Double Assembly Method with Characteristics of k-mer's Coverage for Contig

Ayako OHSHIRO[†], **Takeo OKAZAKI**[††] **AND Morikazu NAKAMURA**[††]

[†]Graduate School of Engineering and Science, University of the Ryukyus,
[††]Faculty of Engineering, University of the Ryukyus, Nakagami, Okinawa 903-0213, Japan

Summary
Reads sequences from giga sequencer with parallel processing include many read errors. Various de Novo assembly methods by use of k-mer has been proposed in order to remove read error region and derive contigs, such as Velvet, ABySS, SSAKE and so on. Hybrid assembly algorithms by integrating the results of traditional assembly method have been proposed such as MAIA and GAA. Because assembly result depends on assembly algorithm and k value, it is difficult to obtain assembly results robustly. In this paper, we designed a double assembly method merging different k-mers and applied it to combining rules with a characteristic distribution of k-mer's coverage value for contig. We compared our proposed method with traditional assembly method to evaluate the effectiveness. The experiment was carried out by use of E. coli data and evaluated its performance with coverage ratio, the correct ratio of output contigs.

Key words:
DNA double assembly, k-mer, coverage, characteristic distribution

1. Introduction

In present Bioinformatics, the research of genetic information analysis such as sequence alignment, motif detection, DNA assays has been growing and these depend on accurate DNA sequence. We can obtain DNA sequence from the sequencer. Because readable sequence length of sequencer has a limit, sequences are duplicated by PCR processing and fragmented until sequencer can read. The output reads by sequencer are bonded by use of information of overlap region among reads. This process is called DNA assembly. Thanks to the rapid growth of Giga sequencer technologies [1] with parallel processing, we can derive massive read sequences from sequencer today. However, reads containing read-error region makes DNA assembly difficult, and causes misassemble. In order to solve this problem, many studies about DNA assembly by use of subsequences from read sequence called *k*-mer [2] have been carried out such as EDAR[3]. Reads are broken into *k*-mers by shifting the index of positions on each reads. They have coverage value that means frequency value in all read sequences and *k*-mers with the especially low coverage value are removed as read-error of sequencer. Several DNA assembly algorithms by use of *k*-mer has been proposed. Rene, et al [4] applied *k*-mer to construct suffix tree and removed read containing low base coverage in the path searching process. Daniel, et al [5] removed *k*-mer with especially low frequency value and constructed de Bruijn graph. It determined the optimal path for simplified graph by use of Pebble and Rock Band [6] and generated contigs. Jared et al [7] applied Velvet to parallel processing. Yu Peng et al [8] proposed iterative de Bruijn graph by altering *k* value sequential and generated contigs as IDBA. Jurgen [9] integrated multiple assembly methods by evaluating pairwise alignment among all pairs of contigs and constructed overlap graph with weighted edges, and finding highest scoring path in MAIA algorithm. Guohui [10] evaluated performance of traditional assembly methods by use of CE statistics [11] and combined contigs in accordance with pairwise alignment sore in GAA algorithm. Jared et al [12] removing reads containing read-error by use of *k*-mer and constructing substring graph such as SGA algorithm.

From the above, almost traditional assembly methods utilize the feature of overlap region among *k*-mers, reads, and contigs. But it is difficult to derive accurate contigs in the process of de Novo assembly. Fig. 1 shows the relation of overlap length and accuracy of merged sequence. Vertical axis means overlap length among contigs and horizontal axis means consistent or inconsistent in reference about bonded contig.

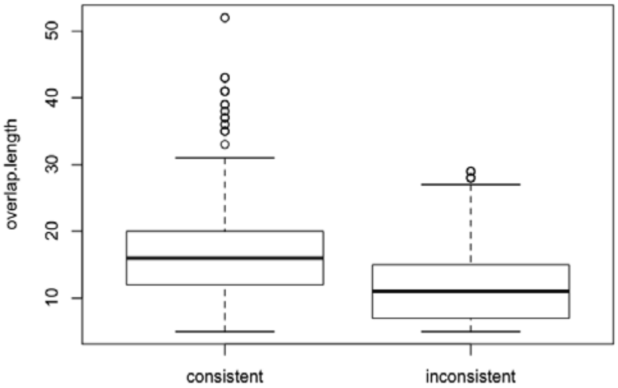


Fig. 1 Relation between overlap length and bonded contig’s accuracy.

