Netw Model Anal Health Inform Bioinforma (2016) 5:27
DOI 10.1007/s13721-016-0132-7

ORIGINAL ARTICLE

MetaG: a graph-based metagenomic gene analysis for big DNA data

Linkon Chowdhury[1] · Mohammad Ibrahim Khan[1] · Kaushik Deb[1] · Sarwar Kamal[2]

Received: 1 October 2015 / Revised: 15 June 2016 / Accepted: 16 June 2016 / Published online: 15 July 2016

Abstract Microbial interactions and relationships are significant for animals, insects and plants. Metagenomic research enables properassessments and analysis for microbial organs and communities. The analysis helps to gain detailed insights on miscopies insects. Recent machine learning techniques focused on algorithms and data mining tools to check the depth of interactions and relationships on metagenomic dataset. Accurate analysis over large genes helps to solve real-world problems for public interest. In this regard, graph-centric big gene dataset representations are very important. De Bruijn graph is one the pivotal media to demonstrate the relationships and interactions of large genes dataset or metagenomic dataset. In this research, mapping-based metagenomic graphical (MetaG) genomes representation has been demonstrated. Data cleaning is done before applying graphical illustration. Random mapping is used to assess the variations in dataset. Euler path-based De Bruijn graph is used to sketch the gene annotation, translations, signaling and coding. This research helps in computational biology to map the genomic information in graphical ways with clear conceptions. Adequate experimental comparisons as well as analysis established the claims with tables and graphs.

Keywords Metagenomic · Euler path · Coding regions · De Bruijn graph

1 Introduction

In the age of digitalization, genomic datasets are increasing exponentially in all respects of biological research and productions. Industries, universities, laboratories, agriculture, healthcare and farm houses are producing billions of data every day. From the millennium, metagenomic data analysis has become one of the key areas in computational biology, bioinformatics and genomics. Parallel processing or next-generation sequencing enables massive computational support to solve big datasets and generate new datasets again and again (Freitas et al. 2015; Hultman et al. 2015; Mitchell et al. 2015; Kopf et al. 2015). In this regard, increasing datasets requires efficient techniques to represent metagenomic information and structures. Now-a-days, researchers are developing reference-free machine learning method to assess the metagenomic data structures. Metagenomic analysis depicts a meaningful process that can find a simpler illustration and sequencing for rRNA dataset for large microbial associations (Sunagawa et al. 2015; Villar et al. 2015). Some popular research demonstrates that there are about 100 trillions of cells constructed by microbes in human bodies. The majority locations of microbes are in the guts that have pivotal impact on human characteristics such as physiology and nutrition. Consequently, these gut microbes generate energy from food and alter the gut elements related to some diseases (Hsiao et al.

✉ Linkon Chowdhury
linkoncuet@gmail.com

Mohammad Ibrahim Khan
muhammad_ikhancuet@yahoo.com

Kaushik Deb
debkaushik99@gmail.com

Sarwar Kamal
sarwar.saubdcoxbazar@gmail.com; skamal@ewubd.edu

[1] Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh

[2] Department of Computer Science and Engineering, East West University, Aftab Nagar, Dhaka, Bangladesh

2014; Markowitz et al. 2014; Hunter et al. 2014). To have enough ideas on gut impact on human body as well as animals, it is essential to assess the interactions of metagenomic datasets. rRNA-centric sequencing helps to get the idea regarding bacterial divisions that determines the functionalities of the major parts of the gut of microbes (Huang et al. 2014). More research shows that the gut has tremendous impact on human metagenomic as well as interactions (Forster and Lawley 2015; Silvester et al. 2015; Bolger et al. 2014).

