Optimal reference sequence selection for genome assembly using minimum description length principle

Bilal Wajid, Erchin Serpedin, Mohamed Nounou and Hazem Nounou

Abstract
Reference assisted assembly requires the use of a reference sequence, as a model, to assist in the assembly of the novel genome. The standard method for identifying the best reference sequence for the assembly of a novel genome aims at counting the number of reads that align to the reference sequence, and then choosing the reference sequence which has the highest number of reads aligning to it. This article explores the use of minimum description length (MDL) principle and its two variants, the two-part MDL and Sophisticated MDL, in identifying the optimal reference sequence for genome assembly. The article compares the MDL based proposed scheme with the standard method coming to the conclusion that "counting the number of reads of the novel genome present in the reference sequence" is not a sufficient condition. Therefore, the proposed MDL scheme includes within itself the standard method of "counting the number of reads that align to the reference sequence" and also moves forward towards looking at the model, the reference sequence, as well, in identifying the optimal reference sequence. The proposed MDL based scheme not only becomes the sufficient criterion for identifying the optimal reference sequence for genome assembly but also improves the reference sequence so that it becomes more suitable for the assembly of the novel genome.

1 Introduction
Rissanen's minimum description length (MDL) is an inference tool that learns regular features in the data by data compression. MDL uses "code-length" as a measure to identify the best model amongst a set of models. The model which compresses the data the most and presents the smallest code-length is considered the best model. MDL principle stems from Occam's razor principle which states that "entities should not be multiplied beyond necessity", http://www.cs.helsinki.fi/group/cosco/Teaching/Information/2009/lectures/lecture5a.pdf, stated otherwise, the simplest explanation is the best one, [1-5]. Therefore, MDL principle tries to find the simplest explanation (model) to the phenomenon (data).

The MDL principle has been used successfully in inferring the structure of gene regulatory networks [6-13], compression of DNA sequences [14-18], gene clustering [19-21], analysis of genes related to breast cancer [22-25] and transcription factor binding sites [26].

The article is organized as follows. Section 2 discusses briefly, the variants of MDL and their application to the comparative assembly. Section 3 explains the algorithm used for the purpose. Section 4 elaborates on the simulations carried out to test the proposed scheme. Section 5 explains the results and finally Section 6 points out the main features of this article.

2 Methods
The relevance of MDL to Genome assembly can be realized by understanding that Genome assembly is an inference problem where the task at hand is to infer the novel genome from read data obtained from sequencing. Genome assembly is broadly divided into comparative assembly and de-novo assembly. In comparative assembly, all reads are aligned with a closely related reference sequence. The alignment process may allow one or more mismatches between each individual read and the reference sequence depending on the user. The alignment
of all the reads creates a “Layout”, beyond which the reference sequence is not used any more. The layout helps in producing a consensus sequence, where each base in the sequence is identified by simple majority amongst the bases at that position or via some probabilistic approach. Therefore, this “Alignment-Layout-Consensus” paradigm is used by genome assemblers to infer the novel genome, [27-35].

Comparative assembly, therefore, is an inference problem which requires to identify a model that best describes the data. It begins the process by identifying a model, the “reference sequences”, most closely related to the set of reads. It then uses the set of reads to build on this model producing a model which overfits the data, the “novel genome”, [27,28,34,36-41]. The task of MDL is to identify the model that best describes the data and within comparative assembly framework the same meaning applies to finding the reference sequences that best describes the set of reads.

MDL presents three variants Two-Part MDL, Sophisticated MDL and MiniMax Regret [1]. The application of these will be briefly discussed in what follows.

2.1 Two-part MDL

Also called old-style MDL, the two-part MDL chooses the hypothesis which minimizes the sum of two components:

A) The code-length of the hypothesis.
B) Code-length of the data given the hypothesis.

The two-part MDL selects the hypothesis which minimizes the sum of the code-length of the hypothesis and code-length of the data given the hypothesis, [1,42-47]. The two-part MDL fits perfectly to the comparative assembly problem. The potential hypothesis which is closely related to the data, in comparative assembly, happens to be the reference sequence whereas the data itself happens to be the read data obtained from the sequencing schemes.

2.2 Sophisticated MDL

The two components of the two-part MDL can be further divided into three components:

A) Encoding the model class: \( l(M_i) \), where \( M_i \) belongs to model class, and \( l(M_i) \) denotes the length of the model class in bits.
B) Encoding the parameters \((\theta)\) for any model \( M_i : l_i(\theta) \).
C) Code-length of the data given the hypothesis is \( \log_2 \frac{1}{p_{\theta}(\mathcal{X})} \).

where \( p_{\theta}(\mathcal{X}) \) denotes the distribution of the Data \( \mathcal{X} \) according to the model \( \theta \). The three part code-length assessment process again can be converted into a two-part code-length assessment by combining steps B and C into a single step B.

A) Encoding the model class: \( l(M_i) \), where \( M_i \) belongs to any Model class.
B) Code-length of the Data given the hypothesis class \( (M_i) = l_i(\mathcal{X},\mathcal{Y}) \), where \( \mathcal{X} \) stands for any data set.

Item (B) above, i.e., the ‘length of the encoded data given the hypothesis’ is also called the “stochastic complexity” of the model. Furthermore, if the data is fixed, or if item (B) is constant, then the job reduces to minimizing \( l(M_i) \), otherwise, reducing part (A), [1,48-53].

2.3 MiniMax regret

MiniMax Regret relies on the minimization of the worst case regret, [49,50,53-59]:

\[
\min_{M} \max_{\mathcal{X}} \left[ \text{loss}(M, \mathcal{X}) - \min_{\hat{M}} \text{loss}(\hat{M}, \mathcal{X}) \right],
\]

where \( M \) can be any model, \( \hat{M} \) represents the best model in the class of all models and \( \mathcal{X} \) denotes the data. The Regret, \( R_{M,\mathcal{X}} \), is defined as

\[
R_{M,\mathcal{X}} = \left[ \text{loss}(M_i, \mathcal{X}) - \min_{\hat{M}} \text{loss}(\hat{M}, \mathcal{X}) \right]
\]

Here the loss function, loss\((M_i, \mathcal{X})\), could be defined as the code-length of the data \( \mathcal{X} \) given the model class \( M_i \).