Manuscript received February 5, 2014
Manuscript revised February 20, 2014

We can see that when the range of overlap length is more than 35 bases, merged sequence can be expected to be consistent sequence. But there are a mixture of consistent and inconsistent sequence about the range of 5-30base. Overlap length can be said to be insufficient to evaluate accuracy of contig combining. Furthermore, we found that assembly result depends on *k* value and assembly method and proposed DAwH(Double Assembly method with Heuristic) [13] to derive the assembly result robustly by integrating the result of different *k*-mer and methods. But, DAwH outputs many negative contigs. Generally, derived contigs from traditional assembly methods with *k*-mer is composed by *k*-mers and each of them has coverage value shown as Fig. 2. We used the set of coverage value to generate discriminant rule combined contigs and proposed DAwML (Double Assembly method with Machine Learning) to apply contig combining rules on DAwH.

We proposed DAwCC (Double Assembly method with Characteristics of Contig for *k*-mer's coverage) applying contig combining rule to DAwH.

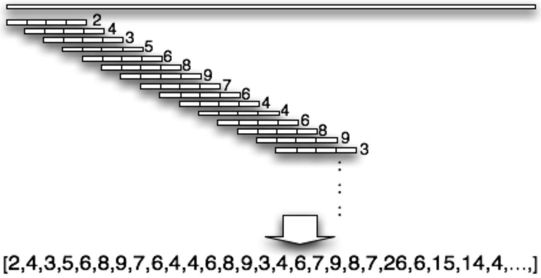


Fig. 2 Contig is constructed by *k*-mer

2. Relation between distribution of *k*-mer's coverage value for contig and contig's accuracy

As mentioned in the introduction, contigs by traditional assembly method with *k*-mer may be considered as the sets of coverage value for *k*-mers. We had a pre - experiment in order to affirm about the relation between the distribution of *k*-mer's coverage value for contig and contig's accuracy. Firstly, we derived contigs from ABySS and Velvet with multiple *k* value and combined them with conditions by overlap region more than 5 bases. Secondly, we classified them to consistent and inconsistent group by comparison with reference. Thirdly, because *k*-mer's coverage value ($c_{i,(i=1,\ldots,n)} \in C$) depends on *k* value, we applied *p*-value for them in accordance with (1). Finally, we extracted *p*-value's distribution feature of all contig pairs for each consistent group and inconsistent group.

$$\mathrm{p}_{c_i} = \frac{|\{c_i \in C \mid c^C \geq c_i\}|}{|C|} \qquad (1)$$

We used E.coli genome registered in NCBI [14]. We will report on some of the results by fixing former or latter contig in this paper. Fig. 3~Fig. 6 show the classified result about combined contig and *p*-value's distribution feature. Left figures represent former contig and light figures represent latter contig. Horizontal axis means the position of *k*-mer constructing contig and vertical axis means *p*-value of *k*-mer for each position. We report two results in the case of fixed former or latter contig, in order to clarify the difference of distribution of *p*-value. When we show *p*-value's fluctuation character as waveform, we can say that there are similarities about latter contigs on Fig. 3 and Fig. 5 and former contig on Fig. 4 and Fig. 6.

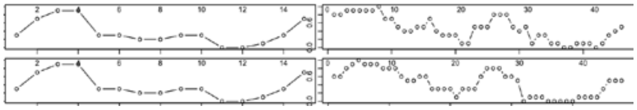


Fig. 3 Pair of *p*-value's fluctuation feature for *k*-mer's coverage value generating correct sequence(former contig is fixed)

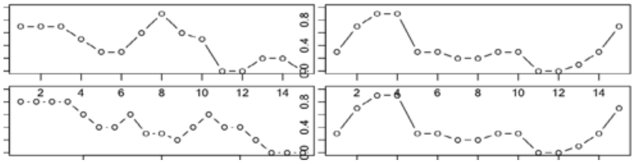


Fig. 4 Pair of *p*-value's fluctuation feature for *k*-mer's coverage value generating correct sequence(latter contig is fixed)

