

Genome analysis

## A machine-learning approach to combined evidence validation of genome assemblies

Jeong-Hyeon Choi<sup>1,\*</sup>, Sun Kim<sup>1,2</sup>, Haixu Tang<sup>1,2</sup>, Justen Andrews<sup>1,3</sup>, Don G. Gilbert<sup>1,3</sup> and John K. Colbourne<sup>1</sup>

<sup>1</sup>The Center for Genomics and Bioinformatics, <sup>2</sup>School of Informatics and <sup>3</sup>Department of Biology, Indiana University, IN 47405, USA

Received on October 9, 2007; revised on November 29, 2007; accepted on December 5, 2007

Advance Access publication January 18, 2008

Associate Editor: Alex Bateman

### ABSTRACT

**Motivation:** While it is common to refer to ‘the genome sequence’ as if it were a single, complete and contiguous DNA string, it is in fact an assembly of millions of small, partially overlapping DNA fragments. Sophisticated computer algorithms (*assemblers* and *scaffolders*) merge these DNA fragments into contigs, and place these contigs into sequence scaffolds using the paired-end sequences derived from large-insert DNA libraries. Each step in this automated process is susceptible to producing errors; hence, the resulting *draft* assembly represents (in practice) only a likely assembly that requires further validation. Knowing which parts of the draft assembly are likely free of errors is critical if researchers are to draw reliable conclusions from the assembled sequence data.

**Results:** We develop a machine-learning method to detect assembly errors in sequence assemblies. Several *in silico* measures for assembly validation have been proposed by various researchers. Using three benchmarking *Drosophila* draft genomes, we evaluate these techniques along with some new measures that we propose, including the good-minus-bad coverage (GMB), the good-to-bad-ratio (RGB), the average Z-score (AZ) and the average absolute Z-score (ASZ). Our results show that the GMB measure performs better than the others in both its sensitivity and its specificity for assembly error detection. Nevertheless, no single method performs sufficiently well to reliably detect genomic regions requiring attention for further experimental verification. To utilize the advantages of all these measures, we develop a novel machine learning approach that combines these individual measures to achieve a higher prediction accuracy (i.e. greater than 90%). Our combined evidence approach avoids the difficult and often *ad hoc* selection of many parameters the individual measures require, and significantly improves the overall precisions on the benchmarking data sets.

**Availability:** <http://people.cgb.indiana.edu/jeochoi/gav/>

**Contact:** jeochoi@indiana.edu

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

### 1 INTRODUCTION

Since the shotgun strategy was first introduced by Sanger and colleagues in sequencing the genome of bacteriophage  $\lambda$  (Sanger *et al.*, 1982), significant progress has been made in applying this method to progressively larger genomes. These improvements are mainly because of the rapid advancement in DNA sequencing technologies and in the algorithmic development for DNA fragment assembly. Today, the whole genome shotgun (WGS) sequencing and fragment assembly are applied to entire eukaryotic genomes in a near-fully automatic fashion; compared to phage genomes of ~2–200 kilobases, some sequenced eukaryotic genomes contain several billion base pairs, like those of human (Venter *et al.*, 2001), mouse (Mouse Genome Sequencing Consortium, 2002), dog (Lindblad-Toh *et al.*, 2005), chicken (International Chicken Genome Sequencing Consortium, 2004), and opossum (Mikkelsen *et al.*, 2007). Yet despite its high throughput and low cost at reaching a draft genome sequence assembly, WGS sequencing is still ineffective at ultimately achieving an accurate and complete genome sequence.<sup>1</sup> A finished genome project requires extensive (and expensive!) efforts to validate the draft assembly and to fill in sequence gaps (Green, 1997). As a result, most WGS assembly are released in ‘draft’ form, whose quality is seldom reported and largely unknown.

Although draft genome sequences are incomplete, they are extremely valuable for biologists, especially for the functional and evolutionary study of protein coding genes. However, recent analyses show disturbingly large numbers (from hundreds to thousands for each genome) of potential mistakes in draft genome sequences (Salzberg and Yorke, 2005). Note that these mistakes are not individual base-calling errors that are often corrected with additional sequencing. These errors relate to assembled sequence fragments (*mis-assemblies*) that incorrectly delete or wrongly arrange the location and/or orientation of long stretches of DNA. Most of these mis-assemblies result from the *repetitive* DNA, which is a common feature of large eukaryotic genomes and some microbial genomes. Fragment assemblers mainly follow the ‘overlap-layout-consensus’

<sup>1</sup>According to the Bermuda standard, a complete genome means a single DNA sequence (with no gap) for each chromosome, containing no more than 1 in 10 000 (0.01%) erroneous or ambiguous bases.

\*To whom correspondence should be addressed.

paradigm (Bonfield *et al.*, 1995; Kececioglu and Myers, 1995; Kim *et al.*, 2007; Myers, 1995). Since many repeated DNA segments have nearly identical sequences, assemblers mistakenly overlap sequence reads belonging to copied regions of the genome (Pop *et al.*, 2002). With *pseudo-overlaps* incorrectly promoted to the layout step, the assemblers can create large-scale rearrangements of DNA segments in the final consensus genome sequence (Tang, 2007).

