_002283711.1 GI:225459044) { PREDICTED: hypothetical protein } [Vitis vinifera]	10588	3651	
LSat_DN_X01_37	Viti_vini.XP_002283711	(XP_002283711.1 GI:225459044) { PREDICTED: hypothetical protein } [Vitis vinifera]	10568	1948	

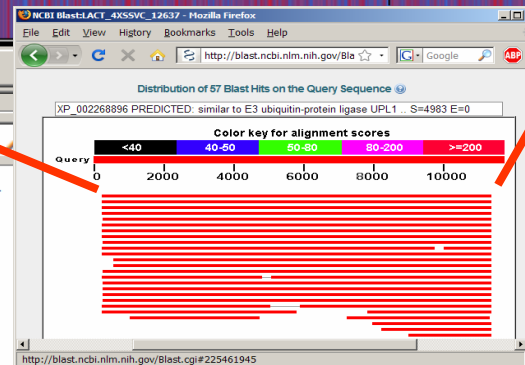
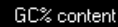
Majority (over 80 - 90%) of longest contigs produce uninterrupted ORFs of expected size

76 Viewer *LACT_4X55VC_12637*

BLAST-X alignment:
11,085 nt
from 256 to 11,340 nt


-  - CLC
-  - Velvet
-  - *L. sativa* ESTs
-  - *L. serriola* ESTs

Lact_eldr_3816 CLC+
Lact_eldo_108304 12+
Lact_eldr_7179 CDC-
Lact_eldr_7690 CLC+
Lact_eldo_13934 CLC-
Lact_eldr_8655 CLC+
Lact_eldo_2152 CLC+
Lact_eldo_47597 19+
Lact_eldo_5746 15+
Lact_eldr_811 CLC-
Lact_eldo_172353 15+
Lact_eldo_36311 T6+
Lact_eldo_8429 T1+
Lact_eldo_201121 17+
Lact_eldr_1570 CDC+
Lact_eldo_629 T6+
Lact_eldr_172896 CLC-
Lact_sati.Dw125308+
Lact_eldo_30011 T6-
Lact_eldo_219908 T6+
Lact_eldr_3633 CLC-
Lact_eldo_39099 16+
Lact_eldo_59739 16+
Lact_eldr_780 CLC-
Lact_eldo_93179 15+
Lact_eldo_53364 17-
LACT_SERR.CSA1.668+
Lact_eldo_132391 20-
Lact_sati.BQ854193+
Lact_eldo_21419 24+
Lact_eldr_766 CDC+
Lact_eldo_86788 26+
LACT_SAT1.CSA1.410+
Lact_eldo_85485 25+
Lact_eldo_9168 22+
Lact_serr.BQ998047+
LACT_4XSSVC.12637



Query start

Query end

 [ref|XP_002268896.1](#) **UG** PREDICTED: similar to E3 ubiquitin-protein ligase UPL1
vinifera]
Length=3754 - protein length - 3754 aa

GENE ID: 100246297 LOC100246297 | similar to E3 ubiquitin-protein ligase UPL1
[Vitis vinifera]

Score = 4983 bits (12926), Expect = 0.0
Identities = 2683/3855 (69%), Positives = 3043/3855 (78%), Gaps = 261/3855 (6%)
From = 1

Query	256	MLKRRRASEVPPIKISFINGVTAAPLESIEEPLKSFFWDKDGDFHHWIDLFNHFDTF	435
Sbjct	1	MLLRRRAEVPPIKISFINGVTAPLEIEEPLKFWFWDKDGDFHHWDLFNHFDTF	60
Query	436	EKYIKPRKLDQEDFLSDPFFPFRESVLQILRVRIILENTCNKHFFSYSEHHLSSLLA	615
Sbjct	61	EKIKPRKLDQEDFLSDPFFPFREAVLQILRVRIILENTCNKHFFSYSEHLSLHLLA	120

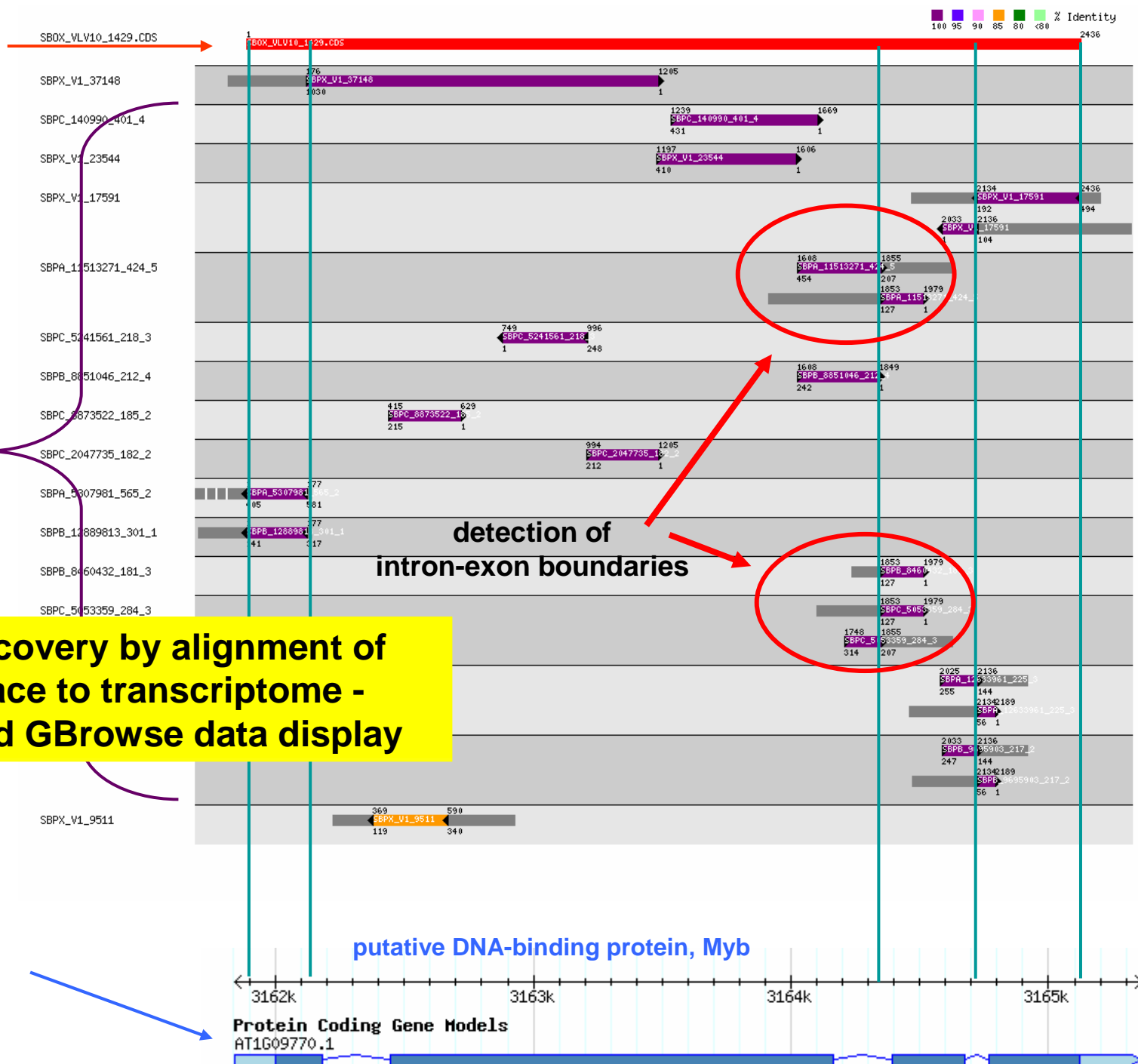
VFDLVAEHILTNAIRPQINSFLEGFNELIPRDLI	10935
VFDLVAEHILTNAIRPQINSFLEGFNEL+PR+LI	3619
FEYTGTVYGSVVVWFVWVKA FNKEDRARLLQF	11115
FEYTGTVYGSVVVWFVWVKA FNKEDRARLLQF	3679
QHKAYGAPERLPSAHTCFNQLDPEYTSKEQLQER	11295
QHKAYGAPERLPSAHTCFNQLDPEYTSKEQLQER	3739