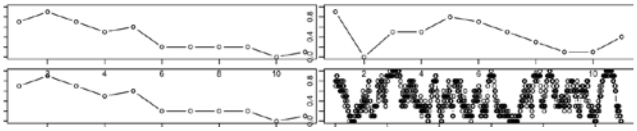


Fig. 5 Pair of p-value's fluctuation feature for *k*-mer's coverage value generating inconsistent sequence (former contig is fixed)

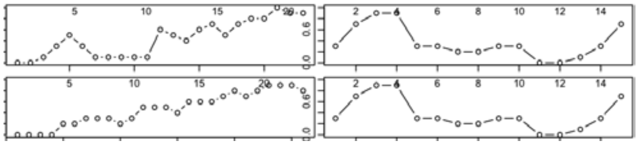


Fig. 6 Pair of p-value's fluctuation feature for *k*-mer's coverage value generating inconsistent sequence (latter contig is fixed)

Therefore, we found that there is a relation between distribution of *k*-mer's coverage value for contig and combining contig accuracy. In this research, we generate contig combining rules by use of the characteristics of *k*-mer's coverage value for contig.

3. Double assembly method with DAwCC

In the process of DAwCC, we used the set of contigs as input by several *k* values and methods. After extracting all pairs of contigs with more than 5 base overlaps, we determined binding contigs by the use of combining rules, binding accuracy to be higher.

Traditional assembly methods by use of machine learning are proposed as Jeong-Hyeon [16]. They applied machine learning and predicted short reads and contigs containing read-error by use of *p*-value of *k*-mer's coverage value for each contig. For the parameters that have a relationship with the discriminant, we had each paired contig's distribution of *p*-value for *k*-mer's coverage value. For constructing discriminant rules from these parameters, we focused on machine learning technique, especially on the supervised learning. In order to obtain training data, we generated artificial contig data by use of all pairs of contig with overlap region more than 5 bases and determined them consistent or inconsistent by comparing to the original sequence. We obtained the discriminant rules by applying machine learning to training data.

As for the parameters that have a relationship with discriminant, we had *p*-value's distribution of *k*-mer's coverage for each former and latter contig. We focused on actuation, frequency distribution and correlation of distribution of *p*-value in this research. Firstly, because we can regard *p*-value's distribution as waveform as Fig. 3~Fig. 6, we applied Fourier transform [17] used in the field of frequency analysis. High and low frequency component of Fourier transform and powered value of them were used to represent the contour of waveform. Furthermore, because actuation information for each position are important, we transformed *p*-value for each position $q_{i(i=1,\ldots,n)}$ with accordance in (2) and defined $Coef_{wav}$ as the gradient of a waveform as (3).

$$q_{i(i=1,\ldots,n)} = \begin{cases} -1 \ (i > i\text{-}1) \\ 0 \ \ (i = i\text{-}1) \\ 1 \ \ (i < i\text{-}1) \end{cases} \qquad (2)$$

$$Coef_{wav} = \sum_{i=1}^{n} q_i \qquad (3)$$

In addition, we defined the rate of increase value in all elements *Q* as (4).

$$R_{inc} = \frac{|\{q_i \in Q \mid q_i = 1\}|}{|Q|} \qquad (4)$$

Secondly, we applied the feature of frequency distribution of *p*-value (range=0.1). We defined p_{null} meaning *p*-value with null frequency value, and powered value of the Fourier transform of frequency distribution. Finally, we applied correlation of *p*-value's distribution such as maximum cross-correlation function, correlation coefficient and the hamming distance of frequency distribution of *p*-value. In addition, we added norm value of end point of former contig's *p*-value and start point of latter contig's *p*-value. Table 1 shows the list of feature parameters.

Table 1: List of feature parameters by use of *p*-value

actuation	$Coef^{f,l}_{wav}$: Gradient of waveform
	$R^{f,l}_{inc}$: Rate of increase value
	$F^{f,l}_{high}$: High-frequency component of Fourier transform
	$Sum^{f,l}_{f}$: Powered value of Fourier transform
	$F^{f,l}_{low}$: Low-frequency component of Fourier transform
distribution	$p^{f,l}_{null}$: *p*-value with null frequency value
	$Sum^{f,l}_{F,freq}$: Powered value of Fourier transform for frequency distribution
	CC : Correlation coefficient
correlation	CC_{freq} : Correlation coefficient of frequency distribution
	CCF_{freq} : Maximum cross-correlation function for Fourier transform of frequency distribution
	M^{F}_{ccf} : Maximum cross-correlation function for Fourier transform
	M_{ccf} : Maximum cross-correlation function
	D_{ham} :Hamming distance of frequency distribution
	M^{freq}_{ccf} : Maximum cross-correlation function for frequency distribution
	$\lvert p_{f,l} \rvert$: Norm value of end point of former and start point of latter

We used C4.5 [15] as decision tree to derive discriminant rules. Decision tree constructs effective rules from training data by weighting for accuracy of classifiers and output useful characteristic parameter for discriminant rules.