The application of Sophisticated MDL in the framework of comparative assembly will be discussed in what follows.

2.4 Sophisticated MDL and genome assembly

In reference assisted assembly, also known as comparative assembly, a reference sequence is used to assemble a novel genome from a set of reads. Therefore, the best model is the reference sequence most closely related to the novel genome and the data at hand are the set of reads.

However, it should be pointed out that the aim is not to find a general model, rather, the aim is to find a “model that best overfits the data” since there is just one or maybe two instances of the data, based on how many runs of the experiment took place. One “run” is a technical term specifying that the genome was sequenced once and the data was obtained. The term “model that best overfits the data” can be explained using the following example.

Assume one has three Reads \( \{X, Y, \text{and } Z\} \) each having \( n \) number of bases. Say reference sequences \( \{L\} \) and \( \{M\} \), where \( \{L\} = \text{XXYYZZ} \) and \( \{M\} = \text{XYZ} \) contains all three reads placed side by side. Since both models contain all the three reads, the stochastic complexity of both \( \{L\} \) and \( \{M\} \) is the same and both overfit the data perfectly. However, since \( \{M\} \) is shorter than \( \{L\} \), therefore \( \{M\} \) is the model of choice on account of being the model that “best” overfits the data.
Table 1 Counting number of reads not enough

| S.No. | Reference sequence                    | Number of bases in genomes | Number of reads found |
|-------|---------------------------------------|----------------------------|----------------------|
| 1     | Fibrobacter succinogenes subsp. succinogenes S85 (NC_013410.1) | 3842635                   | 157                  |
| 2     | Human Chromosome 21 (AC_000044.1)      | 32992206                   | 158                  |

The table shows that choosing the reference sequence which has the highest number of reads present is not a sufficient condition. Just by looking at the “Data given the model” ≡ “Number of reads found” one ends up choosing Human Chromosome 21. However, looking at the fact that Chromosome 21 is about $9 \times$ larger than S85 one realizes that actually S85 is the model of choice. Furthermore, S85 is a bacterial genome whereas Chromosome 21 comes from a eukaryote genome. PAb1 is also a bacteria, therefore, S85 is most definitely the model of choice.

To formalize the MDL process, the first step would be to identify the following considerations:

A) Encoding the model class: $l(M_i)$, $M_i$ belongs to Model classes.
B) Encoding the parameters ($\theta$) of the Model $M_i$ : $l(\theta)$.
C) Code-length of the data given the hypothesis is $\log_2 \frac{1}{P(D)}$.

The model class in comparative assembly would be the reference (Ref.) sequence itself. The parameters of the model $\theta$, are such that, $\theta \in \{-1, 0, 1\}$. In the process of encoding the model class regions of the genome that are covered by the reads of the unassembled genome are flagged with “1”(s). Areas of the Ref. genome not covered by the reads are flagged as “0”(s), whereas areas of the Ref. genome that are inverted in the novel genome are marked with “−1”(s). In the end, every base of the Ref. sequence is flagged with $\{-1, 0, 1\}$. Therefore, the code-length of the parameters of the model is proportional to length of the sequence.

Data given the hypothesis is typically defined as “Number of reads that align to the Ref. sequence”. In the case presented below “data given the hypothesis” is defined in an inverted fashion as the “Number of reads that do not align to the reference sequence”. These two are interchangeable as the “Total number of reads” is the sum total of the “number of reads that aligned to the Ref.” and the “number of reads that do not align to the Ref.”.

Table 1 shows that choosing the reference sequence having the highest number of reads present is not a sufficient condition for selecting the optimal reference sequence. The simulation carried out compared two reference sequences Fibrobacter succinogenes S85 (NC_013410.1), [60,61], and Human Chromosome 21 (AC_000044.1), [62-64], with the reads of Pseudomonas aeruginosa PAb1 (SRX000424), [48,65,66]. It shows that in order to choose the optimal reference sequence one has to take into account both the “Code-length of the model” and “Number of reads found” to be the sufficient conditions for choosing the optimal reference sequence.

Therefore, a simple yet novel scheme is proposed for the solution to the problem, see Figure 1 and Table 2. The proposed scheme follows the three assessment process of Sophisticated MDL. The MDL based proposed scheme stores the model class (Ref. sequence), the parameters of the model (where each base of the sequence is flagged with $\{-1, 0, 1\}$) and the data given the hypothesis (reads of the novel genome that do not align to the Ref. sequence) is one file. The file is then encoded using either Huffman Coding [67-70] or Shannon–Fano coding [68-71] to determine the code-length. For a simplistic three bits per character coding the code-length is measured according to Equation (3).

![Figure 1 MDL proposed scheme: The output of the system shows that the three components of the encoding scheme are separated from one another by “>”](image)
| S.No. | Ref. Seq. | Model given by the Data | Reads that do not align to the reference sequence | Data given the hypothesis (Bits) | Regret | Proposed scheme | Code-length (Bits) |
|-------|-----------|-------------------------|-------------------------------------------|---------------------------------|--------|----------------|-------------------|
| 1     | ATAT      | 0110111101-1-1-1-1      | CCAA                                      | 12                              | 0      | ATACGCGGCTATA >11110 1111 0-1-1-1-1 >CCAA | 102               |
| 2     | ATGGGCCCTTATTGC | 0000000000000000 | ATAT>GGGG>CCAA | 42   | 30 | ATGGGCCCTTATTGC > 0000000000000000 >ATAT>GGGG >CCAA | 138               |
| 3     | GGGGCCCCGGGG | 1111-1-1-1-11111      | ATAT>CCAA                                 | 27                              | 15     | GGGGCCCCGGGG >1111-1-1-1111 >ATAT>CCAA | 105               |