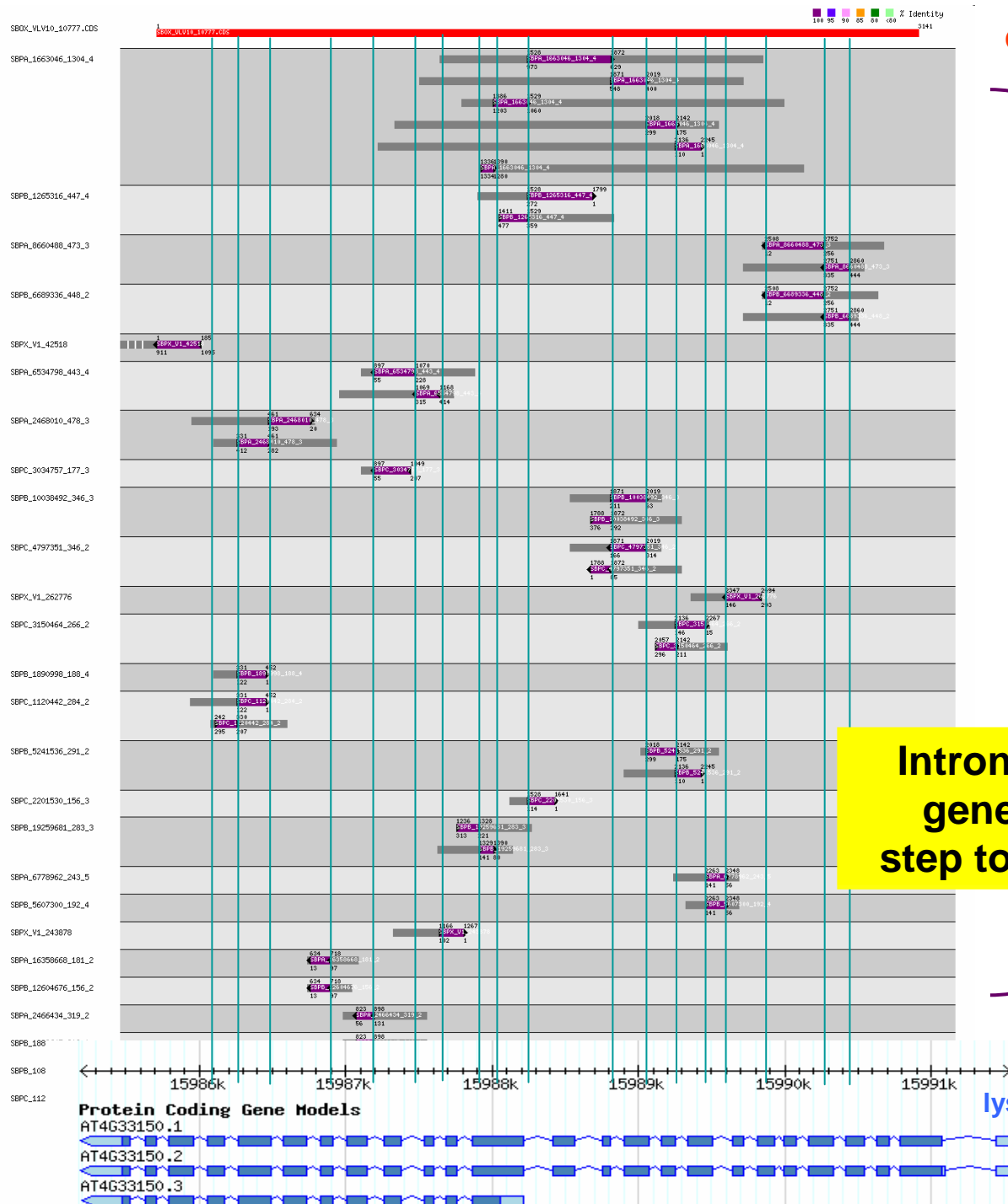
**de novo
transcriptome
assembly**

de novo
gene space
assemblies

Intron discovery by alignment of gene space to transcriptome - step toward GBrowse data display