Basic metagenomic research is to computer the pair-wise distinctions between genomes (Lozupone et al. 2011). This approach is simple; however, it works in small datasets. One known and common analysis is beta-variations analysis that numerically measures the dissimilarities between two microbial genome groups. Basic characteristics of metagenomic representations are done by considering important factor such as taxonomic comparisons, total groups of genomic data, phylogenetic framework and geometrical orientations. Mathematical and statistical analysis helps to obtain meaningful information from thousands of genomes. These genome dimensions are very essential for getting faster information from disarray datasets. Dissimilarity matrix arranges all the adjacent distances among the collected datasets in row and column orientations. For big datasets, there are large-scale metagenomic genome sequences that require easy representations and processing. For these reasons high-performance algorithms and techniques are in demand in metagenomic research and analysis (Lu et al. 2015; Li et al. 2015; Jing et al. 2004; De Cruz et al. 2015).

Recently some high-quality research projects are going on metagenomic data analysis such as Ocean Sampling Expedition and Human Microbiome Project (Rusch et al. 2007). These research and scientific analysis are significant at all levels of metagenomic orientations. However, excessive cost affects the analysis. To ensure better results and impact, representations of genomes and its factors are critical. Graphical metagenomic data representations help to assess the factors with clear ideas as well as configurations. One-dimensional, two-dimensional and multi-dimensional representations help to have clear view of genomes. Of course, high-dimensional representations and illustrations are very important for proper genomic view. The key factors that graphical view enables are dissimilarity measurements as well as higher dimensional scaling for all data levels. One popular graphical measurement of metagenomic process is UniFrac; it computes dissimilarity among genomes (Ayyala and Lin 2015).

Mapping-centric graphical representations of metagenomic data enable faster and impactful data representations. In this mapping, genomes are classified into the several groups first. In phylogenetic cases the crab-RNA are organized in a traditional structures (Fig. 1).

Then De Bruijn graph guided to demonstrate the divided genomes into specific order. This order clearly represents the complete datasets over time. Moreover, this graph can part in next-generation sequencing and small read genome assembly. When there are no reference genomes, this graph arranges the genomes in probable orientations. Sometimes most of the sample has proper references that help to adjust the framework accurately visualize. Consequently, De Bruijn graph helps to combine the bacterial genomes and reflects common interest. When two genomes of microbes are not similar, the constructed De Bruijn graph will be mostly different. While if two constructed genomes are similar, then combined genomes are transformed into common structures (Chang et al. 2015; Franzosa et al. 2014; Brown 2015; Wu et al. 2016; Brown et al. 2015; Kang et al. 2015; Gibbons et al. 2015; Deng et al. 2015).

2 Related work

Advanced metagenomic research opens a set of area such as genome variations of profile sample, taxonomic computations, sequence assembly, datasets clustering, binning, protein code predictions and functional assessments of related data and genome referencing. Computational intelligence and machine learning are dominating in a wide range of biological data even genome assembly are easily manageable by advanced data structures along with data mining algorithms (Sato and Sakakibara 2015). In a study, authors have reviewed 25 tools and the sizes are continuously expanding (Bazinet and Cummings 2012). There are set of new challenges for large metagenomic data to handle with effective machine learning solutions. Ongoing research with metagenomic datasets and machine learning environment are generating new dimensions for handling excessively big data to find meaningful and hidden information.

Assembling of metagenomic datasets is critical in recent data mining under machine learning environments. These assemblies permit accurate formation of genomes into database. Moreover, genetic variations, depth of sequencing and genome binning are assured by these assemblies (Sangwan 2016). However, there are some problems during the visualization of genomic datasets with details depth in the assemblies. Consequently, redundancy frequently generates wrong predictions. These problems can be easily overcome by using graph-based approach. De Bruijn graph with Euler path helps to find exact path to represent the whole genomes. Many other significant metagenomic research work focused on microbial function ignoring