Regret is defined as $R_{M, \mathcal{X}} = [\text{loss}(M, \mathcal{X}) - \min_{\hat{M}} \text{loss}(\hat{M}, \mathcal{X})]$. Here the loss function, $\text{loss}(M, \mathcal{X})$, happens to be code-length of the data $\mathcal{X}$, given the model class $M$. Whereas, "Data given the hypothesis", is the code-length of the "Reads that do not align to the reference sequence". The code-length in the last column is measured according to Equation (3). The experiment shows that given the MDL, proposed scheme Ref. 1 is the optimal choice for a reference sequence.
The proposed scheme not only allows to determine the best model, amongst the pool of models to choose from, but also improves the model to be better suited according to the novel genome to be assembled. This is done by identifying all insertions and inversions, larger than one read length. It then removes those insertions and rectifies those inversions to get a better model, better suited to assemble the novel genome compared to what was started from, see Figures 2 and 3.

\[
\text{Code length} = (\text{Length}_{\text{Ref. Seq.}} \times 3) \\
+ (\text{Length}_{\text{Parameters of the Model}} \times 3) \\
+ (\text{Length}_{\text{Read}} \times 3 \times \text{No. of Unique Unaligned Reads}).
\] (3)

3 MDL algorithm

The pseudo code for analysis using sophisticated MDL and the scheme proposed in Section 2.4 is shown in Algorithm 1.

Given the reference sequence \( S_R \) and \( K \) set of reads, \( \{r_1, r_2, \ldots, r_K\} \in R \), obtained from the FASTQ [72,73] file, the first step in the inference process is to filter all low quality reads. Lines 3–10 filters all the reads that contain the base \( N \) in them and also the reads which are of low quality leaving behind a set of \( O \) reads to be used for further analysis. This pre-processing step is common to all assemblers. Once all the low quality reads are filtered out, the remaining set of \( O \) reads are sorted and then collapsed so that only unique reads remain.

Lines 13–27 describe the implementation of the proposed scheme as defined in Section 2.4. Assume that \( S_R \) is \( l \) bases long, and the length of each read is \( p \). Therefore, \( \psi_{S_R} \) picks up \( p \) bases at a time from \( S_R \) and checks whether or not \( \psi_{S_R} \) is present in the set of collapsed reads \( R' \). In the event \( \psi_{S_R} \in R' \) then the corresponding location on \( S_R \), i.e., \( j \rightarrow j + p \) are flagged with “1(s)”. If \( \psi_{S_R} \not\in R' \), then invert

![Figure 2 Correcting inversions in the reference sequence.](image)

(A) Reads are derived from the novel sequence. (b) The reference sequence, \( S_R \), contains two inversions, shown as yellow and blue regions. (c) The sequence generated \( \Theta \) has both yellow and blue regions rectified. Notice that using a simple ad-hoc scheme of counting the number of reads in the reference sequence one would have made use of (b) for assembly of novel genome. However, using MDL one can now use (c) for the assembly of the novel genome.

Algorithm 1 MDL Analysis of a Ref. sequence given a set of reads of the unassembled genome.

1. Input reference sequence \( S_R \);
2. Input read data set \( \{r_1, r_2, \ldots, r_K\} \in R \);
3. for \( i : 1 \rightarrow K \) do
   4. if \( r_i \) contains base \( N \) then
      5. remove \( r_i \) from the set of reads;
   6. end if
   7. if \( r_i \) has low quality bases then
      8. remove \( r_i \) from the set of reads;
   9. end if
10. end for
11. Sort remaining set of reads \( \{r_1, r_2, \ldots, r_O\} \in R' \)
12. Collapse duplicated reads.
13. for \( j : 1 \rightarrow l \) do
   14. read \( \psi_{S_R} = (s_{R}^{+p}, s_{R}^{+p+1}, \ldots, s_{R}^{l}) \);
   15. if \( \psi_{S_R} = r_k \in R' \) then
      16. flag \( 1(s) \) in locations \( j \rightarrow j + p \)
      17. flag read \( r_k \) to be present.
   18. else
      19. invert read \( \psi_{S_R} \rightarrow \psi_{S_R} \)
      20. if \( \psi_{S_R} = r_k \in R' \) then
         21. flag \( -1(s) \) in locations \( j \rightarrow j + p \)
         22. flag read \( r_k \) to be present
      23. else
         24. flag \( 0(s) \) in locations \( j \rightarrow j + p \)
   25. end if
   26. end if
27. end for
28. for \( j : 1 \rightarrow l \) do
29. modified sequence \( \Theta \leftarrow S_R \)
30. identify all inversions by looking at \(-1\) flags
31. start = start of an inversion
32. end = end of an inversion
33. invert genome \( \Theta_{\text{start}} \rightarrow \Theta_{\text{end}} \)
34. end for
35. for \( j : 1 \rightarrow l \) do
36. identify all insertions by looking at \( 0 \) flags
37. start = start of an insertion
38. end = end of an insertion
39. if \( r_1 < \text{end} - \text{start} < r_2 \) then
40. remove segment of genome \( \Theta_{\text{start}} \rightarrow \Theta_{\text{end}} \)
41. else
42. segment of genome is either too large or too small.
43. end if
44. end for
45. for \( i : 1 \rightarrow O \) do
46. if read \( r_i \) is flagged, remove from \( R \);
47. end for
48. \( \xi = \text{Code-length of encoded modified sequence } \Theta \)
49. \( \gamma = \text{Code-length of reads } R' \text{ not present in } S_R \)
50. Total code-length \( \xi = \xi + \gamma \).
φ_{SR} → ψ_{SR} and check whether or not ψ_{SR} ∈ R′. If yes, then mark the corresponding location on SR, i.e., j → j + p with “−1(s)” and flag φ_{SR} to be present in R′. Otherwise, mark the corresponding locations on SR as “0(s)

Lines 28–34 generates a modified sequence Θ which has all the inversions rectified in the original sequence SR. Lines 35–44 identifies all insertions larger than τ₁ and smaller than τ₂ and removes them, see Figure 3. Here τ₁ and τ₂ are user-defined. Care should be taken to avoid removing very large insertions as this may affect the overall performance in deciding the best sequence for genome assembly. Lines 45–47 removes all the reads that are present in the original SR and the modified sequence Θ identified by flags 1 and −1. In the end the code-lengths are identified by any popular encoding scheme like Huffman [67-70] or Shannon-Fano coding [68-71]. If ξ is the smallest code-length amongst all models then use Θ as a reference for the assembly of the unassembled genome rather than using SR.