Unlike the base-calling errors, mis-assemblies are rarely fixed, since the finishing projects mainly focus on generating additional reads targeting gaps in the genome sequence and attempt not to validate the existing assemblies. Furthermore, even if the finishing projects were to experimentally verify assemblies, until recently, there is no automatic method to help guide these experiments toward areas deserving attention (Nelson *et al.*, 2005; Schmutz *et al.*, 2003, 2004; West *et al.*, 2006). However, there are several types of measures that are known to be helpful to identify assembly errors. For example, assembly errors often appear within the regions with low read coverage (RC), containing chimeric or recombined reads, having wrongly oriented paired end reads, or paired end reads with compressed distances. Based on these commonly observed characteristics, several simple measures are suggested for assembly validation, such as measuring the RC and the clone coverage (CC) (Table 1). More sophisticated methods include proposals made by Kim and Liao (Kim *et al.*, 2001), which are based on the null distributions built from randomly sampled reads, and an entropy measure within contigs derived from the probabilistic models. Sutton and colleagues detected the breakpoints between mis-assembled DNA segments by scanning the number of unsatisfied mate-pairs, i.e. the paired reads from clone ends that show distances (measured in kilobases) within the assembly that deviate from the known distribution of insert sizes in sequenced genomic libraries (Dew *et al.*, 2005). Yorke and colleagues propose the compression/expansion (CE) statistics for unsatisfied mate-pairs, and identify the regions containing potentially collapsed repeats (Zimin *et al.*, 2005). Finally, visualization tools for assembly validation are also developed. For example, BACCardI is a graphical tool for the construction of virtual clone maps by using paired-end reads (Bartels *et al.*, 2005). Hawkeye is a visual analytic tool for fragment assembly, which can be used to aid in manually finding and correcting mis-assemblies (Schatz *et al.*, 2007).

In this article, we propose several new measures for assembly validation. We compare their performance with those of existing measures, by evaluating them using three benchmarking *Drosophila* draft genomes. Our results show that, although the new measures are more accurate than the existing ones, the performance of any individual measure is not satisfactory for all assembled genomes. To improve on the potential strengths of each formulation in general applications, we develop a machine learning approach that combines multiple measures, and we demonstrate that this combined evidence approach is far better than any individual measure, and generally achieves acceptable results in genome assembly validation. The implementation of this algorithm provides a useful software tool for the genome sequencing communities and for biologists to gain confidence in the draft genome sequences.

**Table 1.** Summary of individual measures for detecting potential errors in draft genome assemblies, including several new measures proposed here (see text for detailed descriptions)

Abbr.	Measure
RCN and RCX	Minimum and maximum RC
CCN and CCX	Minimum and maximum CC
CE	Compression/expansion statistic
GMB	Number of good clones minus number of bad clones
RGB	Ratio of good and bad clones
AZ	Average z-scores
ASZ	Maximum of absolute values of average positive Z-scores and average negative Z-scores

## 2 METHODS

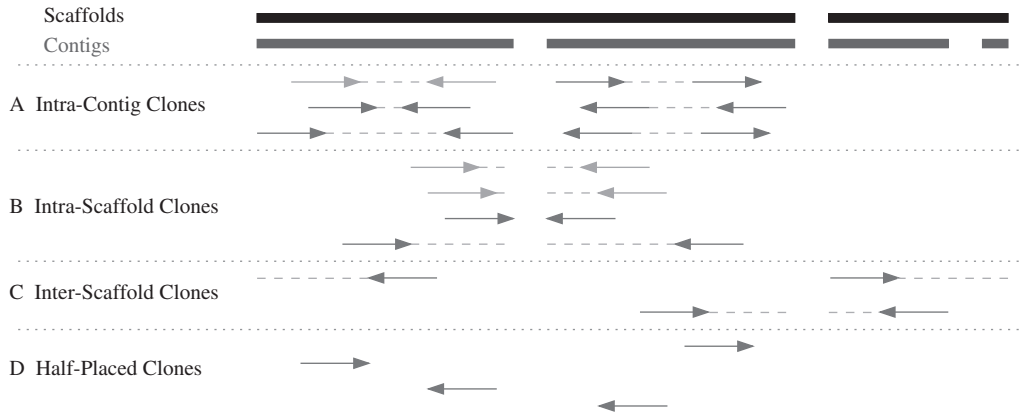
### 2.1 Individual measures for assembly validation

We explore the performance of five existing and four new measures on assembly validation. Table 1 summarizes the definition of all nine individual measures.