Arabidopsis gene model





de novo transcriptome assembly

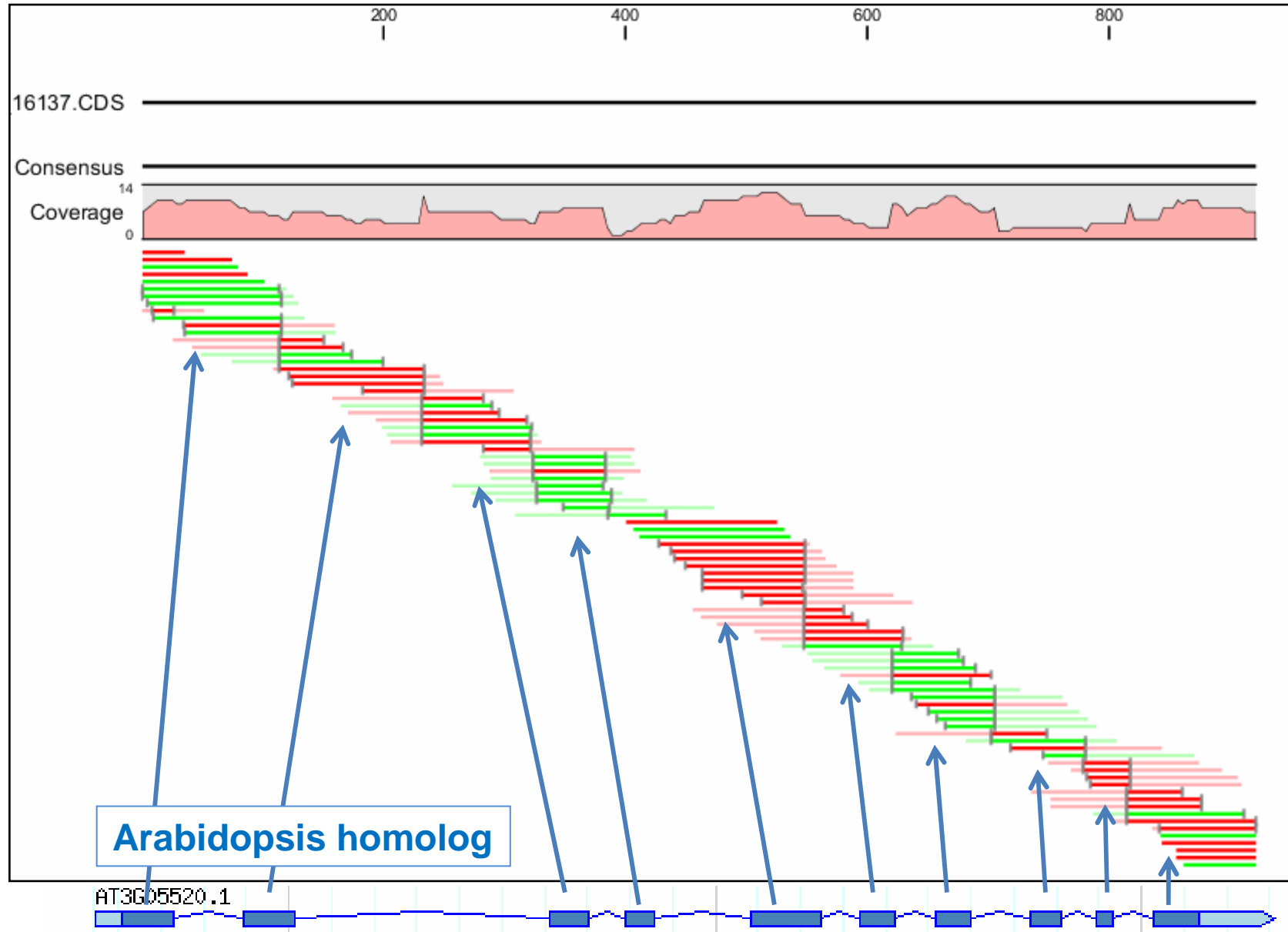
de novo gene space assemblies

Intron discovery by alignment of gene space to transcriptome - step toward GBrowse data display

lysine-ketoglutarate reductase/saccharopine dehydrogenase (LKR/SDH)
Arabidopsis gene model

How Low Can We Go to Figure the Exon-intron Structure?

E.g. gene covered by only 6.6 genomic reads per nt of coding sequence



project overview and dataflow

IGA output files

IGA quality scores

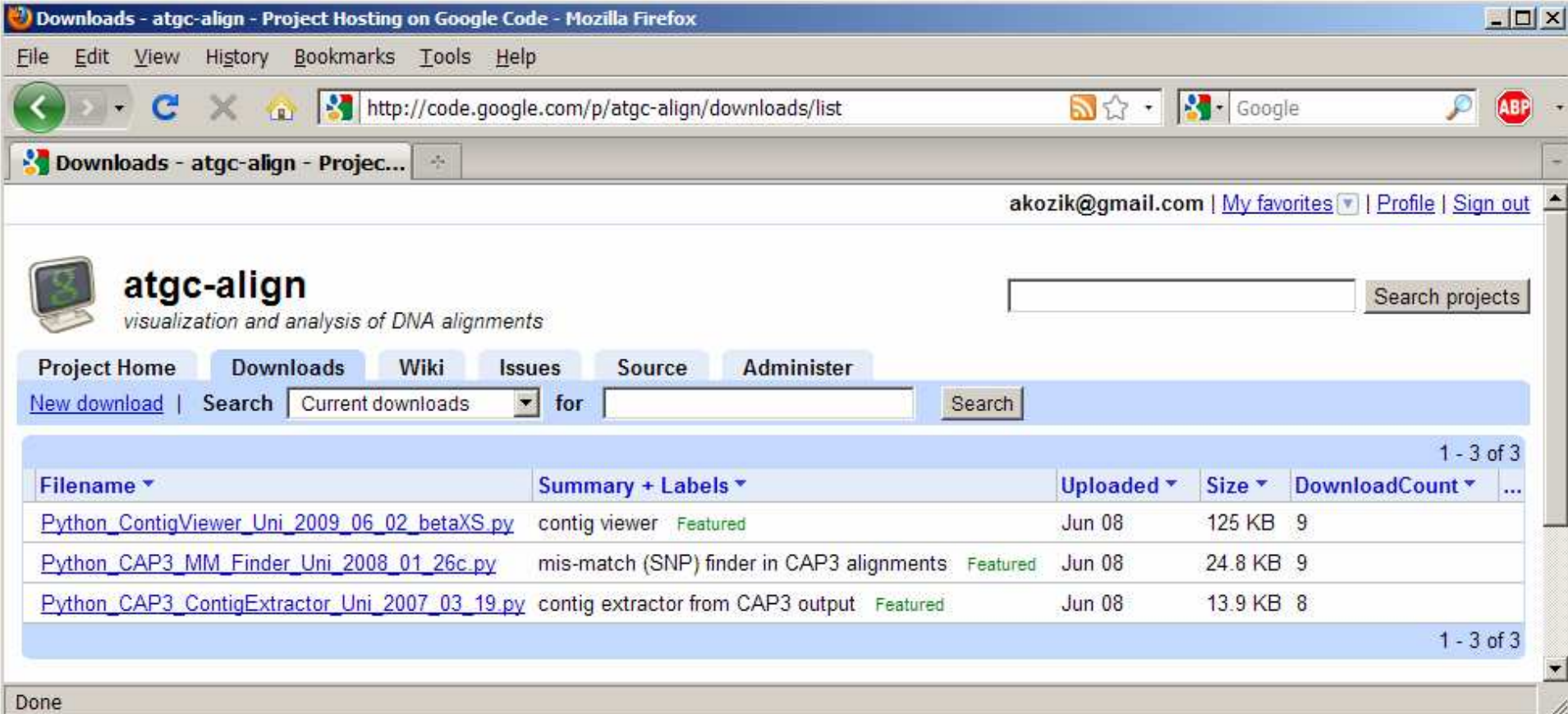
filtering for high quality reads

assembly parameters

visualization with Python Contig Viewer

<http://code.google.com/p/atgc-align/>

post-processing of CAP3 output and assembly visualization with Python Contig Viewer



The screenshot shows the 'Downloads' page of the 'atgc-align' project on Google Code. The browser window is Mozilla Firefox. The page header includes the project name 'atgc-align' and the description 'visualization and analysis of DNA alignments'. The user 'akozik@gmail.com' is logged in. The 'Downloads' tab is selected, showing a list of three files:

Filename	Summary + Labels	Uploaded	Size	DownloadCount
Python_ContigViewer_Uni_2009_06_02_betaXS.py	contig viewer Featured	Jun 08	125 KB	9
Python_CAP3_MM_Finder_Uni_2008_01_26c.py	mis-match (SNP) finder in CAP3 alignments Featured	Jun 08	24.8 KB	9
Python_CAP3_ContigExtractor_Uni_2007_03_19.py	contig extractor from CAP3 output Featured	Jun 08	13.9 KB	8

The status bar at the bottom of the browser window shows 'Done'.

```
LSat_DN_X01_2012.aln - WordPad
File Edit View Insert Format Help

LVK31B_73738_10+
SLNA_CLC2_9733-
SLNB_CLC2_17+
SLN2_CLC1_15388+
SLNA_CLC2_7411+
SLN6_CLC1_12360-
SLNA_CLC2_6050-
SLN2_CLC1_13630-
SLN7_CLC1_14614-
LVK31B_315329_13-
SLNA_CLC2_5251-
LVK31B_136615_14-
SLNA_CLC2_1312+
SLN2_CLC1_21081+
SLNB_CLC2_8044+
SLNB_CLC2_2307+
SLNA_CLC2_4811-
SLN7_CLC1_21128-
SLNA_CLC2_4570-
SLN6_CLC1_560+
SLN8_CLC1_10755+
SLN34_CLC1_7899+
SLN2_CLC1_2363+
SLN7_CLC1_12930+
SLNB_CLC2_1589-
LVK31B_3747_21+
SLNA_CLC2_6098-
SLN2_CLC1_1363+
SLN8_CLC1_4683+
SLN7_CLC1_3141+
SLN6_CLC1_2802+
SLNC_CLC3_1350+

-----
GTGAATTTCTGATCTCATCTTTAAACACTCATTATTGGGCATTGAAAGTGTTCATAATGTACTCAGCTCTTCCCATTCATGCTCAATTTTACCTAATACGTTTACTCAAAATCTGGTCAGGATACACCGTTTGATGTTCTTAGTTCAGACCACT
GTTTCATAATGTACTCAGCTCTTCCCATTCATGCTCAATTTTACCTAATACGTTTACTCAAAATCTGGTCAGGATACACCGTTTGATGTTCTTAGTTCAGACCACT
CTCAATTTTACCTAATACGTTTACTCAAAATCTGGTCAGGATACACCGTTTGATGTTCTTAGTTCAGACCACT

LSat_DN_X01_2012
GTGAATTTCTGATCTCATCTTT
For Help, press F1
```