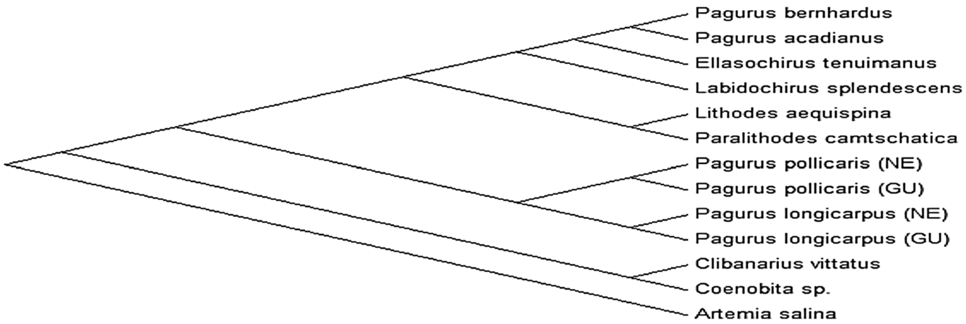

Fig. 1 Phylogenetic tree for Crab-RNA genome sequence

genome structures. Functionalities depict only few features and factors (Markowitz et al. 2014; Hunter et al. 2014; Huang et al. 2014; Sharma et al. 2010). Typical research in this domain includes statistical predictions of genomic functions as well as RNA sequence reads (Leimena et al. 2013) or protein sequences (Franzosa et al. 2014). Some other tools also focused the same such as MG-RAST (Meyer et al. 2008), MEGAN (Huson et al. 2011) and HUMAnN (Abubucker et al. 2012). Text mining processes for phylogenetic motif finding analyze genomes in different dimensions as well as structured. Motif findings permit support for small datasets. This is not suitable for large datasets (Wang et al. 2016). Principal coordinate analysis (PCoA) is used frequently to measure the Euclidean distances between genomes. This distance maintain in a matrix can keep a small amount of values. PCoA is identical to the principal component analysis (PCA) that has great influences in dimension reduction process. GrammaR constructed by PCoA and PCA provides user-friendly graphical genome representation under a choice to remove irregularities as well as multidimensional orientations (Brum et al. 2015; De Vargas et al. 2015; Lima-Mendez et al. 2015; Ten Hoopen et al. 2015). Some other microbial research surveys say about the impact and importance of metagenomic analysis towards the proper visualizations (Gilbert et al. 2014; Pylro et al. 2014; Reddy et al. 2015). Metatranscriptomic synthesis during the gut microbiome orientations for dietary (McNulty 2011) and xenobiotic (Maurice et al. 2013) do not find any changes after making huge changes on genomes functionalities. Moreover, current research work on genomes mainly focused on orientation of the gene structures. So there should have sufficient graphical analysis for metagenomic analysis. Therefore, recent tools have emerged to address these problems for metagenomic reads. Three programs are widely used for this purpose: Orphelia (Hoff 2009), MetaGene (MG) (Noguchi et al. 2006), Meta Gene Annotator (MGA) (Noguchi et al. 2008), and Gene-Mark (Besemer and Borodovsky 1999). Shotgun metagenomic analyses have been done on genomes and microbial datasets for large functionalities. These are also considering the assembly with critical memory efficiency. However, this analysis is not always effective due to its less graphical structures (Eikmeyer et al. 2013; Schlüter et al. 2008; Wirth et al. 2012).

VirAmp (Yinan et al. 2015) is a combined assembler that compared with traditional assembler by web-based graphical user interface. This assembler supports data grouping in parallel process. The parallel process performs in a single platform for large biological data processing and provides a user-interactive platform for the users. However, this package does not efficiently handle the overlapped genomes and time complexity is high for interactive genome sets. Bridgers (Chang et al. 2015) is an application system that measures the genome rearrangement by the help of de novo assembler. In this tool Cufflinks algorithm is used to overcome the limitations of de novo assembler. It needs less computational time and storage than other assemblers. But this tool does not fit in accuracy and sensitivity of Cufflinks algorithm and does not efficiently handle the overlap genome. ClusDCA (Wang and Cho 2015) is an ontological based approach that rearranges the information for all biological datasets that have unique activity of gene annotation function. In interconnected process, ontology takes more time for data mapping. Edena (Hernandez et al. 2008) is another graph-based de novo assembler that follows the procedure of another graph-based assembler. This approach used suffix tree to handle the overlap genome sequence. Edena used heuristic approach for finding overlap length gene and construct a bidirectional graph. However graph traversing cost is too much high along with high space complexity. A mapping-based algorithm can overcome the problem where reads are mapping into short read by using de Bruijn graph (Yuzhen and Haixu 2015). Hashing function and other data structure techniques are used to handle the *k*-mers for graph mapping. This technique is used in metagenomic transcription to utilize the metagenome data.