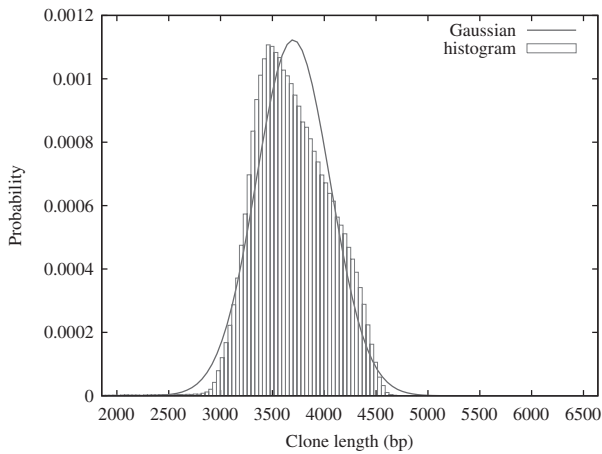
RC and CC are two basic measures that count the number of *local* reads and clones (i.e. paired-end reads), respectively, which span a specific segment in the genome assembly. The deviation of the RC/CC from the average coverage along the entire genome (e.g. RC=8, for a typical shotgun genome sequencing project) indicates a putative mis-assembly within this segment, such as the collapse of repeats or insertion of a DNA segment. As an example that utilizes this information, CE statistic (Zimin *et al.*, 2005) computes the length distribution of the clones spanning specific genomic segment, and compares it with the length distribution from all assembled clones. To define CE precisely, we begin by classifying clones into four groups (Fig. 1):

- intra-contig clones: paired-end reads placed within the same contig;
- intra-scaffold clones: paired-end reads placed within the same scaffold, yet anchored in different contigs;
- inter-scaffold clones: paired-end reads placed among different scaffolds; and
- half-placed clones: one of the paired-end reads placed in the assembly, but not the other.

For each size-specific DNA library, the clone length distribution is calculated based on the respective intra-contig clones by counting the number of nucleotides spanning the length of the assembly between the paired-end reads. Not surprising, clone length distributions best fit a Gaussian model with narrow skews as shown in Figure 2. After we obtain the overall length distributions, we compute the mean and standard deviation of the clone lengths. Then, the CE statistics are defined as the magnitude in standard deviations that the average length of *local* clones differs from the average length of all clones for each clone library (Zimin *et al.*, 2005). Similar to the CE statistics, the Z-score is defined as the number of standard deviations that the length of one clone differs from the average length of all clones from the same clone library. An intra-contig or intra-scaffold clone is called *good* if the absolute Z-score of its length is smaller than a threshold (e.g. 2), or otherwise the clone is called *bad* as shown in Figure 1. All half-placed clones and clones with paired-end reads that are placed in the same or outer orientation (Fig. 1A) are also called *bad*. After good and bad clones are classified, we compute the measure of good-minus-bad (GMB) by subtracting the number of bad clones from the number of good clones that span a specific genomic



**Fig. 1.** Clone classification. (A) intra-contig clones; (B) intra-scaffold clones; (C) inter-scaffold clones; (D) half-placed clones. The green and red lines represent good and bad clones, respectively.



**Fig. 2.** Distribution of clone lengths deduced from locations of paired-end reads placed in the 2004 draft assembly of *Drosophila virilis*. The Gaussian distribution ( $\mu = 3699$ ,  $\sigma = 355$ ) fitting the library of 3704 bp is shown in the blue dotted line.

segment in the assembly. Similarly, the measure of good-to-bad ratio (RGB) is computed by the logarithmic ratio between the numbers of good and bad clones. Finally, the measure of average Z-score (AZ) is computed by averaging the Z-scores of local clones, and the measures of positive and negative Z-scores (ASZ) are computed by averaging the positive and negative Z-scores, respectively. GMB was tested previously on the assembly of a small bacterial genome, *Mycoplasma genitalium*, using Phrap. In this article, we extend and improve the method for assemblies of larger genomes.

## 2.2 Machine-learning approach to combined evidence assembly validation

We test five different machine-learning algorithms that are implemented in the Weka package (Witten and Eibe, 2001), including the decision tree (J48) (Quinlan, 1993), the random forest (RF) (Breiman, 2001), the random tree (RT) (Dietterich, 2000), the naive in french bayes classifier (NB) (Duda and Hart, 1973) and the Bayesian network (BN) (Heckerman et al., 1995). For each method, we supply three kinds of features: coverage statistics, length statistics and repeat measurements. The coverage statistics include RC, CC, GMB and RGB; the length statistics include average Z-score, average positive and negative Z-scores

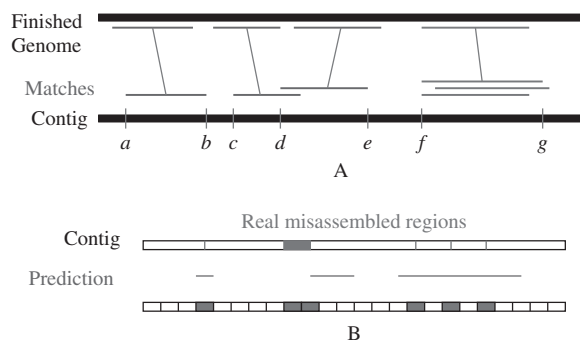
and CE statistics; the repeat measurements include the number of repeats identified by RepeatMasker, and by a self-comparison of the genomic segment.

## 2.3 Evaluation of predicted mis-assembled regions

To evaluate the assembly validation methods, we use both simulated datasets and draft assemblies from eukaryotic genome projects.