CAP3 Usage: cap3 File_of_reads [options]

cap3 my_fasta_file.fa -o 80 -p 90 > my_fasta_file.cap3.out &

processing of CAP3 output with
Python_CAP3_ContigExtractor_Uni_2007_03_19.py:

1. - Info table
2. - Alignment files

```
ATGATGTCATCGATCTTTCTTGTCGGTTTGATTTCTGGATCAAAATGATGCTCTCATCGAGGCCAGTATTTTCAGCACTTATTAAGCTTGGAAAGACAGACCTTTAAGAAAGTTAGATATGGTAAACTCCGGGGTCATTGATAATTGCTTGTCTTACTCCCAACTGCCTCAAGTGGTTT

ATGATGTCATCGATCTTTCTTGTCGGTTTGATTTCTGGATCAAAATGATGCTCTCATCG

ATGATGTCATCGATCTTTCTTGTCGGTTTGATTTCTGGATCAAAATGATGCTCTCATCGAGGCCAGTATTTTCAGCACTTATTAAGCTTGGAAAGACAGACCTTTAAGAAAGTTAGATATGGTAAACTCCGGGGTCATTGATAATTGCTTGTCTTACTCCCAACTGCCTCAAGTGGTTT
ATGATGTCATCGATCTTTCTTGTCGGTTTGATTTCTGGATCAAAATGATGCTCTCATCGAGGCCAGTATTTTCAGCACTTATTAAGCTTGGAAAGACAGACCTTTAAGAAAGTTAGATATGGTAAACTCCGGGGTCATTGATAATTGCTTGTCTTACTCCCAACTGCCTCAAGTGGTTT
TATTAAGCTTGGAAAGACAGACCTTTAAGAAAGTTAGATATGGTAAACTCCGGGGTCATTGATAATTGCTTGTCTTACTCCCAACTGCCTCAAGTGGTTT

-----
ATGATGTCATCGATCTTTCTTGTCGGTTTGATTTCTGGATCAAAATGATGCTCTCATCGAGGCCAGTATTTTCAGCACTTATTAAGCTTGGAAAGACAGACCTTTAAGAAAGTTAGATATGGTAAACTCCGGGGTCATTGATAATTGCTTGTCTTACTCCCAACTGCCTCAAGTGGTTT
For Help, press F1
```

Info table - Contig Complexity information

SBOX_V07_10677	99	SBOD_Q65_46506_491_1-(1,521) SBOA_Q70_63067_184_1+(11,224) ...
SBOX_V07_15959	99	SBOD_Q60_38311_405_1+(1,435) SBOB_Q65_71217_151_6+(11,191) ...
SBOX_V07_33750	98	SBO80K31_405877_272_+(1,302) SBO80K29_460109_274_+(1,302) ...
SBOX_V07_2653	96	SBOC_Q60_29343_843_2-(1,873) SBOA_Q60_29209_863_2-(1,893) ...
SBOX_V07_27418	96	SBOB_Q60_23920_307_1-(1,337) SBOC_Q60_24036_283_2-(22,332) ...
...		
SBOX_V07_9935	18	SBO80K29_44764_509_4+(1,537) SBO80K27_49894_511_4+(1,537) ...
SBOX_V07_9961	18	SBOD_Q65_112060_507_+(1,537) SBOD_Q60_84313_508_8+(1,537) ...
SBOX_V07_10000	17	SBOC_Q60_49726_506_8+(1,536) SBOA_Q60_49428_287_1+(11,327) ...
SBOX_V07_10018	17	SBOD_Q70_259498_506_+(1,536) SBOB_Q65_196026_486_+(16,523) ...
...		

contig ID

sequence position and orientation in contig

number of sequences in contig

Python_CAP3_ContigExtractor_Uni_2007_03_19.py usage:

```
bash-2.03$
```

```
bash-2.03$ python Python_CAP3_ContigExtractor_Uni_2007_03_19.py
```

```
What type of output do you want? (1/2): 1
```

```
Enter the SOURCE file name: my_fasta_file.cap3.out
```

← option '1' to extract Info table

```
Enter the DESTINATION file name: my_fasta_file.cap3.out.Info
```

```
Default contig file name prefix is Contig
```

```
Enter the contig file name prefix: ASSY_V1_
```

```
read data from the input file my_fasta_file.cap3.out ...
```

```
bash-2.03$
```

```
bash-2.03$ python Python_CAP3_ContigExtractor_Uni_2007_03_19.py
```

```
What type of output do you want? (1/2): 2
```

```
Enter the SOURCE file name: my_fasta_file.cap3.out
```

← option '2' to extract alignments

```
Enter the DESTINATION directory with alignments: my_fasta_file.cap3.out.Align
```

```
Default contig file extension is aln
```

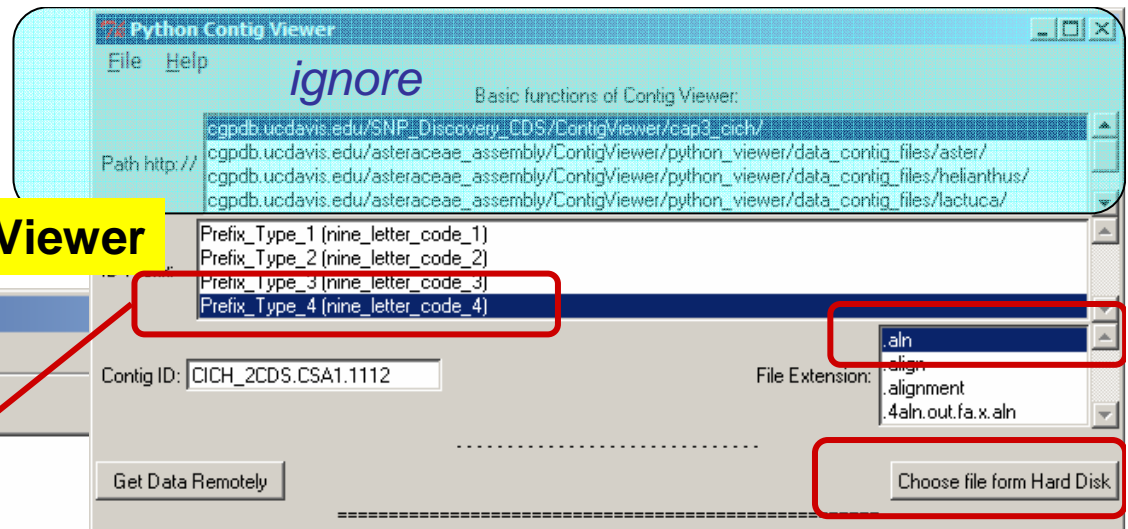
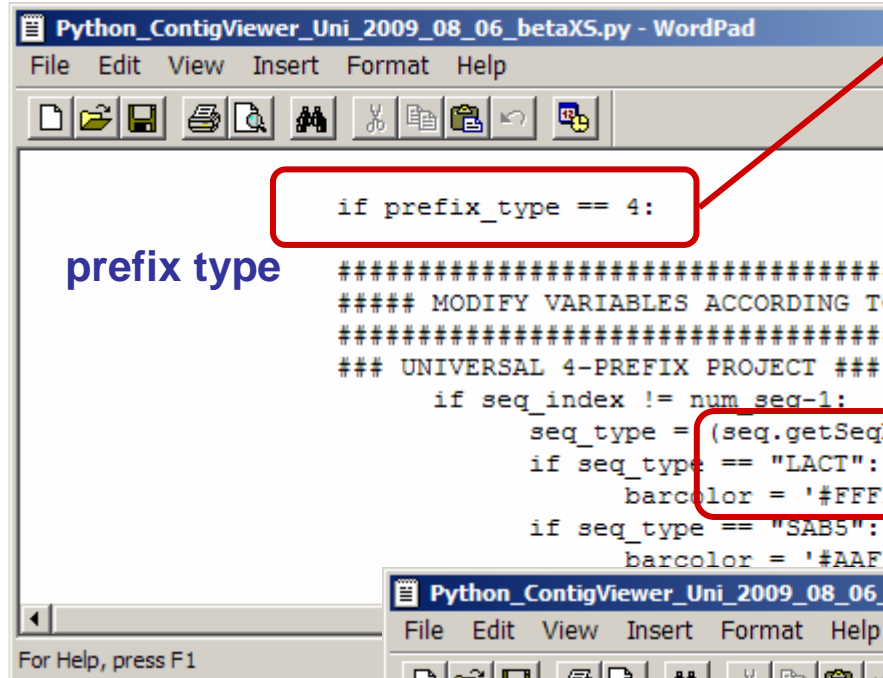
```
Enter the contig file extension :
```

```
Default contig file name prefix is Contig
```

```
Enter the contig file name prefix: ASSY_V1_
```