Recently a significant number of techniques are used for gene annotation or gene prediction. An ensemble gene selection method is used for cancer gene prediction that contains conditional mutual information (Liu et al. 2010).

Multiple gene subsets serve to train the prediction approach and outputs are combined with ranking approach. Multiple filters and multiple wrapper approach (Leung and Hung 2010) enhance the accuracy and robustness of biological data classification for gene selection. Ensemble gene grouping selection (Liu et al. 2010) is another approach that drives multiple gene subsets. This method is based on approximate markov blanket and virtue of information theory. Bolón-Canedo et al. (2012) proposed another gene selection method for ensemble of gene and annotation. A voting approach is used to combine the outputs of gene selection that helps to reduce the variability of features for certain domain. A hybrid generative discriminate approach (Bicego et al. 2012) used biological data for gene selection. Interpretable feature extraction for topics model is used for hybrid approach.

Laplace naïve Bayes (Wu et al. 2012) model for gene classification and annotation. These approaches focus on the robustness of gene outliers and take group effects because of their chemical and electrical reason. Gene pair combination inputs (Chopra et al. 2010) are used for cancer classification algorithm rather than gene original profiler. Supervised and unsupervised approaches (Basford et al. 2013) are used for biological gene prediction. Supervised classification classified the tissues based on specific gene and unsupervised techniques classified the gene based on tissues. A computational protocol (Xu et al. 2010) is used as a gene markers for cancer cells of various cancer tissues. An under-sampling method (Xu et al. 2010) is used the idea of ant colony optimization to predict imbalanced gene data analysis. Association rules (Giugno et al. 2013) are also used for gene classification and prediction, but it needs enhanced system complexity. The author suggested that the transcript expression interval demonstration discriminates subtype in the same class. A web-based interactive tool (Reboiro-Jato et al. 2014) is used to assess the discriminate of hypothesis performance of biological gene datasets. The tool is able to evaluate for medical diagnosis and management decision. Many methods and classification approaches are used to gene pattern. These approaches are applicable and comprehensive for clinical and real practice. The behavior of prediction rules is also used for biological data size (Ives et al. 2004; Raman and Joseph 2001).

3 Methods and materials

The structure of this work is built based on gene annotation (Fig. 2). Our method works based on three phases: data collection, randomized approach, graph generation and gene annotation. The sample data are the dataset of DNA nucleotides of human, plants or micro-organ. Sample data are character of set of DNA nucleotide data stream. Collected DNA data are divided into several parts that are known as sampling operation. In data sampling phase we used randomized approach (Sect. 3.1) for data preprocessing. Then data sampling data is ready for graph generation. Graph generation phase is divided into sub-phases: generation of signed graph and graph reduction. In graph generation phase, we generate an undirected sign graph with multiple edges and loop (Sect. 3.2). Graph reduction rules are used for graph rewriting. Basic reduction rule for graph rewriting technique is used for specification and generation of graph optimization (Sect. 3.3).

After graph optimization we performed gene annotation. In the next phase, gene annotation process is applied with Euler path by using optimize de Brujin graph. We transform de Bruijn graph into series of equivalence sub-graphs. Euler