From the above, we proposed DAwCC (Double Assembly method with Characteristics of Contig for *k*-mer's coverage) by the following procedure.

step1 Prepare the whole sequence and read dataset whose base allocation is known.
step2 Obtain contigs from traditional assembly methods for some *k*-mers.
step3 Extract the all pairs of contigs with more than 5 base overlap region.
step4 Distinguish combined contigs consistent or inconsistent by comparing to original sequence, and generate training data by defining feature parameters.
step5 Derive discriminant rules by use of machine learning algorithm with training data that consists of parameters as step4 and consistency judgement.
step6 Apply combining rule to the result of DAwH and derive outputs satisfying rules.

Because we can derive correct combining rule and incorrect combining rule, we assigned each rule to generate the result of *k*-mers and assembly method, about two cases as follows.

case1 Eliminate combined contigs that don’t satisfy the correct combining rule.
case2 Eliminate the combinations that satisfy the incorrect combining rule.

Fig. 7 shows the flow of proposal method.

4. Experiments for comparison and availability

In order to confirm the effectiveness of DAwCC, we carried out comparative experiment to traditional method with single *k*-mer. We prepared adequacy of contig combining rules, evaluation indices of the experimental result, experimental data, result and discussion.

Firstly, in order to confirm adequacy of contig combining rules derived from training data, we calculated learning ability of them. With the combining rules from C4.5 for training data, we applied to the training data themselves and observed learning ability. Paired contigs satisfying correct combining rules regarded ‘correct combining’ and satisfying incorrect combining rules regarded ‘incorrect combining’. Because of discriminant errors, we classified combined contigs four types, correct combining judged as ‘correct combining’, correct combining judged as ‘incorrect combining’, incorrect combining judged as ‘correct combining’, incorrect combining judged as ‘incorrect combining’ as shown in Table 2.

Table 2: Discriminant result form

	correct	incorrect
consistent	**num1**	**num2**
inconsistent	**num3**	**num4**

num1 : contigs judged as correct for ”consistent”
num2 : contigs judged as correct for ”inconsistent”
num3 : contigs judged as for ”consistent”
num4 : contigs judged as incorrect for ”inconsistent”