4 Results

Simulations were carried out on both synthetic data as well as real data. At first, the MDL process was analyzed on synthetic data on four different sets of mutations by varying the number and length of (Single nucleotide polymorphisms (SNPs), Inversions, Insertions, and Deletions). The experiments using synthetic data were carried out by generating a sequence SN. The set of reads were derived from SN and sorted using quick sort algorithm [74,75]. Each experiment modified SN to produce two reference sequences SR₁ and SR₂ by randomly putting in the four set of mutations. The choice of the best reference sequence was determined by the code-length generated by the MDL process. See Tables 3, 4, 5, and 6 for results.

Once the robustness of MDL scheme on each of the four types of mutations was confirmed two set of experiments were carried out on real data using Influenza viruses A, B, and C which belong to the Orthomyxoviridae group. Influenza virus A has five different strains, i.e., {H1N1, H5N1, H2N2, H3N2, H9N2}, while Influenza viruses B and C each have just one. The genomes of Influenza viruses is divided into a number of segments. Influenza virus A and B each have eight segments while virus C has seven segments, [76-78]. Amongst the first segments of each of the viruses only one was randomly selected and then modified to be our novel genome, SN. Reads were then derived from SN and compared

Table 3 Variable number of SNPs: the experiment shows the effect of increasing the number of SNPs on choice of the reference sequence

| Ref. Seq. | SNPs | No. of inversions | No. of insertions | No. of deletions | Code-length using proposed scheme (Kb) |
|----------|------|------------------|------------------|-----------------|--------------------------------------|
| 1        | 183  | 52 / 52          | 62 / 59          | 62              | 1815.14                              |
| 2        | 224  | 50 / 51          | 66 / 58          | 63              | 1843.35                              |

SN₁ has higher number of SNPs as opposed to SN₂. The code-length suggests that SN₁ is the model of choice as it has a smaller code-length. The results show that the MDL scheme works successfully on variable number of SNPs by choosing the model with a lower number of SNPs in them.

Table 4 Variable number of insertions: the experiment shows the effect of increasing the number of insertions on choice of the reference sequence

| Ref. Seq. | SNPs | No. of inversions | No. of insertions | No. of deletions | Code-length using proposed scheme (Kb) |
|----------|------|------------------|------------------|-----------------|--------------------------------------|
| 1        | 0    | 0                | 136 / 196        | 0               | 1200.3                               |
| 2        | 0    | 0                | 132 / 203        | 0               | 1228.25                              |

The location and length of these insertions was chosen randomly. ! shows that out of 196 insertions in SN₁ only 136 were removed. The remaining insertions were not recovered due to the choice of τ₁ and τ₂. SN₂ has higher number of insertions as opposed to SN₁. The code-length suggests that SN₁ is the model of choice as it has a smaller code-length.

Figure 3 Removing insertions in the reference sequence. (a) Reads are derived from the novel sequence. (b) The reference sequence, SR, contains two insertions, shown as shaded grey boxes. (c) The proposed MDL process generates Θ. The process removes only those insertions which are larger than τ₁ but smaller than τ₂; where τ₁ and τ₂ are user-defined. To remove the other insertion the value of τ₂ could be increased.
Table 5 Variable number of deletions: the experiment shows the effect of increasing the number of deletions on choice of the reference sequence

| Ref. Seq. | SNPs | No. of inversions | No. of insertions | No. of deletions | Code-length using proposed scheme (Kb) |
|-----------|------|-------------------|-------------------|------------------|----------------------------------------|
| 1         | 0    | 2 / 0             | 182               | 1997.28          |
| 2         | 0    | 3 / 0             | 189               | 2015.35          |

The location and length of these deletions was chosen randomly. SR₂ has higher number of deletions as opposed to SR₁. The code-length suggests that SR₁ is the model of choice as it has a smaller code-length. The experiment show that although no insertions were put in the actual sequence yet still two and three insertions were found for SR₁ and SR₂, respectively. This may be due to a large section of reads that could not align to the reference sequence on the edges of these deletions.

Table 6 Variable number of inversions: the experiment shows the proposed scheme is robust to the number of inversions in the reference sequence

| Ref. Seq. | SNPs | No. of inversions | No. of insertions | No. of deletions | Code-length using proposed scheme (Kb) |
|-----------|------|-------------------|-------------------|------------------|----------------------------------------|
| 1         | 0    | 0                 | 0                 | 0                | 586.04                                |
| 2         | 0    | 176 / 176         | 0                 | 0                | 586.04                                |

Both SR₁ and SR₂ have the same code-length. This is because the MDL scheme not only detected all the inversions for SR₂ but also recovered all of them. So effectively SR₂ ≡ SR₁ after the MDL process as explained in Figure 2.

Table 7 Simulations with Influenza virus A, B, and C

| S.No. | Ref. Seq. (Influenza virus) | No. of inversions | No. of deletions | Code-length using proposed scheme (Kb) |
|-------|----------------------------|-------------------|------------------|----------------------------------------|
| 1     | A, H1N1 (NC_002023.1)      | 0 / 4             | 1                | 254.109                                |
| 2     | A, H5N1 (NC_007357.1)      | 0 / 4             | 1                | 254.109                                |
| 3     | A, H2N2 (NC_007378.1)      | 0 / 4             | 1                | 254.109                                |
| 4     | A, H3N2 (NC_007373.1)      | 0 / 4             | 1                | 254.109                                |
| 5     | A, H9N2 (NC_004910.1)      | 0 / 4             | 1                | 254.109                                |
| 6     | B (NC_002204.1)            | 4 / 4             | 1                | 68.62                                  |
| 7     | C (NC_006307.1)            | 0 / 4             | 1                | 254.027                                |