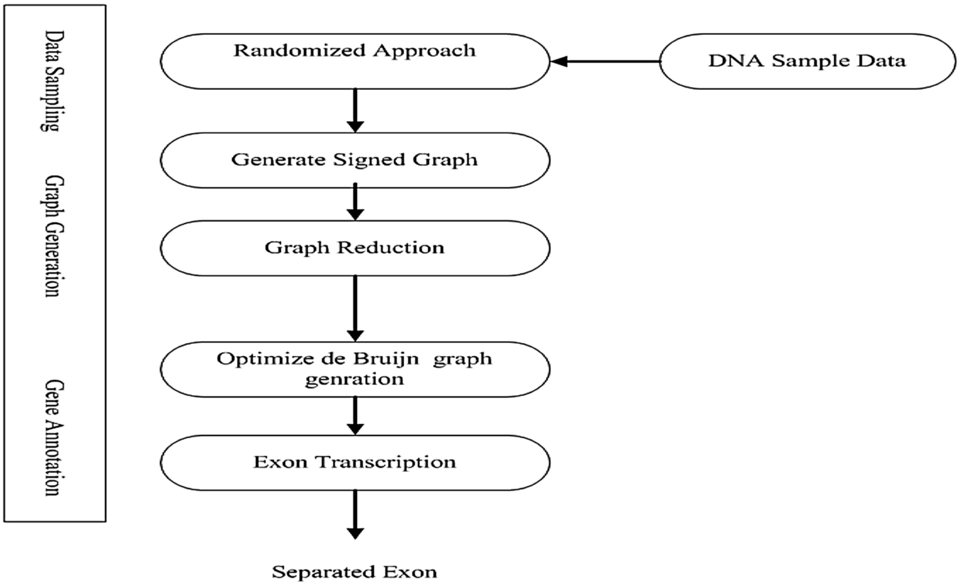

Fig. 2 Semantic view of the proposed methodology for exon separation

paths of all sub-graphs represent the sub-solution of the problem. Euler path is an efficient algorithm that is solved in a linear time. To combine every solution of the sub-graphs represents the solution of gene annotation (Sects. 3.4, 3.5). In exon transcription, we marked initial and stopping sites in optimize de Brujin graph and find out Euler path from initial site to stopping sites (Sect. 3.6). Initial and stopping site indicates the exon annotation region.

3.1 Randomized approach

Data mapping features provide an environment of faster analysis and noise-free computations. Training datasets will be collected from either biological databases or wet laboratories. It is difficult to handle the large biological data. Collected dataset is processed by randomized algorithms. Randomized algorithms provide unique facilities for noise-free and faster data processing. Randomized algorithm considers a rank matrix M_i with some scaling parameters k for ith iteration. Matrix M_i contain two limit parameters as a_j and b_j and primary values a_0 and b_0 satisfies the certain condition such as $a_0 > 0,\ b_0 < 0.$ We can define these two limit parameters a_j and b_j in a systematic way for every iteration that data proportional satisfy the following condition:

$$a_j I < b_j I \tag{1}$$

Eigenvalue of matrix M is measured for limit parameters a and b. An implied function is used to measure the behavior of matrix M eigenvalues between the desired limit parameters:

$$\varphi_{a,b}(M) = \emptyset(aI - M)^{-1} + \emptyset(bI - M)^{-1} \tag{2}$$

An implied iteration function is designed for data sampling is $\theta\left({}^{q}/_{\varepsilon^2}\right)$, here C constant is introduced that exists $a_i \geq Cb_i$ and ε is constant for data subdivision.

```
Randomized Algorithm
j=0
$a_0 = {}^{4q}/_{\varepsilon}$ , $b_0 = {}^{4q}/_{\varepsilon}$
$M_0 = 0$
while $b_j - a_j < {}^{4q}/_{\varepsilon}$ do
    n= $\emptyset(aI - M)^{-1} + \emptyset(bI - M)^{-1}$
    A vector $p_i$ with probability
    $W_i = {}^{p_i^T (b_j I - M_j)^{-1} p_i + p_i^T (M_j - a_j I -)^{-1} p_i}/_{n}$
    $M_{j+1} = M_j + \frac{\varepsilon}{n} . \frac{1}{w_i} . p_i . p_i^T$
    $b_{j+1} = b_j + \frac{\varepsilon}{n(1-\varepsilon)}$
    and $a_{j+1} = a_j + \frac{\varepsilon}{n(1-\varepsilon)}$
    j←j+1
return $M_j$
```

Randomized algorithm is basically used for DNA datasets mapping. DNA datasets mapping are used for data cleaning and integration (Carreira and Helena 2004; Raman et al. 2001; Lenzerini 2002). Cleaning and integration process are responsible for generating system that handle large dataset and peer-to-peer data management system (Raman et al. 2001). DNA datasets mapping is essential because it helps in exon prediction and gene annotation. Basically, mapping is considered as Al-complete problem that data mapping have concentrated on controlled mapping such as one-to-one data schema and structural mapping (Lenzerini 2002).