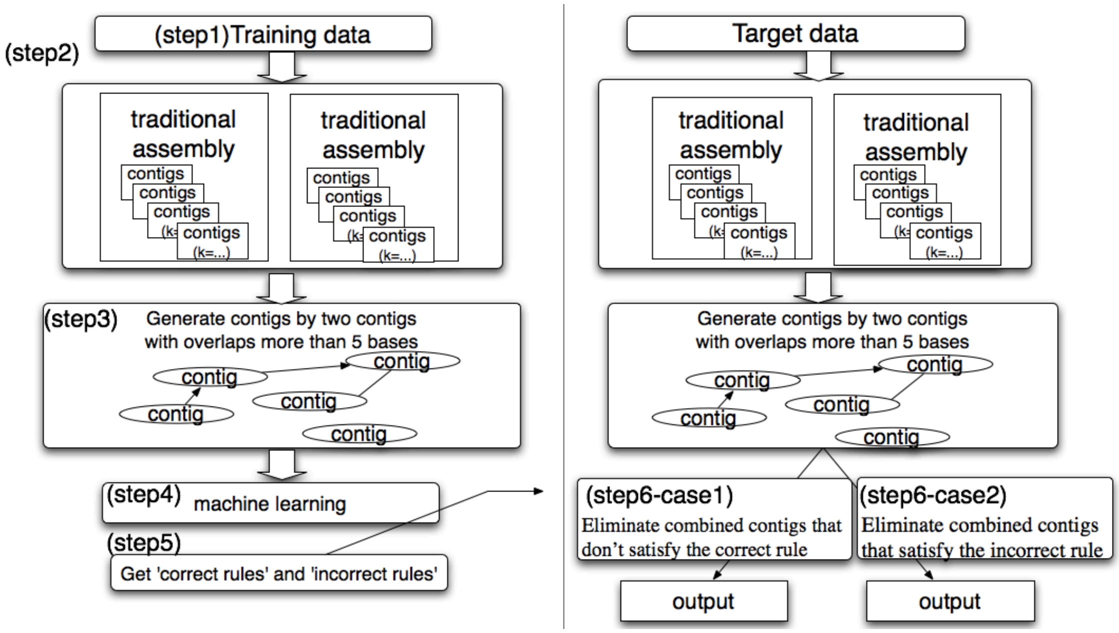


Fig. 7 Flow of proposal method

Because we can regard **num1** and **num4** as positive discriminant, we defined the learning ratio (*LeR*) by the ratio of positive discriminant in all combined contigs as (5).

$$LeR = 1 - \frac{num2 + num3}{num1 + num2 + num3 + num4} \qquad (5)$$

Secondly, we prepared 3 types experimental data, training data, test data, and simulated data in order to confirm effectiveness of contig combining rules for double assembly. Training data is the data to get contig combining rules and test data is from same reference as training data but consists of different *k*-mers. Simulated data is the data that is from different reference and *k*-mers from training data. We defined two indices to evaluate the performance of proposal. *CorR* is defined the rate of correct contigs (N_{cor}) for output contigs (N_{output}) as (6).

$$CorR = \frac{N_{cor}}{N_{output}} \qquad (6)$$

In addition, we defined *CovR* meaning the ratio of mapped contigs to the reference.

We prepared E.coli sequences as experimental data from NCBI[14], and derived contigs consists of four *k*-mers and two assembly methods. In order to prevent the occurrence of palindromic sequence, we can use only odd number for *k* value about Velvet, but ABySS also accepts even number. Therefore we applied Velvet and ABySS as traditional assembly methods for double assembly. Furthermore, *k* value has been assigned *k*=15~21 at the traditional assembly methods, as training data, we applied *k* =16,18 of ABySS and *k* =17,19 of Velvet and as test data, *k* =15,19 of ABySS and *k* =17,21 of Velvet. Furthermore, we prepared simulated data *k*=15,16 of ABySS and *k*=15,17 of Velvet from different E.coli sequence from training data or teat data as shown in Table 3.

Table 3: Detail of experimental data

	Training data		Test data		Simulated data	
Combination of Method	ABySS	Velvet	ABySS	Velvet	ABySS	Velvet
Combination of *k* value	*k* = 16,18	*k* = 17,19	*k* = 15,19	*k* = 17,21	*k* = 15,16	*k* = 15,17

Finally, we describe about experimental results and discussion. Table 4 shows discriminant result and learning ratio about combining rules for training data.

Table 4: Rule's learning effect for training data

	consistent	inconsistent	*LeR*
correct	531	6	0.971
incorrect	19	292	

We can find that rules from C4.5 could discriminant consistent and inconsistent contigs although with a little discriminant error from Table 4.

We got 4 correct combining rules and 18 incorrect combining rules from training data for C4.5. Table 5 shows rule list from decision tree obtained by training data. Each rule is constructed by parameters defined in training data and accuracy of them. For example, rule1 means that when the high frequency component of Fourier transform of former contig is less than 3.3, combined contig is correct with a probability of 99.0%.