For each simulated dataset, we first generate a random DNA sequence of length 10 Mbp with 70% GC content, then insert multiple copies of nearly identical (~99%) repeats with total length of 3.5 Mbp, consisting of 200, 1000, 2000 and 5000 copies of repeats with length 5000, 1000, 500 and 100, respectively. The average difference between repeat copies was set as 1% substitutions and 1% indels. Afterwards, we sampled randomly 107994 (i.e. coverage ~8) paired-end reads with expected distances of 3, 10 and 150 Kbp which were allowed 2% variance. The clone length which is the distance between the pairs of reads are drawn from an even distribution. The reads are sampled with expected length of 1000 which were allowed 2% variance. Their quality scores are assigned to the same value. The sequencing error rates are assumed uniform across the reads, and were set at various levels (i.e. 0.001, 0.003 and 0.005) to test the performance of validation methods under different conditions. Finally, the assemblies of these simulated reads were automatically generated using the Arachne assembler (Batzoglou et al., 2002) which are used for evaluating the assembly validation methods.

In addition to the simulated data, we also use the draft assemblies of three *Drosophila* species, i.e. *D. mojavensis*, *D. erecta* and *D. virilis*, for the evaluation purpose. For each genome, we choose two versions of draft assemblies that are generated using Arachne in 2004 and 2005, respectively. The mis-assemblies in the test draft assemblies are determined by aligning the assembled contigs with the corresponding final genome sequences, downloaded from Drosophila Assembly/Alignment/Annotation Website (<http://rana.lbl.gov/drosophila/>). After alignment, each contig in the draft assembly may contain *matched* and *unmatched* regions (Fig. 3). A *matched* region may be further classified as (1) a *unique* match, e.g. the segment  $[a, b]$  (Fig. 3A), (2) two or more overlapping matches, e.g. the segment  $[c, e]$  or (3) a match along with the other alternative matches, e.g. the segment  $[f, g]$ . We then define two classes of *breakpoints*: (1) the breakpoints corresponding to the overlapping matches, e.g. around  $d$  with overlapping matches  $[c, d]$  and  $[d, e]$ ; (2) the breakpoints defined by unmatched regions, e.g.  $[b, c]$  and  $[e, f]$ . These breakpoints are considered *true* mis-assemblies, and are used as reference to evaluate assembly validation methods. We note that these breakpoints typically represent very short genome regions, hence,



**Fig. 3.** Determining true mis-assembly regions by comparing the draft assembly with a finished genome. **(A)** Classification of mis-assembly breakpoints. The matches between a contig in the draft assembly and the finished genome are represented by the blue and red lines. The first match is unique. Therefore, there is no evidence of mis-assembly. The second and third matches along the contig have overlapping matches, which represent mis-assembly regions. In total, seven breakpoints (*a–g*) are considered true mis-assemblies from this evaluation. **(B)** Uncertainties up to 500 bp are allowed in the predicted mis-assembly breakpoints for a fair evaluation of the validation methods. To achieve this, a contig is split into blocks of 500 bp, and the number of blocks for true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) are counted at the block level. For this example, if the blue lines are predicted mis-assembly regions by a validation method and the red blocks are true mis-assembled blocks, we count 5 TPs, 7 FPs, 1 FN and 11 TNs, respectively.

it is difficult to predict their accurate positions within mis-assembled regions (Fig. 3B). Therefore, we consider a prediction (of mis-assembly) to be true if it is located within 500 bp from an actual breakpoint.

The draft assemblies of all three *Drosophila* genomes are downloaded from the AAA (Assembly, Alignment and Annotation) of the now 12 sequenced *Drosophila* genomes website at <http://rana.lbl.gov/drosophila/>. The statistics of these published genome sequence assemblies and the versions of draft assemblies are listed in Table 2 and Supplementary Table 1. The final assemblies in CAF1 (Comparative Analysis Freeze 1) which are used as the finished genome sequences in our experiments are reconciliations of independent assemblies performed using Arachne and the Celera Assembler (Zimin *et al.*, 2005). The statistics of these genomes are listed in Table 3, which are published on FlyBase (<http://flybase.bio.indiana.edu/>).

## 2.4 Implementation details

We implemented the methods described above in C++ and PERL. The whole program consists of three steps. First, we compute all single measures for given training and testing draft assemblies from the input of read layout (in ACE or Washington University format) and mate-pair information (in table-delimited or XML format used by NCBI Trace Archive). Next, we analyze the repetitive structure of the draft assembly by RepeatMasker and a self-comparison using BLAST and MUMmer. Finally, based on these pre-computed measures, we predict putative mis-assembly regions using weka package with prepared data in the first and second steps.