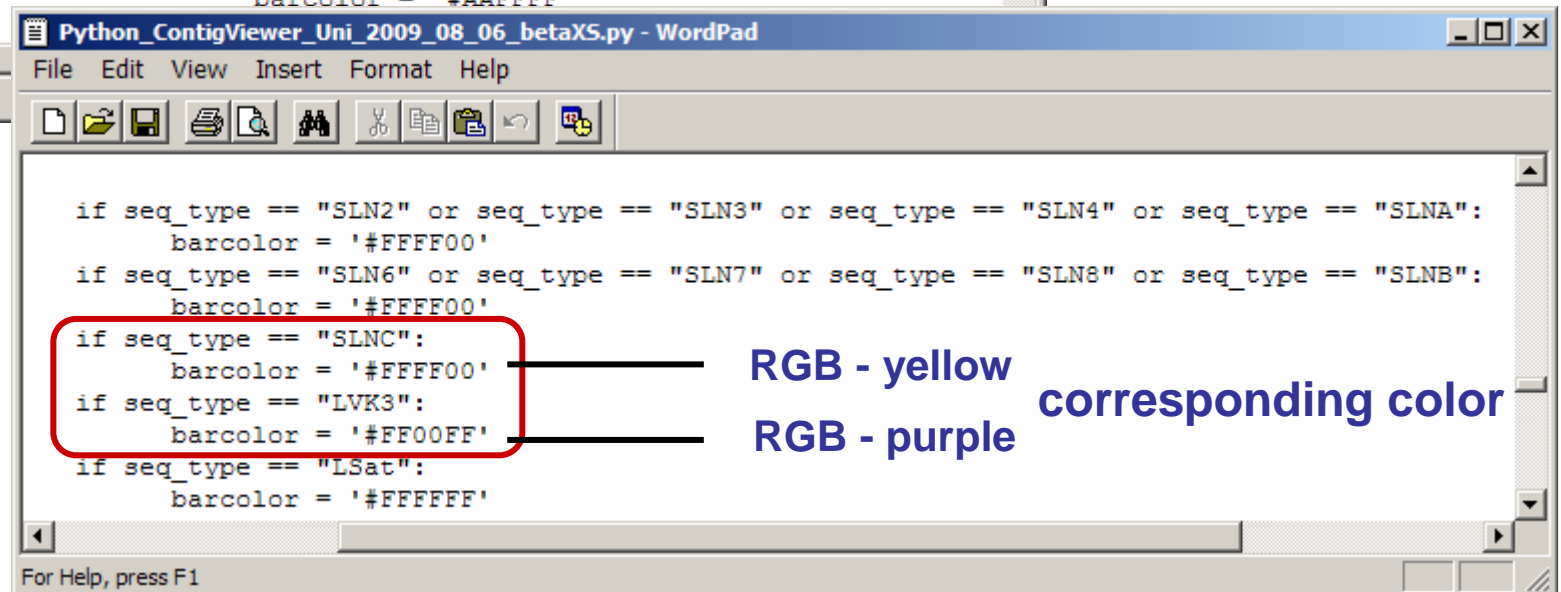
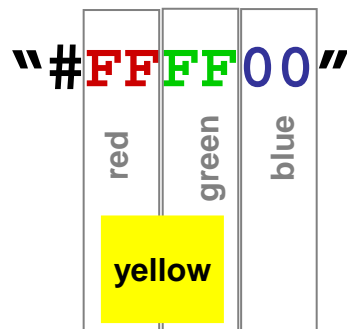
```
read data from the input file my_fasta_file.cap3.out ...
```