Table 5: Combining rules

rule1	$F^{f}_{high} \leq 3.3$ -> correct combining[0.990]
rule2	$p^{f}_{mill} \leq 0.7$ and $p^{l}_{mill} \leq 0.2$ $F^{f}_{high} > 12$ and $F^{l}_{high} > 4.8$ -> correct combining[0.929]
rule3	$Sum^{l}_{f} > 1983.868$ and $F^{f}_{high} > 3752.9$ -> correct combining[0.909]
rule4	$Sum^{l}_{f} \leq 1983.868$ -> correct combining[0.658]
rule5	$Sum^{l}_{F} > 1983.868$ and $F^{f}_{high} > 3.3$ $F^{f}_{high} \leq 3752.9$ and $F^{f}_{low} \leq 0.7$ -> incorrect combining [0.976]
rule6	$M^{F}_{ccf} > 120142$ and $CCF_{freq} \leq 911099.5$ -> incorrect combining [0.969]
rule7	$CCF_{freq} \leq 911099.5$ and $D_{ham} > 4$ $p^{l}_{mill} > 0.2$ and $R^{l}_{inc} \leq 0.2285485$ $F^{f}_{high} > 3.3$ and $F^{l}_{low} > -5.5$ -> incorrect combining [0.968]
rule8	$CCF_{freq} \leq 6.56175e+07$ and $Sum^{l}_{F} > 5.823744$ $Sum^{l}_{F} \leq 1983.868$ and $F^{f}_{high} > 2275.2$ -> incorrect combining [0.967]
rule9	$\lvert p_{f,l} \rvert > -0.7$ and $\lvert p_{f,l} \rvert \leq 0.2$ $p^{l}_{mill} \leq 0.2$ and $F^{f}_{high} > 3.3$ $F^{f}_{high} \leq 12$ and $F^{l}_{high} > 6$ $F^{l}_{high} \leq 9.5$ -> incorrect combining [0.960]

rule10	$\lvert p_{f,l}\rvert > -0.7$ and $\lvert p_{f,l}\rvert \le 0.8$ $p^{l}_{null} \le 0.2$ and $Sum^{l}_{F} \le 5.235331$ $F^{f}_{high} > 3.3$ and $F^{f}_{high} \le 12$ $F^{f}_{low} > 4.8$ -> incorrect combining [0.957]
rule11	$CCF_{freq} > 99$ and $R^{f}_{inc} \le 0.218525$ $F^{f}_{high} \le 4.8$ and $F^{f}_{low} \le 0.3179367$ -> incorrect combining [0.950]
rule12	$Coef^{f}_{wav} \le 0$ and $p^{l}_{null} > 0.1$ $CC \le 2.15$ and $Sum^{f}_{F} \le 6.59273$ $F^{f}_{high} > 3.3$ -> incorrect combining [0.947]
rule13	$CCF_{freq} \le 87884.5$ and $p^{l}_{null} \le 0.2$ $Sum^{l}_{F} > 15.59678$ and $F^{f}_{high} > 3.3$ -> incorrect combining [0.933]
rule14	$M^{F}_{ccf} \le 56.88169$ and $p^{l}_{null} > 0.2$ $p^{l}_{null} \le 0.5$ and $F^{f}_{high} > 3.3$ $F^{f}_{low} > 6.3$ -> incorrect combining [0.933]
rule15	$\lvert p_{f,l}\rvert > 0.2$ and $R^{l}_{inc} \le 0.1666667$ $F^{f}_{low} > 4.8$ -> incorrect combining [0.929]
rule16	$M^{F}_{ccf} \le 133.7969$ and $p^{l}_{null} \le 0.2$ $F^{f}_{high} > 3.3$ and $F^{f}_{low} > 15.5$ -> incorrect combining [0.929]
rule17	$CCF_{freq} \le 400365$ and $Coef^{l}_{wav} > 10$ $F^{f}_{high} > 3.3$ -> incorrect combining [0.923]
rule18	$M^{F}_{ccf} > 19.95666$ and $CCF_{freq} > 99$ $CC \le 2.15$ and $Sum^{f}_{F} \le 17.03178$ $Sum^{l}_{F} \le 4.8$-> incorrect combining [0.920]
rule19	$CCF_{freq} \le 87884.5$ and $p^{f}_{null} > 0.7$ $p^{l}_{null} \le 0.2$ and $CC_{freq} \le 0.6083328$ $F^{l}_{high} > 4.8$ -> incorrect combining [0.920]
rule20	$M^{F}_{ccf} \le 133.7969$ and $p^{l}_{null} \le 0.2$ $R^{f}_{inc} \le 0.1764706$ and $Sum^{l}_{F} > 7.45922$ $F^{f}_{high} > 3.3$ -> incorrect combining [0.917]
rule21	$\lvert p_{f,l}\rvert \le -0.5$ and $M^{F}_{ccf} > 56.88169$ $CC_{freq} \le 0.4219435$ and $F^{f}_{high} > 3.3$ $F^{l}_{high} > 4.8$-> incorrect combining [0.857]
rule22	$M^{F}_{ccf} > 5085.468$ and $F^{l}_{high} \le 4.8$ -> incorrect combining [0.800]

Furthermore, Table 6 shows eight referenced feature parameters during combining rule generation with citation rate in the top. We can say that feature parameter with higher citation rate is more effective parameter.