## 3 RESULTS

### 3.1 Performance of nine individual measures

We first tested the performance of these measures on the draft assemblies of three *Drosophila* genomes. The results are

**Table 2.** Statistics of draft assemblies for *D. erecta*, *D. mojavensis* and *D. virilis*

	2005 Aug <i>D.moj</i>	2005 Aug <i>D.ere</i>	2005 Aug <i>D.vir</i>
N. of reads	2 717 401	2 728 578	3 320 772
L. of reads (bp)	1 838 125 872	1 885 877 180	1 939 679 395
Coverage*	9.5	12.3	9.4
N. of clones	1 074 817	1 137 308	1 085 481
N. of half-placed clones	164 347	59 658	224 235
N. of contigs	12 351	7 759	19 138
Length of contigs (bp)	180 519 631	145 196 048	189 914 823
N. of n50 contigs	437	95	475
L. of n50 contigs (bp)	100 418	365 805	101 385
N. of scaffolds	5 124	13 562	
L. of scaffolds (bp)	194 270 144	152 862 534	206 998 770
N. of N50 scaffolds	4	4	6
L. of N50 scaffolds (bp)	24 782 941	18 750 251	10 165 514
Total blocks	259 140	263 083	291 908
Mis-assembled blocks	12 959	6 985	10 848

\*Coverages are calculated based on the genome sizes in Gilbert (2007). N and L stand for number and length, respectively.

presented as Receiver Operating Characteristic (ROC) curves (Fig. 4). Among all individual measures, GMB reached the best performance (accuracy between 0.7 and 0.8). Yet the simple minimum clone coverage measure (CCN) achieved surprisingly high accuracy, by performing better than, or equally well to, more sophisticated measures, such as CE or Z-value-based measures in the 2005 assembly of *D. erecta*. Although the accuracy of some validation measures appears high, the precisions are extremely low, typically below 10%. Because the number of mis-assembled genomic regions is small compared to the number of correctly assembled regions, the predicted mis-assembled regions may contain mostly (>90%) false positives. It is impractical to rely on these individual measures to guide experiments for correcting errors in the assembly. We therefore attempt to improve the precision of assembly validation measurements to a reasonably high level (i.e. >30%) so to provide useful information for finishing efforts in genome projects.

### 3.2 Performance of combined evidence approach

We adopt a combined evidence approach that uses a machine-learning method to integrate the individual validation measures. We evaluate five different machine-learning classifiers that are implemented in the Weka: the decision tree (J48), the RF, the RT, the NB and the BN. Their relative performance are tested using both simulated data and draft assemblies of *Drosophila* genomes.

Table 4 summarizes the results of different machine-learning classifiers for the combined evidence assembly validation on simulated DNA sequences. The five machine-learning classifiers exhibit similar trends in their prediction accuracy across the varying sequence error rates used for our simulations; in most cases, the machine-learning classifiers perform better than

**Table 3.** Statistics describing the finished genome sequences (i.e. CAF1) of three *Drosophila* species that are used for benchmarking purposes in this article (FlyBase <http://flybase.bio.indiana.edu/>)

Abbr.	Species	T	Nt	E	Ne	%Gt	%Ge	% Gc	% Gaps	% Repeats
<i>D.mel</i>	<i>Drosophila melanogaster</i>	169	14	120	6	100.0%	100.0%	42.5%	0.0%	6
<i>D.ere</i>	<i>Drosophila erecta</i>	153	5124	125	7	99.4%	98.6%	42.3%	5.0%	19
<i>D.moj</i>	<i>Drosophila mojavensis</i>	194	6841	150	9	94.6%	92.7%	39.5%	7.1%	22
<i>D.vir</i>	<i>Drosophila virilis</i>	206	13530	140	15	94.8%	90.3%	40.0%	8.2%	28

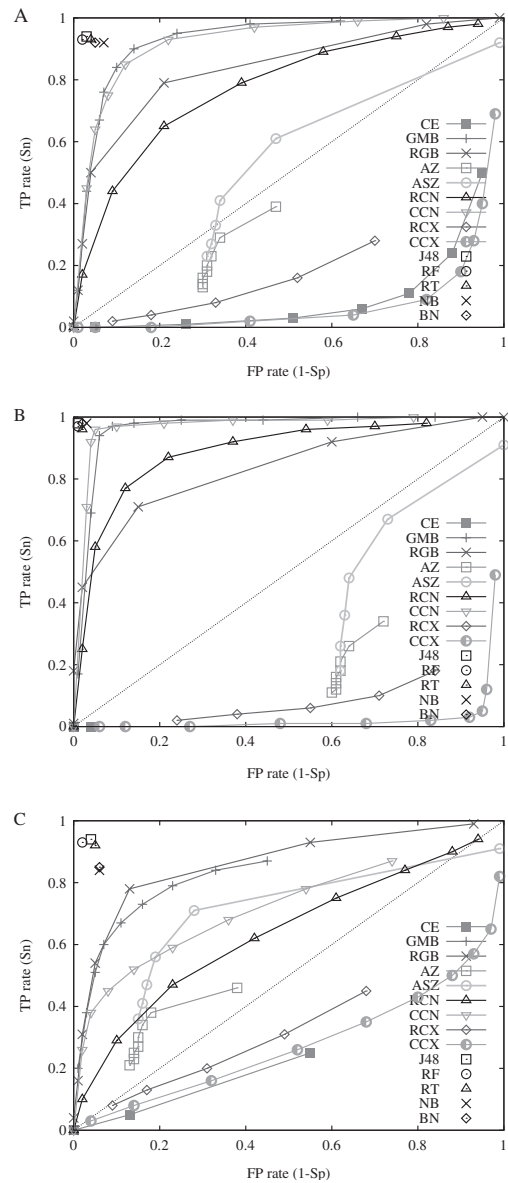
Abbreviations: size in Mbp of total assembly size (T) and euchromatic regions (E), numbers of total (Nt) and euchromatic scaffolds (Ne), percentages of *D.mel* genes matched in total (%Gt) and euchromatic (%Ge) scaffolds, GC content (%GC), estimated percentage of remaining gaps (%Gaps), and repeat coverage (% Repeats). Species are listed by increasing phylogenetic distance from *D.mel*.