One of the sequences from Influenza virus (A, B, C) was randomly selected and modified to include [SNPs = 7, inversions = 4, deletions = 1, insertions = 3]. As Influenza virus A has five different strains while both Influenza viruses B and C each have one the MDL process was used to compare the seven sequences to determine which is the best reference sequence. Ref. Seq 6, Influenza virus B was found to have the smallest code-length (68.62 Kb), and is therefore, the model of choice. The experiment also shows that given the optimal reference sequence, in this case Influenza virus B, the MDL process rectifies all inversions (4/4). However, given non-optimal reference sequences, the proposed MDL process is not able to rectify the inversions (0/4). So the proposed algorithm chooses the optimal reference sequence, and given the optimal reference sequence if not all, at least most of the inversions are also corrected.
Table 8 The experiment uses the proposed MDL scheme on the same set of reads but different set of reference sequences

| S.No. | Ref. Seq. (%) | No. of unaligned reads | Code-length (KB) | Execution time (s) | Length of new Seq. |
|-------|---------------|------------------------|------------------|--------------------|-------------------|
| 1     | 1             | 696                    | 128.60           | 0.046              | 14                |
| 2     | 2             | 696                    | 128.73           | 0.031              | 47                |
| 3     | 5             | 693                    | 128.57           | 0.046              | 113               |
| 4     | 10            | 684                    | 127.576          | 0.046              | 229               |
| 5     | 25            | 668                    | 126.615          | 0.093              | 565               |
| 6     | 50            | 650                    | 126.615          | 0.109              | 650               |
| 7     | 100           | 3                      | 14.276           | 0.078              | 2342              |
| 8     | 150           | 2                      | 21.164           | 0.062              | 2341              |
| 9     | 200           | 2                      | 27.808           | 0.124              | 2341              |
| 10    | 300           | 2                      | 41.525           | 0.140              | 2341              |

The set of reads contained 3817 reads all of which were derived from ‘Influenza A virus (A Puerto Rico 834 (H1N1)) segment 1, complete sequence’. Out of 3817 reads the method extracted 696 unique reads which were then used in the MDL proposed scheme. All the reference sequences were derived from the same Influenza A (H1N1) virus. Ref. Seq. 1% used in S.No. 1, has a length which is 1% of the actual genome. Similarly Ref. Seq. 25% has a length which is a quarter of the length of the actual genome. All other genomes were derived in a similar way. For, e.g., Ref. Seq. 200% has two H1N1 viruses concatenated together making the length twice that of the original H1N1 sequence. The code-length is calculated using Equation (3). The results show that the MDL proposed scheme chooses the best reference sequence, one which has the smallest code-length as determined by Equation (3). The MDL scheme does not choose smaller reference sequences with more unaligned reads rather than choosing larger reference sequence with smaller unaligned reads. The experiment also proves the correctness of the optimal reference sequence as it chooses Ref. Seq. 7 (shown underlined), since it has the smallest code-length, as the optimal reference sequence. It was Ref. Seq. 7 from which all the reads were derived from. Since the MDL scheme chooses Ref. Seq. 7 as the optimal sequence, the experiment also proves the correctness of the reference sequence chosen.

The experiment carried out using Influenza viruses is shown in Table 7. One sequence was randomly chosen amongst the seven sequences and modified at random locations, using the same ‘change_sequence’ program, to form the novel sequence \( S_N \). The novel sequence contained \{SNPs = 7, inversions = 4, deletions = 1, insertions = 3 \} as compared to the original sequence. The MDL process used the reads derived from \( S_N \) to compare seven sequences and determined Influenza virus B to be optimal reference sequence as it had the smallest code-length. The MDL process rectified all inversions while only one insertion was found. This meant that the remaining two SNPs, see Table 3, smaller number of insertions, see Table 4, and smaller number of deletions compared to the novel genome, see Table 5. The proposed MDL scheme is also seen to detect and rectify most, if not all, of the inversions present in the reference sequence, see Table 6. Since the code-length of \( S_{R1} \) is the same as \( S_{R2} \), and all the inversions of \( S_{R2} \) are rectified, the corrected \( S_{R2} \) sequence and \( S_{R1} \) sequence are equally good for reference assisted assembly.

The experiment carried out using Influenza viruses is shown in Table 7. One sequence was randomly chosen amongst the seven sequences and modified at random locations, using the same ‘change_sequence’ program, to form the novel sequence \( S_N \). The novel sequence contained \{SNPs = 7, inversions = 4, deletions = 1, insertions = 3 \} as compared to the original sequence. The MDL process used the reads derived from \( S_N \) to compare seven sequences and determined Influenza virus B to be optimal reference sequence as it had the smallest code-length. The MDL process rectified all inversions while only one insertion was found. This meant that the remaining two

Table 9 The experiment tests the proposed MDL scheme on a single set of reads yet on a number of reference sequences

| S.No. | Ref. Seq. (%) | No. of unaligned reads | Code-length (KB) | Length of new Seq. |
|-------|---------------|------------------------|------------------|--------------------|
| 1     | 75            | 172                    | 25.91            | 1755               |
| 2     | 85            | 148                    | 25.10            | 1989               |
| 3     | 95            | 123                    | 24.20            | 2223               |
| 4     | 100           | 109                    | 23.62            | 2341               |
| 5     | 105           | 108                    | 24.22            | 2458               |
| 6     | 115           | 107                    | 25.50            | 2692               |
| 7     | 125           | 106                    | 26.78            | 2926               |

The set of reads, 390 in total, were derived from ‘Influenza A virus (A Puerto Rico 834 (H1N1)) segment 1, complete sequence’ using the ART read simulator for NGS with read length 30, standard deviation 10, and mean fragment length of 100.[79] Similarly the reference sequences were also derived from the same H1N1 virus. Ref. Seq. 75% used in S.No. 1, has a length which is 75% of the actual genome. Similarly Ref. Seq. 125% has a quarter of the actual genome concatenated with the complete H1N1 genome making the total length 125% of H1N1. All other genomes were derived in a similar way. The code-length is calculated using Equation (3). The results show that the MDL proposed scheme chooses the correct reference sequence, Ref. Seq. 100%, (shown underlined) even when all the contending sequences are closely related to one another in terms of their genome and length.
insertions were smaller than $\tau_1$. The set of reads and Influenza virus B was then fed into MiB (MDL-IDITAP-Bayesian estimation comparative assembly pipeline) [80]. The MiB pipeline removes insertions and rectifies insertions using the MDL proposed scheme. IDITAP is a de-bruijn graph based denovo assembler that identifies the Deletions and Inserts them as Appropriate Places. BECA (Bayesian Estimator Comparative Assembler) helps in rectifying all the SNPs. The novel genome reconstructed by the MiB pipeline was one contiguous sequence with a length of 2368 bases and a completeness of 96.62%.