Color selection in Python Contig Viewer



color defined by prefix
of sequence ID

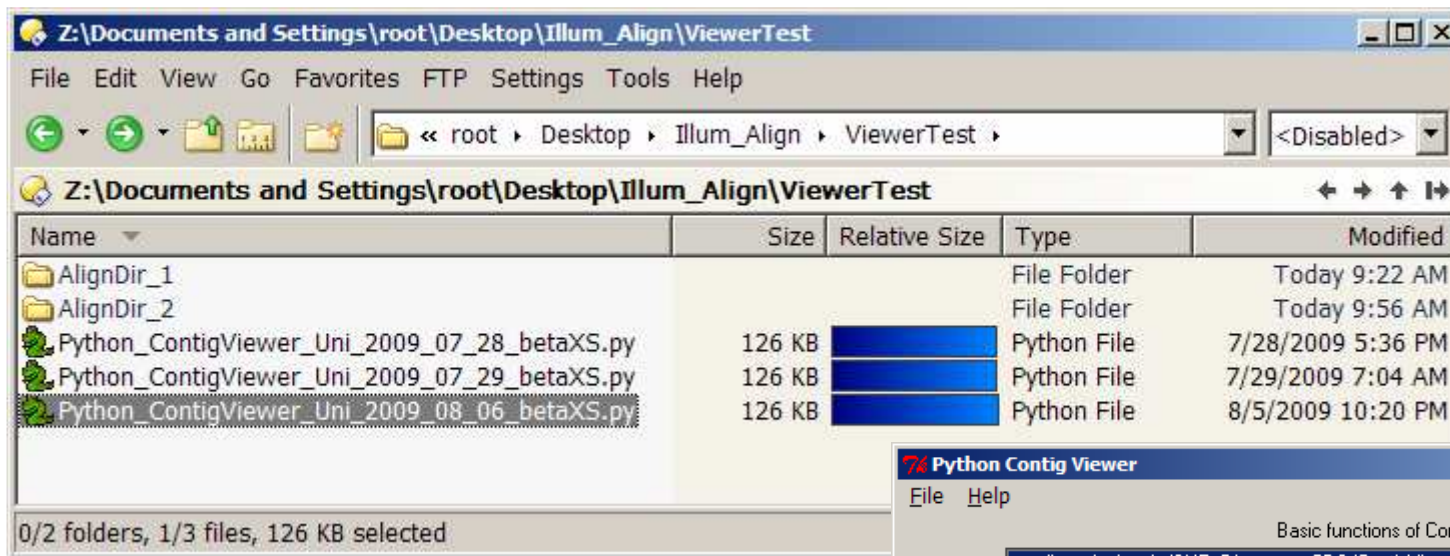
prefix size



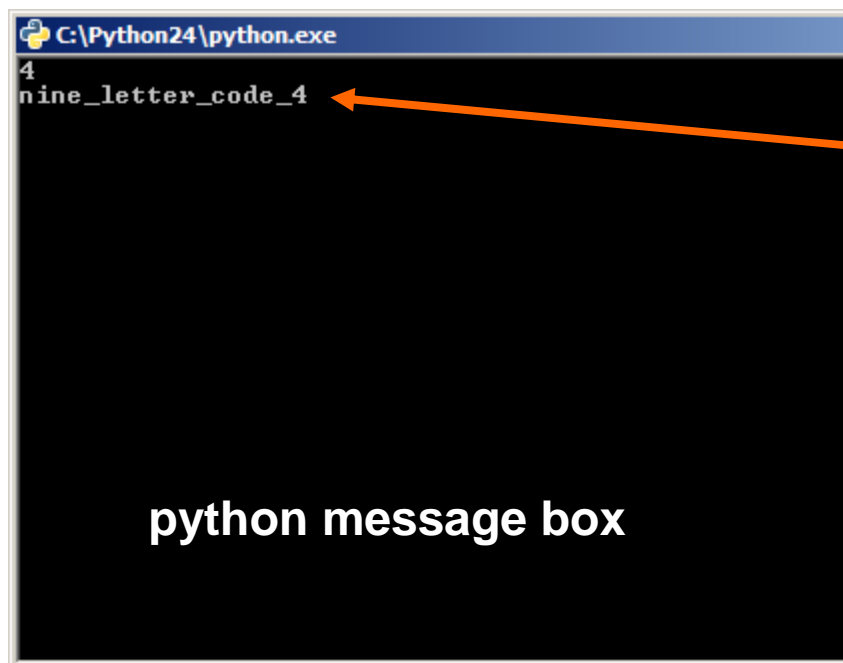
RGB - yellow

RGB - purple

corresponding color

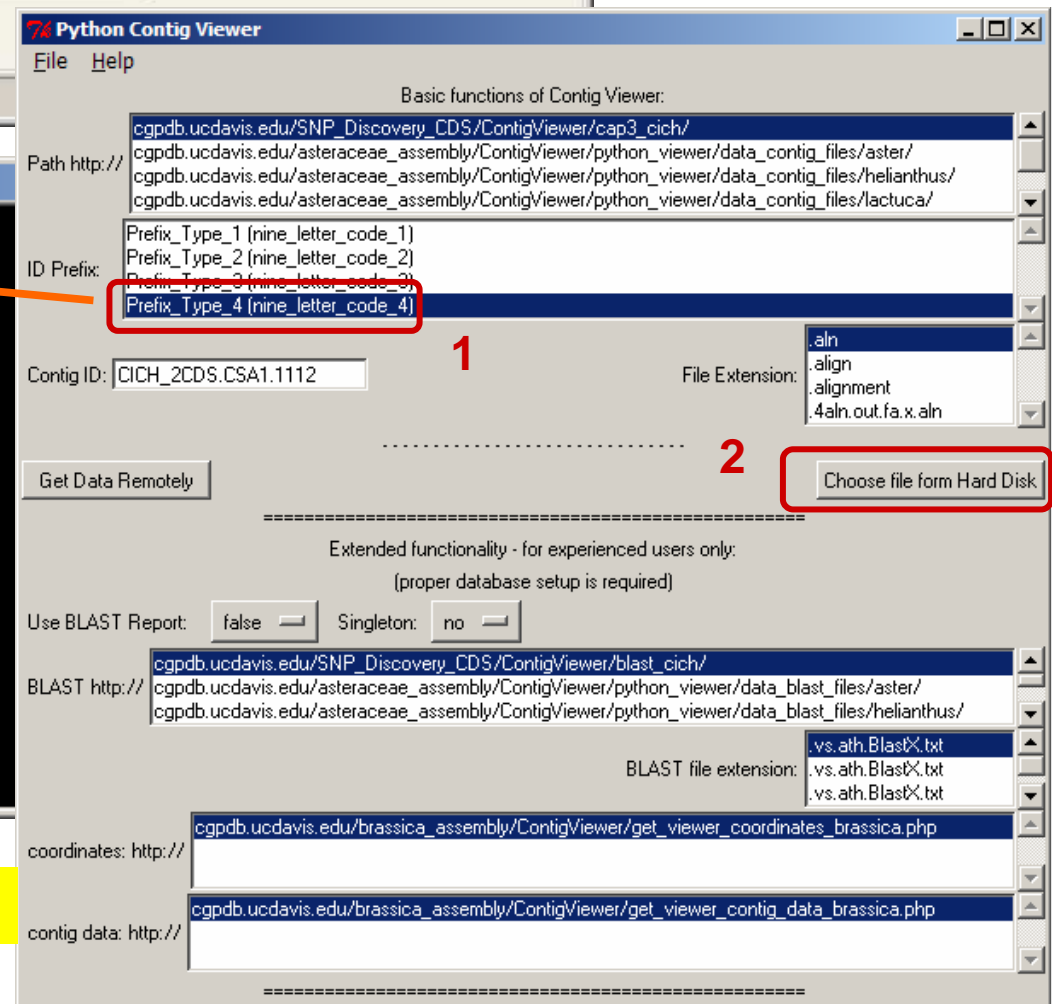


cross-platform
application



python message box

Python Contig Viewer input selection

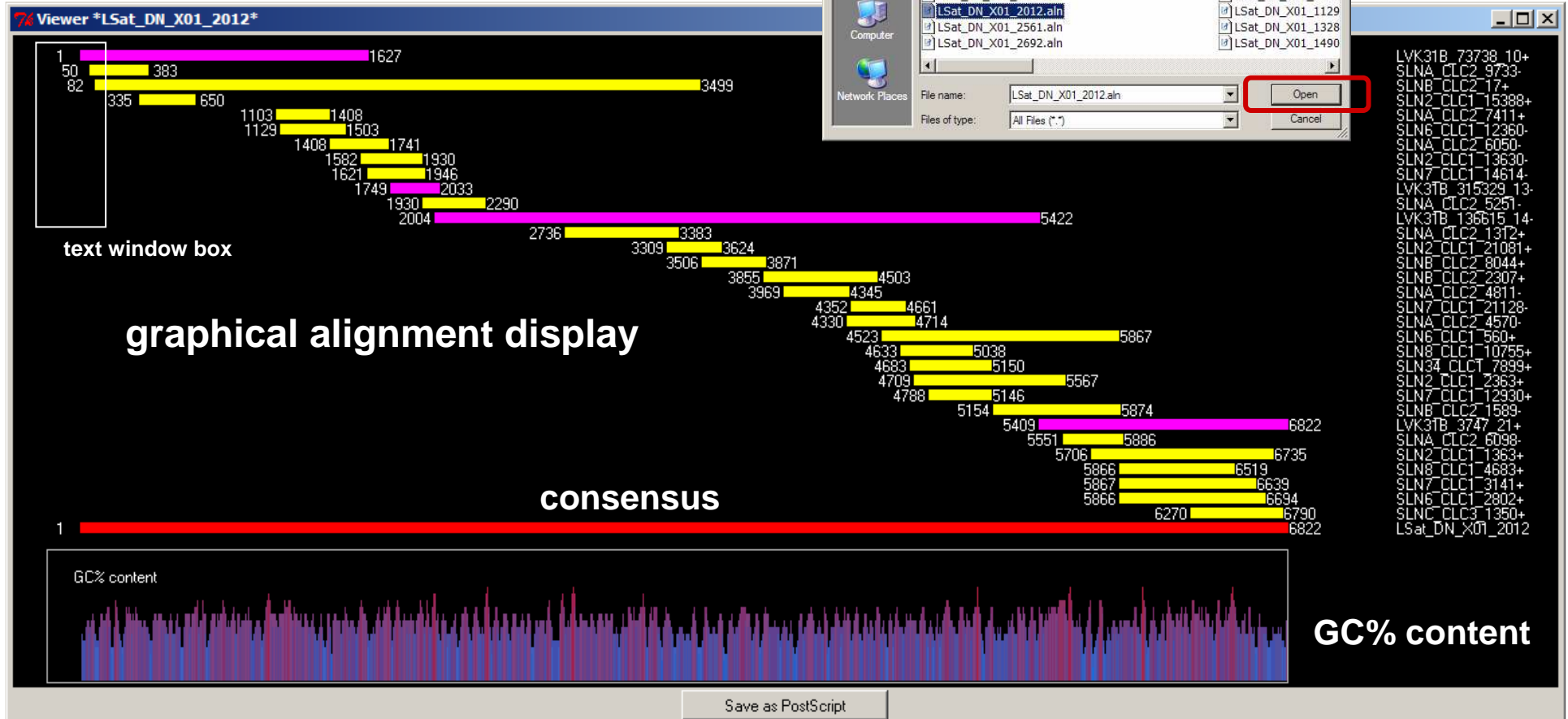
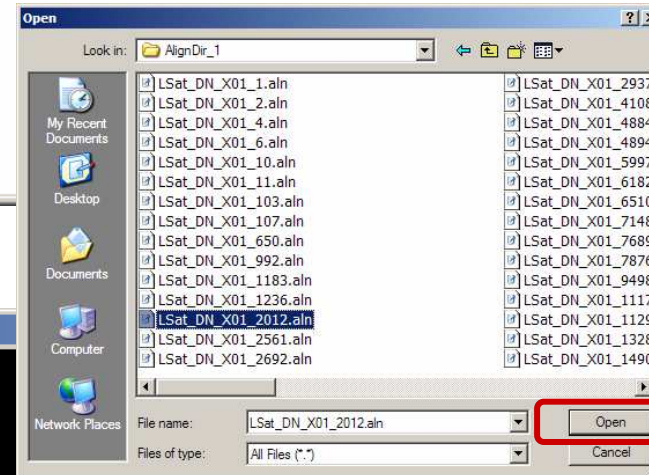


7% Sequence Text *LSat_DN_X01_2012*

```
LVK31B_73738_10+      GTGAATTTCTGATCTCATCTTTAAACACTCATTATTGGGCATTGAAGTCGTTTCATAATGTACTCAGCTCTTCCCATTCAATGCTCAATTTTACCTAAT.  
SLNA_CLC2_9733-      GTTCATAATGTACTCAGCTCTTCCCATTCAATGCTCAATTTTACCTAAT.  
SLNB_CLC2_17+      CTCAATTTTACCTAAT.  
SLN2_CLC1_15388+  
SLNA_CLC2_7411+  
SLN6_CLC1_12360-  
SLNA_CLC2_6050-  
SLN2_CLC1_13630-  
SLN7_CLC1_14614-  
LVK31B_315329_13-  
SLNA_CLC2_5251-  
LVK31B_136615_14-
```

prefix defines the color

text box



last remarks

UNIX tricks

fasta header modifications:

*regular expressions and
perl /find/replace/ function*

>NODE_24_length_783_cov_47.872288

perl -p -i -e 's/NODE_/SBPD_/' my_fasta_file.txt

perl -p -i -e 's/_length_/_/' my_fasta_file.txt

perl -p -i -e 's/_cov_/_/' my_fasta_file.txt

perl -p -i -e 's/\\.*/_/' my_fasta_file.txt

>SBPD_24_783_47

trimming:

>SB08_8_120_1640_1715

CCAAGAACGACATATAAACATTTTACATAATCTGGGGAACACTTGAAGAACATCAAAAGCTGGAAAAATGGATGAAAAACAGCTC

perl -p -i -e 's/\\>SBO/XXXXXXXXXXXXXXXX\\>SBO/' my_fasta_file.orig.txt

XXXXXXXXXXXXXXXXXX>SB08_8_120_1640_1715

CCAAGAACGACATATAAACATTTTACATAATCTGGGGAACACTTGAAGAACATCAAAAGCTGGAAAAATGGATGAAAAACAGCTC

cut -c16-75 my_fasta_file.orig.txt > my_fasta_file.trim.txt

>SB08_8_120_1640_1715

AAACATTTTACATAATCTGGGGAACACTTGAAGAACATCAAAAGCTGGAAAAATGGATGA

Unix 'cut' command

Troubleshooting

```
pgfmars.ucdavis.edu - PuTTY

Lact_sati.DY960922+ TTCCCAGG