**Table 4.** Results of different machine learning methods for the combined evidence assembly validation on simulated DNA sequences of about 13.5 Mbp that contain multiple copies of nearly identical repeats

Error rate	ML	TP	FN	FP	TN	PP	SN	SP
0.001	J48	53	15	709	14559	0.07	0.78	0.95
	RF	48	20	488	14786	0.09	0.70	0.97
	RT	48	20	1200	14055	0.04	0.70	0.92
	NB	63	5	2996	12194	0.02	0.92	0.80
	BN	57	11	2154	13064	0.03	0.84	0.86
0.003	J48	156	62	553	8335	0.22	0.72	0.94
	RF	132	86	454	8449	0.22	0.60	0.95
	RT	132	86	755	8135	0.15	0.61	0.92
	NB	207	11	3247	5502	0.06	0.95	0.63
	BN	185	33	1472	7363	0.11	0.85	0.83
0.005	J48	217	114	444	5991	0.33	0.66	0.93
	RF	186	145	374	6073	0.33	0.56	0.94
	RT	184	147	557	5881	0.25	0.56	0.91
	NB	316	15	3098	3150	0.09	0.95	0.50
	BN	280	51	799	5580	0.26	0.85	0.87

Various sequencing error rates (0.001, 0.003 and 0.005, respectively) are considered (see methods section and cross-species evaluation for details about the simulation and sampling procedures respectively). Abbreviations: machine learning classifiers (ML), number of true positives (TP), number of false negatives (FN), number of false positives (FP), number of true negatives (TN), precision (PP), sensitivity (SN), and specificity (SP).

individual measures (Supplementary Fig. 1). In the experiments employing higher sequencing errors, a great number of assembly errors are made by the Arachne assembler, while the validation methods achieve higher precisions, but slightly lower prediction accuracies. Overall, decision tree (J48) and random forest (RF) classifiers outperform (with higher accuracy and precision) the other classifiers across different experiments. Nevertheless, although the precision of the combined evidence approach is improved over the individual measures, it is still not great (e.g. only ~0.2–0.3), owing to the small total number of mis-assembled regions in the simulated data. The performance of combined evidence approaches compared to individual measures is shown in Figure 4, in relation to various machine learning applications used to validate the *Drosophila* genome draft assemblies. Most machine-learning classifiers outperform individual measures



**Fig. 4.** ROC curves of individual measures and the combined evidence approaches based on different machine learning methods on validating the draft assemblies of (A) *D. mojavensis*, (B) *D. erecta* and (C) *D. virilis* genomes, respectively.

**Table 5.** Results of a cross-evaluation of the machine learning approaches for validating the draft assemblies of *Drosophila* genomes

Species	ML	TP	FN	FP	TN	PP	SN	SP
<i>D.moj</i>	J48	6091	389	5925	201577	0.51	0.94	0.97
	RF	6021	459	4463	203860	0.57	0.93	0.98
	RT	6021	459	7837	201020	0.43	0.93	0.96
	NB	5953	527	13986	190119	0.30	0.92	0.93
	BN	5936	544	9455	194787	0.39	0.92	0.95
<i>D.ere</i>	J48	3417	76	3386	229717	0.50	0.98	0.99
	RF	3387	106	2600	231082	0.57	0.97	0.99
	Rt	3371	122	3861	230365	0.47	0.96	0.98
	NB	3425	68	7054	225139	0.33	0.98	0.97
	BN	3406	87	4490	227872	0.43	0.98	0.98
<i>D.vir</i>	J48	5118	307	9081	238663	0.36	0.94	0.96
	RF	5030	395	6125	242835	0.45	0.93	0.98
	RT	4979	446	11517	237915	0.30	0.92	0.95
	NB	4572	853	15601	230372	0.23	0.84	0.94
	BN	4611	814	14485	231404	0.24	0.85	0.94

The abbreviations and sampling follow Table 4.

on these real data sets. Again, random forest (RF) performs the best among the five machine-learning classifiers (with the highest precision) in all experiments (Table 5 and Supplementary Table 2, which are represented by dots in Fig. 4). The decision tree classifier reaches a slightly higher accuracy than RF, although its precision is lower. In general, the performance of RF is satisfactory (accuracy >0.9 and precision ~0.3) except for one case in which a limited number of true mis-assembled regions are identified (*D. erecta*, 2004 draft assembly).