The second-set of experiment tests the correctness of the MDL proposed scheme, by testing the MDL scheme on a single set of reads but on a number of different reference sequences having a wide range of lengths. In the first test 3817 reads were derived from ‘Influenza A virus (H1N1) segment 1’ without any mutations, of which only 696 reads remained after collapsing duplicate reads. The reference sequences were also derived from the same H1N1 virus, with reference sequence (Ref. Seq.) 1% having a length which is 1% of the actual genome. Similarly Ref. Seq. 25% has a length which is a quarter of the length of the actual genome. Similarly Ref. Seq. 125% has a quarter of the actual genome concatenated with the complete H1N1 genome making the total length 125% of H1N1. All other reference sequences were derived in a similar way, see Table 8. The unique set of reads and the reference sequences were tested using the MDL proposed scheme, where the code-length was calculated using Equation (3). The results show that the MDL scheme does not choose smaller reference sequences with more unaligned reads rather it chooses the correct reference sequence, Ref. Seq. 7. It was Ref. Seq. 7 from which all the reads were derived from. Since the MDL scheme chooses Ref. Seq. 7 as the optimal sequence, this experiment further proves the correctness of the reference sequence chosen.

Lastly, the above experiment was repeated using a single set of reads derived from the same H1N1 virus segment 1, but this time containing mutations. The set of reads, 390 in total, were derived using the ART read simulator for NGS with read length 30, standard deviation 10, and mean fragment length of 100, [PUT ART Reference], see Table 9. The results show that the MDL proposed scheme chooses the correct reference sequence, Ref. Seq. 100%, even when all the contending reference sequences are closely related to one another in terms of their genome and length.

All simulations were carried out on Intel Core i5 CPU M430 @ 2.27 GHz, 4 GB RAM. Execution time of MDL proposed scheme have been provided in Table 8.

6 Conclusions
The article explored the application of Two-Part MDL qualitatively and the application of Sophisticated MDL both qualitatively and quantitatively for selection of the optimal reference sequence for comparatively assembly. The article compared the MDL scheme with the standard method of “counting the number of reads that align to the reference sequence” and found that the standard method is not sufficient for finding the optimal sequence. Therefore, the proposed MDL scheme encompassed within itself the standard method of ‘counting the number of reads’ by defining it in an inverted fashion as ‘counting the number of reads that did not align to the reference sequence’ and identified it as the ‘data given the hypothesis’. Furthermore, the proposed scheme included the model, i.e., the reference sequence, and identified the parameters ($\theta_M$) for the model ($M_i$) by flagging each base of the reference sequence with $\tau_1, 0, 1$. The parameters of the model helped in identifying inversions and thereafter rectifying them. It also identified locations of insertions. Insertions larger than a user defined threshold $\tau_1$ and smaller than $\tau_2$ were removed. Therefore, the proposed MDL scheme not only chooses the optimal reference sequence but also fine-tunes the chosen sequence for a better assembly of the novel genome.

Experiments conducted to test the robustness and correctness of the MDL proposed scheme, both on real and simulated data proved to be successful.

Competing Interests
The authors declare that they have no competing interests.

Acknowledgements
This article has been partly funded by the University of Engineering and Technology, Lahore, Pakistan (No. Estab/DBS/411, Dated Feb 16, 2008), National Science Foundation grant 0915444 and Qatar National Research Fund—National Priorities Research Program grant 09-874-3-235. The first author would like to extend special thanks to his family. The authors acknowledge the Texas A&M Supercomputing Facility (http://sc.tamu.edu/) for providing computing resources useful in conducting the research reported in this article.

Author details
1 Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA. 2 Department of Electrical Engineering, University of Engineering & Technology, Lahore, Punjab 54890, Pakistan. 3 Department of Electrical Engineering, Texas A&M University, Doha, Qatar. 4 Department of Electrical and Computer Engineering, Texas A&M University, Doha, Qatar.

Received: 14 January 2012 Accepted: 11 September 2012 Published: 27 November 2012

References
1. T Roos. (Helsinki University Printing House, Helsinki, 2007), pp. 1–82
2. P Domingos, The role of Occam’s razor in knowledge discovery. Data Min Knowledge Discovery. 3(4), 409–425 (1999)
3. M Li, P Vitányi, An Introduction to Kolmogorov Complexity and its Applications. (Springer-Verlag Inc., New York, 2008)
4. C Rasmussen, Z Ghahramani, Occam’s razor. Adv. Neural Inf. Process Syst. 13, 294–300 (2001)
5. V Vapnik, The Nature of Statistical Learning Theory. (Springer-Verlag Inc., New York, 2000)
6. J Dougherty, I Tabus, J Astola, Inference of gene regulatory networks based on a universal minimum description length. EURASIP J. Bioinf. Syst. Biol. 2008, 1–11 (2008)
7. W Zhao, E Serpedin, E Dougherty, Inferring gene regulatory networks from time series data using the minimum description length principle. Bioinformatics. 22(17), 2129 (2006)

8. V Chattanak, P Ghosh, E Perkins, P Gong, Y Deng, C Zhang, A novel gene network inference algorithm using predictive minimum description length approach. BMC Syst. Biol. 4(Suppl 1), S7 (2010)

9. J Androulakis, E Yang, R Almon, Analysis of time-series gene expression data: Methods, challenges, and opportunities. Annual Rev. Biomed. Eng. 9, 205–228 (2007)

10. H Lähdesmäki, I Shmulevich, O Yi-Harja, On learning gene regulatory networks under the Boolean network model. Mach. Learn. 52, 147–167 (2003)