Lact_eldr_2423_CLC- TTCCCAGG
Lact_eldr_5493_CLC- TTCCCAGG

consensus TTCCCAGG

Lact_sati.DY960922+ CTCCTGTT
Lact_eldr_2423_CLC- CTCCTGTT

Lact_sati.DY960922+ CATTCAGC
Lact_eldr_2423_CLC- CATTCAGC
Lact_eldr_5493_CLC- CATTCAGC

consensus CATTCAGC

Lact_sati.DY960922+ ACAAACAC
Lact_eldr_2423_CLC- ACAAACAC
Lact_eldr_5493_CLC- ACAAACAC

consensus ACAAACAC

Lact_sati.DY960922+ TTTCCGATAACAAACAAACCGA
Lact_eldr_2423_CLC- TTTCCGATAACAAACAAACCGA
Lact_eldr_5493_CLC- TTTCCGATAACAAACAAACCGA

consensus TTTCCGATAACAAACAAACCGA
```

```
pgfmars.ucdavis.edu - PuTTY

Lact_eldr_5531_CLC- GTGAGGTTTATCGTTCA^MGAATGGAATGGAAGTGAAGGTTTATGAAC
Lact_eldr_24344_CLC- AAGTTTATGAAC

consensus GTGAGGTTTATCGTTCANGAATGGAATGGAAGTGAAGGTTTATGAAC

Lact_eldr_5531_CLC- CAGGATATATC-AGGCGAT^MGCACATAACTCAGTTTAAAAGCGAAGTTGAGATCATGTTGA
Lact_eldr_24344_CLC- CAGGATATATC^MAGGCGAT-GCACATAACTCAGTTTAAAAGCGAAGTTGAGATCATGTTGA

consensus CAGGATATATC^MAGGCGATNGCACATAACTCAGTTTAAAAGCGAAGTTGAGATCATGTTGA

Lact_eldr_5531_CLC- GGTGAGACATCC-TAATATT^MGTTCCTTTTCATGGGAGCAGTTACTCGTCCGCCAAATCT
Lact_eldr_24344_CLC- GGTGAGACATCC^MTAATATT-GTTCCTTTTCATGGGAGCAGTTACTCGTCCGCCAAATCT

consensus GGTGAGACATCCNTAATATTNGTTCCTTTTCATGGGAGCAGTTACTCGTCCGCCAAATCT

Lact_eldr_5531_CLC- A^MTATTCTACCAAGGGGTAGTTTATTTAAATTATTGCATCGTTTCA
Lact_eldr_24344_CLC- TTCCATACTTACAGANATTTCTACCAAGGGGTAGTTTATTTAAATTATTGCATCGTTTCA

consensus TTCCATACTTACAGANATTTCTACCAAGGGGTAGTTTATTTAAATTATTGCATCGTTTCA

Lact_eldr_24344_CLC- GTATCCAAGTTGATGA^MAAAGAGACGAATGCGTATGGCTCTTGATGTGGCCAAGGGGATG

consensus GTATCCAAGTTGATGANAAAGAGACGAATGCGTATGGCTCTTGATGTGGCCAAGGGGATG

Lact_eldr_24344_CLC- AACTATTTGCACACTAG

consensus AACTATTTGCACACTAG

Lact_eldr_24344_CLC- TAGTTTCAATTCCGTTGCG-TCTTCCGCTTCCATTCA
Lact_eldr_24344_CLC- TAGTTTCAATTCCGTTGCG-TCTTCCGCTTCCATTCA

consensus TTTCCGATAACAAACAAACCGANTAGTTTCAATTCCGTTGCGNTCTTCCGCTTCCATTCA
```