### 3.3 Cross-species evaluation

To employ the machine-learning method, models must be trained using a learning data set of known true mis-assemblies. In the previous experiments, we trained the models by randomly sampling  $n$  blocks of the mis-assembled blocks and  $5n$  blocks of the correctly assembled blocks as learning set to train the model where  $n$  is 50% of the mis-assembled regions. We then used the remaining blocks to test the model. This procedure is impractical since in reality we expect to *de novo* validate the draft assembly for a whole genome, and we are unable to identify true breakpoints used for training. Therefore, a last experiment is performed, where we use one of the *Drosophila* genomes as the training set and test the model on the draft assembly of another genome. Table 6 and Supplementary Table 3 show the result from this cross-species evaluation. Similar to the previous experiments, decision tree (J48) is the most accurate classifier, whereas random forest (RF) reaches the highest precision.

## 4 DISCUSSION

In this article, we tested several individual measures for assembly validations. The results show that no single measure achieves satisfactory performance. Hence, we proposed a

**Table 6.** Results of a cross-species evaluation of the machine-learning approaches for assembly validating

Model	ML	TP	FN	FP	TN	PP	SN	SP
<i>D.moj</i>	J48	6811	174	5308	246323	0.56	0.98	0.98
	RF	6695	290	3414	249921	0.66	0.96	0.99
	RT	6696	289	13737	239968	0.33	0.96	0.95
	NB	6928	57	34957	213981	0.17	0.99	0.86
	BN	6907	78	13478	235543	0.34	0.99	0.95
<i>D.ere</i>	J48	7622	3226	5926	272447	0.56	0.70	0.98
	RF	6771	4077	4099	276010	0.62	0.62	0.99
	RT	8378	2470	13720	266131	0.38	0.77	0.95
	NB	8902	1946	39189	233196	0.19	0.82	0.86
	BN	8899	1949	24257	248488	0.27	0.82	0.91
<i>D.vir</i>	J48	9795	3164	5137	235447	0.66	0.76	0.98
	RF	9427	3532	3693	239027	0.72	0.73	0.98
	RT	9722	3237	8654	235679	0.53	0.75	0.96
	NB	6840	6119	2988	241949	0.70	0.53	0.99
	BN	10215	2744	5053	233086	0.67	0.79	0.98
<i>D.ere</i>	J48	5595	5253	7185	271088	0.44	0.52	0.97
	RF	4987	5861	5383	273505	0.48	0.46	0.98
	RT	6087	4761	12605	267211	0.33	0.56	0.95
	NB	6304	4544	13712	262433	0.31	0.58	0.95
	BN	5836	5012	9699	267384	0.38	0.54	0.96
<i>D.vir</i>	J48	6838	147	19555	231898	0.26	0.98	0.92
	RF	6711	274	7748	245201	0.46	0.96	0.97
	RT	6704	281	28429	225131	0.19	0.96	0.89
	Nb	6913	72	27818	221254	0.20	0.99	0.89
	BN	6904	81	26422	222682	0.21	0.99	0.89
<i>D.ere</i>	J48	12503	456	12890	225864	0.49	0.96	0.95
	RF	12137	822	8881	233032	0.58	0.94	0.96
	RT	12001	958	22876	220678	0.34	0.93	0.91
	NB	11828	1131	9515	227616	0.55	0.91	0.96
	BN	11688	1271	10924	225591	0.52	0.90	0.95

For each experiment, we consider a pair of *Drosophila* genomes, in which we use the first genome as training set, and then test the model on the second genome. The abbreviations and sampling follow Table 4.

combined evidence approach to validating draft assemblies of large eukaryotic genomes. We showed that the machine-learning-based methods consistently outperformed the individual measures on both simulated data and draft assemblies from real sequencing projects. Although the requirement of training data appears to be a limitation for our approach, in practice we argue there are two sources that can provide sufficient training data: experimentally validated mis-assembly and cross-species learning. Many genome projects initiate cDNA sequencing, BAC fingerprinting and genetic mapping projects, in addition to genome sequencing. Recently developed high-throughput technologies, such as the genome-scale DNA tiling array (Samanta *et al.*, 2007) are also able to experimentally validate a significant fraction of a draft assembly, which can then be used as training data in our machine-learning approach. Furthermore, the performance of our approach remains satisfactory (as shown in Table 6) when the model is trained on one genome assembly and tested on another assembly from

a closely related genome. We note that the performance improvement of the ML methods in cross-species evaluations are not uniformly advantageous. One classifier may achieve better results when trained using data from one genome over another. Nevertheless, among five machine-learning algorithms we tested, the decision tree and random forest algorithms in general showed better performance than the other non-linear learning algorithms, and thus can be practically applied.

We emphasize that among the many automatic methods we tested, the best precision of mis-assembly detection is only around ~60%, which means that finishing efforts are still un-avoidable to verify these predicted mis-assemblies. However, since false-negative rates are very low and the majority of the assembly is not found to be ‘suspicious’, finishing efforts can be focused on the regions that are flagged as potential errors. Wrongly predicted mis-assemblies indeed reflect the complication for sequencing validations. They are predicted as mis-assemblies because they fall into the ‘difficult-to-assemble’ regions, but the assemblers may still assemble them correctly. Therefore, the true performance of our method may be better than it appears in the evaluation.

Our combined evidence approach described here significantly improves the existing methods based on individual measures, and this is a useful tool for verifying confidence in genome assembly. Currently, we are working on applying our approach to the draft assemblies of *Daphnia* and *Drosophila* genomes. We will report our findings in the future publications.