11. V Chattanak, C Zhang, P Ghosh, E Perkins, P Gong, Y Deng, in IEEE International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, 2009. UCB509. Gene regulatory network inference using predictive minimum description length principle and conditional mutual information. (Shanghai, China, 2009), pp. 487–490

12. E Dougherty, Validation of inference procedures for gene regulatory networks. Curr. Genom. 8(6), 351 (2007)

13. X Zhou, X Wang, R Pal, I Ivanov, M Bittner, E Dougherty, A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. Bioinformatics. 20(17), 2918–2927 (2004)

14. G Korodi, I Tabus, An efficient normalized maximum likelihood algorithm for DNA sequence compression. ACM Trans. Inf. Syst. (TOIS). 23, 3–34 (2005)

15. G Korodi, I Tabus, J Rissanen, J Astola, DNA sequence compression-Based on the normalized maximum likelihood model. IEEE Signal Process. Mag. 24, 47–53 (2006)

16. I Tabus, G Korodi, J Rissanen, in IEEE Proceedings on Data Compression Conference, Snowbird. DNA sequence compression using the normalized maximum likelihood model for discrete regression. (Utah, USA, 2003), pp. 253–262

17. S Evans, S Markham, A Torres, A Kourtidis, D Conklin, in IEEE First Asilomar Conference on Signals, Systems and Computers, 2006. ACSSC’06. An improved minimum description length learning algorithm for nucleotide sequence analysis. (Pacific Grove, CA, 2006), pp. 1843–1850

18. A Milosavljevic, J Jurka, Discovery by minimal length encoding: a case study in molecular evolution. Mach. Learn. 12, 69–87 (1993)

19. R Jornsten, B Yu, Simultaneous gene clustering and subset selection for sample classification via MDL. Bioinformatics. 19(9), 1100 (2003)

20. I Tabus, J Astola, in Proceedings of the Seventh International Symposium on Signal Processing and its Applications, ISSPA 2003, vol. 2. Clustering the non-uniformly sampled time series of gene expression data. (Paris, France, 2003), pp. 61–64

21. A Jain, Data clustering: 50 years beyond K-means. Pattern Recogn. Lett. 31(8), 651–660 (2010)

22. S Evans, A Kourtidis, T Markham, J Miller, D Conklin, A Torres, MicroRNA target detection and analysis for genes related to breast cancer using MDLcompress. EURASIP J. Bioinf. Syst. Biol. 2007, 1–16 (2007)

23. E El-Sebakhy, K Faisal, T Helmy, F Azzedin, A Al-Sulfain, in the 4th ACS/IEEE International Joint Conference on Bioinformatics, Systems Biology and Signal Processing. Evaluation of breast cancer tumor classification with unconstrained functional networks classifier. (Los Alamitos, CA, USA, 2006), pp. 281–287

24. A Bulychev, S Semenov, A Sourovov, R Svenson, A Nazarov, Y Sizov, G Tatischeff, Computational modeling of three-dimensional microwave tomography of breast cancer. IEEE Trans. Biomed. Eng. 48(9), 1053–1056 (2001)

25. D Buckel, Minimum description length methods of medium-scale simultaneous inference. (Ottawa Institute of Systems Biology, Techn rep, Ottawa, 2010)

26. J Schug, G O’Connor, in Proc Int Conf Inf Syst Mol Biol vol. S. Modelling transcription factor binding sites with Gibbs sampling and minimum description length encoding. (Halkidiki, Greece, 1997), pp. 268–271

27. B Wajid, E Serpedin, Review of general algorithmic features for genome assemblers for next generation sequencers. Genomics, Proteomics & Bioinformatics. 10(2), 58–73 (2012)

28. B Wajid, E Serpedin, Supplementary information section: review of general algorithmic features for genome assemblers for next generation sequencers. Genomics, Proteomics & Bioinformatics. 10(2), 58–73 (2012). [https://sites.google.com/site/bilalwajid786/research]

29. J Miller, S Koren, G Sutton, Assembly algorithms for next-generation sequencing data. Genomics. 95(6), 315–327 (2010)

30. M Pop, Genome assembly reborn: recent computational challenges. Brief. Bioinf. 10(4), 354–366 (2009)

31. C Alkan, S Sajjadian, E Eichler, Limitations of next-generation genome sequence assembly. Nat. Methods. 8, 61–65 (2010)

32. P Fliscik, E Bimey, Sense from sequence reads: methods for alignment and assembly. Nat. Methods. 6, S6–S12 (2009)

33. E Marsi, Next-generation DNA sequencing methods. Annu. Rev. Genom. Hum. Genet. 9, 387–402 (2008)

34. M Schatz, A Delcher, S Salzberg, Assembly of large genomes using second-generation sequencing. Genome Res. 20(9), 1165 (2010)

35. M Pop, S Salzberg, Bioinformatics challenges of new sequencing technology. Trends Genet. 24(3), 142–149 (2008)

36. M Pop, A Phillippy, A Delcher, S Salzberg, Comparative genome assembly. Brief. Bioinf. 5(3), 237 (2004)

37. S Kurtz, A Phillippy, A Delcher, M Smoot, M Shumway, C Antonescu, S Salzberg, Versatile and open software for comparing large genomes. Genome Biol. 5(2), R12 (2004)

38. M Pop, D Kosack, S Salzberg, Hierarchical scaffolding with Bambus. Genome Res. 14, 149 (2004)

39. S Salzberg, D Sommer, D Puiu, V Lee, Gene-boosted assembly of a novel bacterial genome from very short reads. PLoS Comput. Biol. 4(10), e1000186 (2008)

40. M Schatz, B Langmead, S Salzberg, Cloud computing and the DNA data race. Nat. Biotechnol. 28(7), 691 (2010)

41. S Gnerre, E Lander, K Lindblad-Toh, D Jaffe, Assisted assembly: how to improve a de novo genome assembly by using related species. Genome Biol. 10(8), R88 (2009)

42. J Rissanen, MDL denoising. IEEE Trans. Inf. Theory. 46(7), 2537–2543 (2000)

43. J Rissanen, Hypothesis selection and testing by the MDL principle. Comput. J. 42(4), 260–269 (1999)

44. R Baxter, J Oliver, MDL and MML: Similarities and Differences, vol. 207. (Dept. Comput. Sci. Monash Univ, Clayton, Victoria, Australia, Tech. Rep, 1994)