corrupted output because of 'new line' problem
<http://en.wikipedia.org/wiki/Newline>

visual inspection of output

record and keep all Log files, run parameters

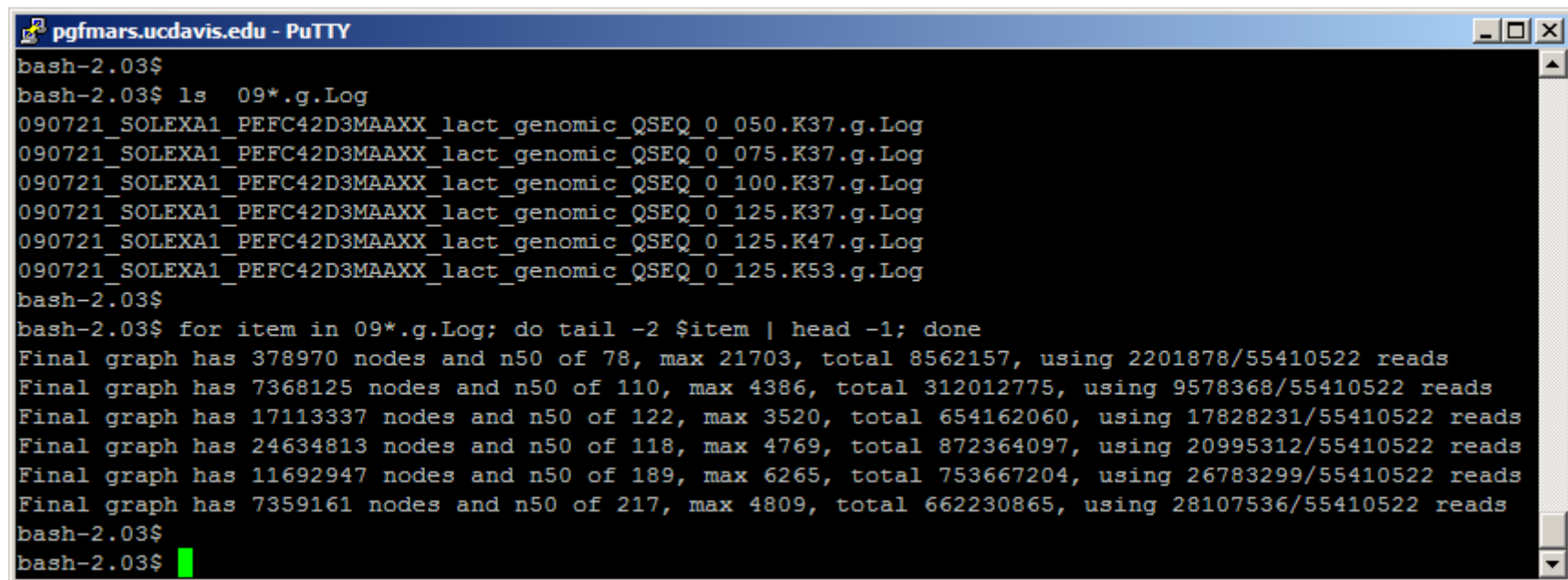
```
bash-2.03$ less 090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_125.K53/Log
```

```
Thu Sep  3 03:07:09 2009
```

```
velvetg.7.49.K53.exec 090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_125.K53 53 -fasta -short  
090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_125.K37/Sequences
```

```
Thu Sep  3 09:02:46 2009
```

```
velvetg.7.49.K53.exec 090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_125.K53 -min_contig_lgth 180  
-unused_reads yes -amos_file yes -read_trkg yes
```



```
pgfmars.ucdavis.edu - PuTTY  
bash-2.03$  
bash-2.03$ ls 09*.g.Log  
090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_050.K37.g.Log  
090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_075.K37.g.Log  
090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_100.K37.g.Log  
090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_125.K37.g.Log  
090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_125.K47.g.Log  
090721_SOLEXA1_PFC42D3MAAXX_lact_genomic_QSEQ_0_125.K53.g.Log  
bash-2.03$  
bash-2.03$ for item in 09*.g.Log; do tail -2 $item | head -1; done  
Final graph has 378970 nodes and n50 of 78, max 21703, total 8562157, using 2201878/55410522 reads  
Final graph has 7368125 nodes and n50 of 110, max 4386, total 312012775, using 9578368/55410522 reads  
Final graph has 17113337 nodes and n50 of 122, max 3520, total 654162060, using 17828231/55410522 reads  
Final graph has 24634813 nodes and n50 of 118, max 4769, total 872364097, using 20995312/55410522 reads  
Final graph has 11692947 nodes and n50 of 189, max 6265, total 753667204, using 26783299/55410522 reads  
Final graph has 7359161 nodes and n50 of 217, max 4809, total 662230865, using 28107536/55410522 reads  
bash-2.03$  
bash-2.03$
```

```
bash-2.03$ for item in 09*.g.Log; do tail -2 $item | head -1; done  
Final graph has 378970 nodes and n50 of 78, max 21703, total 8562157, using 2201878/55410522 reads  
Final graph has 7368125 nodes and n50 of 110, max 4386, total 312012775, using 9578368/55410522 reads  
Final graph has 17113337 nodes and n50 of 122, max 3520, total 654162060, using 17828231/55410522 reads  
Final graph has 24634813 nodes and n50 of 118, max 4769, total 872364097, using 20995312/55410522 reads  
Final graph has 11692947 nodes and n50 of 189, max 6265, total 753667204, using 26783299/55410522 reads  
Final graph has 7359161 nodes and n50 of 217, max 4809, total 662230865, using 28107536/55410522 reads  
bash-2.03$
```

SUMMARY

ILLUMINA output data format

criteria for filtering

trimming and sampling effect on assemblies

assembly parameters

**assembly validation and
exon-intron prediction by alignment**

visualization with Python Contig Viewer

Acknowledgements

Marta Matvienko

UCD Genome Center Bioinformatics Core for IT support

UCD DNA Technologies Core

special credits to Huaqin Xu - database and web programmer

Brian Chan - initial work with Python Contig Viewer

Richard Michelmore

SEQanswers forum and Velvet newsgroup