## ACKNOWLEDGEMENTS

We are grateful to anonymous reviewers for their valuable comments. This research was supported in part by the Indiana METACyt Initiative of Indiana University, funded in part through a major grant from the Lilly Endowment, Inc. and by NSF Career DBI-0237901. Computer support was provided by an allocation TG-MCB060059N through the TeraGrid Advanced Support, by the University Information Technology Services (UITS) and by The Center for Genomics and Bioinformatics computing group. We thank Richard Repasky (UITS) who helped conceive this project.

*Conflict of Interest:* none declared.

## REFERENCES

- Bartels,D. *et al.* (2005) BACCardI – a tool for the validation of genomic assemblies, assisting genome finishing and intergenome comparison. *Bioinformatics*, **21**, 853–859.
- Batzoglou,S. *et al.* (2002) ARACHNE: A whole-genome shotgun assembler. *Genome Res.*, **12**, 177–189.
- Bonfield,J.K. *et al.* (1995) A new DNA sequence assembly program. *Nucl. Acids Res.*, **23** (24), 4992–4999.
- Breiman,L. (2001) Random forests. *Machine Learning*, **45**, 5–32.
- Dew,I.M. *et al.* (2005) A tool for analyzing mate pairs in assemblies (TAMPA). *J. Comput. Biol.*, **12**, 497–513.
- Dietterich,T. (2000) An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, **40**, 139–157.
- Duda,R.O. and Hart,P.E. (1973) Bayes decision theory. In Richard,O.D., Peter,E.H. and David,G.S. (eds.) *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc, pp. 10–43.
- Gilbert,D.G. (2007) DroSpeGe: rapid access database for new *drosophila* species genomes. *Nucl. Acids Res.*, **35** (Suppl 1), D480–485.
- Green,P. (1997) Against a whole-genome shotgun. *Genome Res.*, **7** (5), 410–417.
- Heckerman,D. *et al.* (1995) Learning bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, **20**, 197–243.
- International Chicken Genome Sequencing Consortium. (2004) Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature*, **432**, 695–716.
- Kececioglu,J.D. and Myers,E.W. (1995) Combinatorial algorithms for DNA sequence assembly. *Algorithmica*, **13**, 7–51.
- Kim,S. *et al.* (2001) A probabilistic approach to sequence assembly validation. In *Proceedings of the ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD'01)*. ACM, San Francisco, CA, pp. 38–43.
- Kim,S. *et al.* (2007) *Genome Sequencing Technology and Algorithms*. ArtechHouse, ArtechHouse.
- Lindblad-Toh, *et al.* (2005) Genome sequence, comparative analysis and haplotype structure of the domestic dog. *Nature*, **438**, 803–819.
- Mikkelsen, *et al.* (2007) Genome of the marsupial monodelphis domestica reveals innovation in non-coding sequences. *Nature*, **447**, 167–177.
- Mouse Genome Sequencing Consortium (2002) Initial sequencing and comparative analysis of the mouse genome. *Nature*, **420**, 520–562.
- Myers,E.W. (1995) Towards simplifying and accurately formulating fragment assembly. *J. Comp. Biol.*, **2**, 275–290.
- Nelson,W.M. *et al.* (2005) Whole-genome validation of high-information-content fingerprinting. *Plant Physiol.*, **139** (1), 27–38.
- Pop,M. *et al.* (2002) Genome sequence assembly: Algorithms and issues. *IEEE Computer*, **35**, 47–54.
- Quinlan,J.R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.
- Salzberg,S.L. and Yorke,J.A. (2005) Beware of mis-assembled genomes. *Bioinformatics*, **21** (24), 4320–4321.
- Samanta,M.P. *et al.* (2007) In-depth query of large genomes using tiling arrays. *Methods Mol Biol.*, **377**, 163–174.
- Sanger,F. *et al.* (1982) Nucleotide sequence of bacteriophage [lambda] DNA. *J. Mol. Biol.*, **162**, 729–773.
- Schatz,M. *et al.* (2007) Hawkeye: an interactive visual analytics tool for genome assemblies. *Genome Biology*, **8**, R34.
- Schmutz,J. *et al.* (2003) Assessing the quality of finished genomic sequence. *Cold Spring Harb Symp. Quant. Biol.*, **68**, 31–37.
- Schmutz,J. *et al.* (2004) Quality assessment of the human genome sequence. *Nature*, **429**, 365–368.
- Tang,H. (2007) Genome assembly, rearrangement and repeats. *Chem. Rev.*, **107**, 3391–3406.
- Venter,J.C. *et al.* (2001) The sequence of the human genome. *Science*, **291**, 1304–1351.
- West,J. *et al.* (2006) Validation of *S. pombe* sequence assembly by microarray hybridization. *J. Comput. Biol.*, **13**, 1–20.
- Witten,I.H. and Eibe,F. (2001) *Data Mining*. Hanser Fachbuch, Hanser Fachbuch.
- Zimin,A.V. *et al.* (2005) *Assembly reconciliation method*, unpublished. Available at <http://www.genome.umd.edu/reconciliation.htm>