45. P Grünwald, P Kontkanen, P Myllymäki, T Silander, H Tirir, in Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Minimum encoding approaches for predictive modeling (Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1998), pp. 183–192

46. B Wajid, E Serpedin, in 2011 IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS). Minimum description length based selection of reference sequences for comparative assemblers, (San Antonio, TX, USA, 2011), pp. 230–233

47. T Silander, T Roos, P Kontkanen, P Myllymäki, in 4th European Workshop on Probabilistic Graphical Models, Hirtsalah. Factorized normalized maximum likelihood criterion for learning Bayesian network structures, (Denmark, 2008), pp. 257–264

48. P Grünwald, A tutorial introduction to the minimum description length principle. Arxiv preprint math/0406077 (2004)

49. J Oliver, D Hand, Introduction to Minimum Encoding Inference, (Dept. Comput. Sci., Monash University, Clayton, Vic. 3168, Australia, Tech. Rep, 1994)

50. C Wallace, D Dawe, Minimum message length and Kolmogorov complexity. Comput. J. 42(4), 270–283 (1999)

51. P Grünwald, in Advances in Minimum Description Length: Theory and Applications. Minimum description length tutorial (MIT Press, Cambridge, Cambridge, MA 02142, 2005), pp. 1–80

52. A Barron, J Rissanen, B Yu, The minimum description length principle in coding and modeling. IEEE Trans. Inf. Theory. 44(6), 2745–2760 (1998)

53. Q Xie, A Barron, Asymptotic minimax regret for data compression, gambling, and prediction. IEEE Trans. Inf. Theory. 46(2), 431–445 (2000)

54. S De Rooij, P Grünwald, An empirical study of minimum description length model selection with infinite parametric complexity. J. Math. Psychol. 50(2), 180–192 (2006)

55. T Roos, in IEEE Information Theory Workshop, 2008. ITW'08. Monte Carlo estimation of minimax regret with an application to MDL model selection, (Porto, Portugal, 2008), pp. 284–288
58. Y Yang, Minimax nonparametric classification. II. Model selection for adaptation. IEEE Trans. Inf. Theory. 45(7), 2285–2292 (1999)

59. F Rezaei, C Charalambous, in IEEE Proceedings International Symposium on Information Theory, 2005. ISIT. Robust coding for uncertain sources: a minimax approach, (Adelaide, SA, 2005), pp. 1539–1543

60. G Suen, P Weaver, D Stevenson, F Aylikward, J Boyum, J Deneke, C Drinkwater, N Ivanova, N Mikhailova, O Cherktov, L Goodwin, C Currie, D Mead, P Brumm, The complete genome sequence of Fibrobacter succinogenes S85 reveals a cellulosytic and metabolic specialist. PloS one. 6(4), e18814 (2011)

61. C Luo, D Tsementzi, N Kyprides, T Read, K Konstantinidis, Direct comparisons of Illumina vs. Roche 454 sequencing technologies on the same microbial community DNA sample. PloS one. 7(2), e30087 (2012)

62. M Hattori, A Fujiyama, T Taylor, H Watanabe, T Yada, H Park, A Toyoda, K Ishii, Y Totoki, D Choi, et al, The DNA sequence of human chromosome 21. Nature. 405(6784), 311–319 (2000)

63. R Waterston, E Lander, J Sulston, On the sequencing of the human genome. Proc. Natl. Acad. Sci. 99(6), 3712 (2002)

64. S Istrail, G Sutton, L Florea, A Halpern, C Mоборry, R Lippert, B Walenz, H Shatkay, J Dew, J Miller, et al, Whole-genome shotgun assembly and comparison of human genome assemblies. Proc. Natl. Acad. Sci. US Am. 101(7), 1916 (2004)

65. S Salzberg, D Sommer, D Puiu, V Lee, Gene-boosted assembly of a novel bacterial genome from very short reads. PloS Comput. Biol. 4(9), e1000186 (2008)

66. N Croucher, From small reads do mighty genomes grow. Nature Rev. Microbiol. 7(9), 621–621 (2009)

67. D Huffman, A method for the construction of minimum-redundancy codes. Proc. IRE. 40(9), 1098–1101 (1952)

68. T Cover, J Thomas, J Wiley, et al, Elements of information theory, vol. 6. (Wiley InterScience, New York, 1991)

69. M Rabbani, P Jones, Digital image compression techniques. (SPIE Publications, Bellingham, Washington, vol. TT7, 1991)

70. J Kieffer, Data Compression. (Wiley InterScience, New York, 1971)

71. R Fano, D Hawkins, Transmission of information: a statistical theory of communications. Am. J. Phys. 29, 793 (1961)

72. P Cöck, C Fields, N Goto, M Heuer, P Rice, The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res. 38(6), 1767–1771 (2010)

73. N Rodriguez-Ezpeleta, M Hackenberg, A Aransay, Bioinformatics for High Throughput Sequencing. (Springer Verlag, New York, 2011)

74. C Hoare, Quicksort. Comput. J. 5, 10 (1962)

75. J Kingston, Algorithms and Data Structures: Design, Correctness, Analysis. (Addison-Wesley, Sydney, 1990)

76. K Renegar, Influenza virus infections and immunity: a review of human and animal models. Lab. Animal Sci. 42(3), 222 (1992)

77. K Myers, C Olsen, G Gray, Cases of swine influenza in humans: a review of the literature. Clin. Infect. Diseases. 44(8), 1084 (2007)

78. D Suarez, S Schultz-Cherry, Immunology of avian influenza virus: a review. Develop. Comparat. Immunol. 24(2–3), 269–283 (2000)

79. W Huang, L Li, JY Myers, GT Marth, ART: a next-generation sequencing read simulator. Bioinf. 28(4), 593–594 (2012)

80. B Wajid, E Serpedin, M Nounou, H Nounou, in 2012 IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS’12). MiB: a comparative assembly processing pipeline, (Washington DC., USA, 2012)