Table 6: Referenced feature parameters during rule generation in the top

99.29% Sum^{l}_{F} 39.50% F^{f}_{high} 22.05% F^{l}_{high}
20.64% p^{l}_{null} 18.75% CCF_{freq} 14.15% M^{F}_{ccf}
11.91% $\lvert p_{f,l}\rvert$ 5.54% $F^{f,l}_{low}$

We found that powered value of the Fourier transform for latter contig was most cited characteristic parameter. In other words, waveform information of *p*-value for *k*-mer's coverage may be deeply involved in the accuracy of contig combining.

Subsequently, we compared the performance of traditional assembly method and proposal by use of several indices listed in Table 5. We applied ABySS, Velvet with single *k*-mer, DAwH with multiple *k*-mers as compared. We applied obtained rules to training data to confirm the effectiveness of combining rule by comparing DAwH meaning without rule and DAwCC in Table 7 by use of *CorR* and *CovR*.

Table 7: Effectiveness of rule for training data

	Output	correct	incorrect	*CorR*	*CovR*
DAwH	848	537	313	0.63	1.0
DAwCC (correct rule)	586	376	210	0.64	0.999
DAwCC (incorrect rule)	392	370	22	0.94	0.999

We found that *CorR* was more than 30 % improved from by comparing result about DAwH and DAwCC especially about using incorrect combining rule (94%), and confirmed the effect of rule for training data.

Next, we compared proposal method and DAwH and traditional method by use of specified *k*-mer for test data and simulated data. Table 8 shows the comparative result about test data. We can find that there are some biases for *CovR* about the results of specific *k* value and method. *CovR* was improved with DAwH, however *CorR* was 36 % worse from Table 8. But, *CorR* was improved with keeping *CovR* about DAwCC, especially with the incorrect combining rule, 17 % improved.

Table 8: Effectiveness of rule for test data

	Output	correct	incorrect	CorR	CovR
Velvet (*k* =17)	13	13	0	1.0	0.98
Velvet (*k* =21)	9	9	0	1.0	0.98
ABySS (*k* =15)	66	66	0	1.0	0.93
ABySS (*k* =19)	38	38	0	1.0	0.76
DAwH (*k*=V17,21+ A15,19)	597	387	210	0.64	1.0
DAwCC (correct rule)	558	374	184	0.67	1.0
DAwCC (Incorrect rule)	402	325	77	0.81	1.0

Furthermore, Table 9 shows the comparative result about simulated data. We can find that there are some biases for *CovR* about the result of specific *k* value and method as test data. *CovR* was improved with DAwH, however *CorR* was 33 % worse. But it was improved with keeping *CovR* about DAwCC especially about applying the correct combining rule, 9 % improved. We confirmed effectiveness of combining rule for simulated data although inferior to test data.

Table 9: Variation of *CorR* and *CovR* for simulated data

	Output	correct	incorrect	*CorR*	*CovR*
Velvet (*k*=15)	8	8	0	1.0	0.99
Velvet (*k*=17)	3	3	0	1.0	0.99
ABySS (*k*=15)	21	21	0	1.0	0.97
ABySS (*k*=16)	8	8	0	1.0	0.97
DAwH (*k*=V15,17+ A15,16)	142	96	46	0.67	0.999
DAwCC (correct rule)	69	53	16	0.76	0.999
DAwCC (Incorrect rule)	60	44	9	0.73	0.999

From experimental results, we could obtain assembly results independent specific *k* value by use of combining rules about test data similar to training data from Table 8. Finally, we confirmed same effect of proposal about simulated data, so Table 9 shows some robustness of proposal.

5. Conclusion

In order to derive robust assembly result independent of *k* value and assembly method, we proposed double assembly method merging results of different *k*-mers and traditional methods with contig combining rule. We found that there was relation between distribution of *k*-mer's coverage value and accuracy of contig combining from pre-experiment. With that, we generated traing data constructed by feature parameter about distribution of *p*-value for *k*-mer's coverage value of contig, and derived combining rules. We used decision tree (C4.5) as machine learning and confirmed effectiveness of rules about training data by improvement of *CorR* compared to DAwH. From the comparative experimental results for test data and simulated data, we confirmed that coverage ratio and correct ratio were improved by the proposed method than traditional methods with specific *k*-mer and DAwH. Furthermore, effective characteristic parameters were powered-value of the Fourier transform for latter contig, so we confirmed that information of distribution about *p*-value has deep involvement for accuracy of contig combining.

References

[1] Lincoln D Stein : The case for cloud computing in genome informatics. *Genome Biology*, 11 (2010), 207.

[2] Miller JR, Koren S, Sutton G : Assembly algorithms for next-generation sequencing data. *Genomics*, 95(2010), 315-327.

[3] Xiaohong zhao, Lance E. Palmer : EDAR(An Efficient Error Detection and Removal Algorithm for Next Generation Sequencing Data) , (2010).

[4] Rene L. Warren , Granger G. Sutton , Steve J. M. Jones and Robert A. Holt : Assembling millions of short DNA sequences using SSAKE, *Bioinformatics*, 23 (2007), 500-501

[5] Daniel R. Zerbino and Ewan Birney : Algorithms for de novo short read assembly using de Bruijn graphs, *Genome Research*, 18 (2008), 821- 829.

[6] Daniel R. Zerbino, Gayle K. McEwen, Elliott H. Margulies, Ewan Birney : Pebble and Rock Band: Heuristic Resolution of Repeats and Scaffolding in the Velvet Short-Read de Novo Assembler

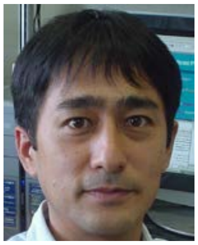
[7] Jared T. Simpson, kim Wong, Shaun D. Jackman, et al ABySS : A parallel assembler for short read sequence data, *Genome Research*, 19(2009), 1117-1123.

[8] Yu Peng, Henry Leung, S.M. Yiu, Francis Y.L. Chin : IDBA - A Practical Iterative de Bruijn Graph De Novo Assembler, Research in Computational Molecular Biology, 14th Annual International Conference, *RECOMB* 6044 (2010), 426-440.

[9] Jurgen Nijkamp, Wynand Winterbach, Marcel van den Broek, Jean-Marc Daran, Marcel Reinders, and Dick de Ridder : Integrating genome assemblies with MAIA, *ECCB*, 26 (2010), 433–439.

[10] Guohui Yao. Liang Ye, Hongyu Gao, Patrick Minx, Wesley C. Warren, George M. Weinstock : Graph accordance of next-generation sequence assemblies, 28(2012),13-16.
[11] Alesky V. Zimin, Douglas R. Smith, Granger Sutton : Assembly reconciliation, *Bioinformatics*, 24 (2008),42-45
[12] Jared T. Simpson and Richard Durbin : Efficient de novo assembly of large genomes using compressed data structures *Genome Research* 22 (2012) 549-556
[13] Ayako Ohshiro, Takeo Okazaki, Hitoshi Afuso, Morikazu Nakamura : A study of double assembly method for DNA sequences, *IPSJ SIG* 33(2013)
[14] National Center for Biotechnology Information : http://www.ncbi.nlm.nih.gov/
[15] Quinlan, J. R : C4.5 Programs for Machine Learning (1993)
[16] Thomas G. Dietterich : Machine-Learning Research Four Current Directions (AAAI) *AI Magazine*, 18 (1997)

Ayako OHSHIRO received the B.S. and M.S. degrees in Information Engineering from University of the Ryukyus in 2009 and 2011, respectively. She belongs to doctoral course in same university. Her research area is Bioinformatics. Especially, she is interested in development of DNA assembly algorithm.



Takeo OKAZAKI took B.Sc. and M.Sc. from Kyushu University in 1987 and 1989, respectively. He had been a research assistant at Kyushu University from 1989 to 1995. He has been a lecturer at University of the Ryukyus since 1995. His research interests are statistical data normalization for analysis, statistical causal relationship analysis.

Morikazu NAKAMURA took B.E. and M.E. from University of the Ryukyus in 1989 and 1991, respectively. He took Ph.D. from Osaka University in 1995. He has been a professor at University of the Ryukyus. His research interest includes design and analysis of parallel and distributed algorithms.