3.2 Signed graph

In this section, we have introduced basic notations for signed graph. Let $G = (V, E)$ be a finite undirected graph with multiple edges and self loop. The number of $|\upsilon|$ the vertices, is called the order of G and the connected number of $|e|$ is called the degree of G. We write $v \in G$ if υ is vertex and $e \in G$ if e is edge of G. The neighborhood of a vertex υ is $N_G(\upsilon) = \{u|(\upsilon, u) \in G\}$. The vertex υ is isolated if $N_G(\upsilon) = 0$. If a vertex has exactly one neighbor, it is denoted as a leap. We called G is discrete graph if all vertices are isolated. A subset $A \subseteq G$ is stable if there is no edges (υ, u) with $\upsilon, u\ \varepsilon\ A$. Graph A is complete if any two vertices of G are adjacent.

A signed graph $G = (V, E, \phi)$ consists of vertices and edges (V, E) together with a labeling function $\phi{:}V \to \{+, -\}$ of vertices V. A vertex $\upsilon\ \varepsilon\ G$ said to be positive and negative if $\varphi(\upsilon) = +$ and $\varphi(\upsilon) = -$, respectively. We let

$$G^+ = \{\upsilon|\varphi(\upsilon) = +\} \quad \text{and} \quad G^- = \{\upsilon|\varphi(\upsilon) = -\} \tag{3}$$

We say that a signed graph is negative if all its vertices are negative (Fig. 3). Also, an edge $e = \{u, \upsilon\}$ is called negative, if $\upsilon, u \in G^-$.

If G and I constitute two signed graphs and its two disjoint vertex sets are $V(G)$ and $V(I)$, let $G \oplus I$ be their disjoint union, the vertex set of $G \oplus I$ is $V(G) \cup V(I)$ (Fig. 4a) and its edge forms,

$$E(G \oplus I) = E(G) \cup E(I) \tag{4}$$

The complete connection $G \otimes I$ has the vertices set and the edge set is (Fig. 4b).

$$E(G \otimes I) = E(G) \cup E(I) \cup \{(\upsilon, u)|\upsilon \in G, u \in I\}. \tag{5}$$

3.3 Graph reduction rule

There are three basic fundamental operations of reductions rule for signed graph (Villar et al. 2015). The molecular operations are translated into following operations:

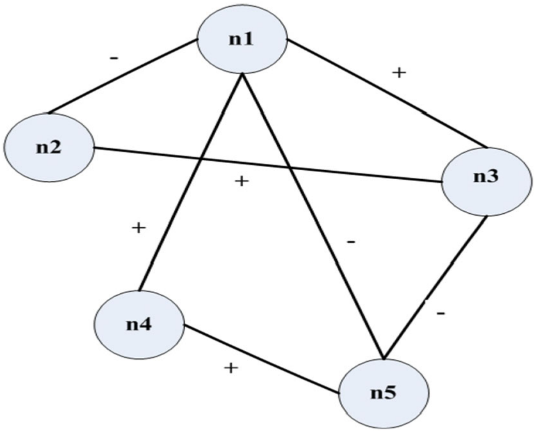

Fig. 3 A simple signed graph with positive and negative edges

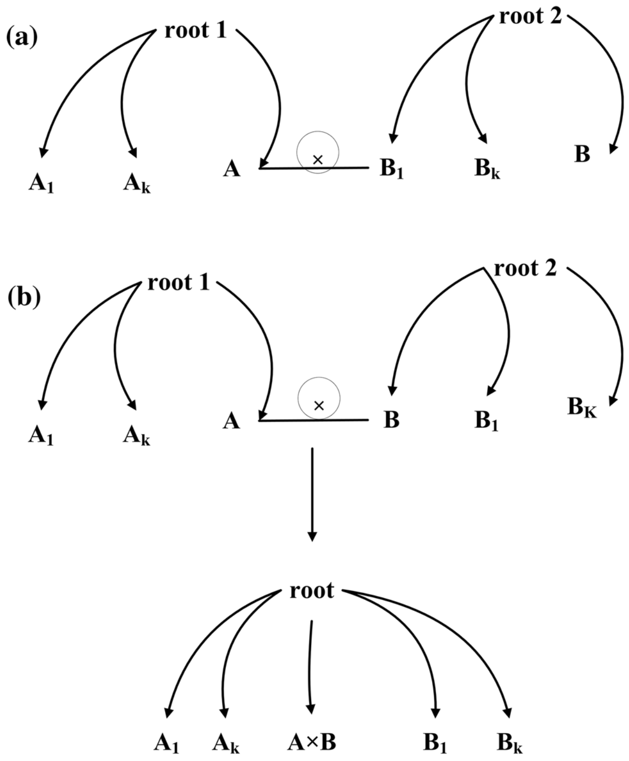

Fig. 4 Operations of signed graph. **a** And operation for two disjoint graphs. **b** Complete operation for disjoint graph

Let u and v be two vertices:

1. The negative graph rule for v is applicable to G if $\upsilon \in G^-$ and it is isolated in G. The result is the signed graph $nr_\upsilon(\upsilon) = G - \{\upsilon\}$. The number of vertices is $|nr_\upsilon| = \{\upsilon\}$.

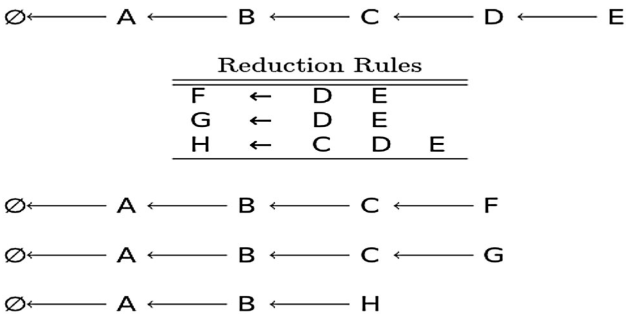

Fig. 5 Graph reduction rule with termination point

2. The positive graph rule for v is applicable to G if $\upsilon \in G^+$. The result is the signed graph $np_\upsilon(\upsilon) = G - \{\upsilon\}$. The number of vertices is $|nr_\upsilon| = \{\upsilon\}$.
3. The double rule for $v \neq$ u is applicable to G if $v, u \in G^-$ and $e = \{\upsilon, u\} \in E(G)$. The result for signed graph $dr(\upsilon) = G - \{(u, \upsilon), E', \varphi'\}$, where φ' is obtained to $G - \{u, \upsilon\}$ and *E*' obtained from the complementary of *E*.

In basic reduction rule, graph rewriting technique is used for specification and generation of graph optimization. Graph analysis and transformation are performed by graph rewriting technique. Analyzing graph means enlarging graph by joining new edges with information and graph transformation means reduced into the graph rewriting by deleting and attaching sub-graphs. In reduction rules, we delete or replace two or more nodes by another node (Fig. 5).

In graph reduction rules, nodes D and E are rewritten by F (Fig. 5). Here D and E nodes can correlate with F, redundant nodes D and E are replaced by F. Nodes C, D and E are rewritten by new node H. Reduced edges are encrypted with new edge. Termination by edges accumulation and termination by edges subtraction is used for graph termination process. When null point (Φ) was reached, it indicated termination (Fig. 5).

3.4 Optimize de Brujin graph

A set of reads $S = \{s_1.......s_n\}$, define the de Bruijn graph $G(S_1)$ with $(l-1)$ vertices. An $(l-1)$ tuple $v \in S_{l-1}$ is joined by directed with *l*-edges. If S_l contains *l*-tuple for which the first $(l-1)$ nucleotides coincide with v and last $(l-1)$ nucleotides coincide with tuple w. Each *l*-tuple from S_l corresponds to an edge in G. If S contains the only sequence S_1, then this sequence corresponds to a path visiting each edge of the de Bruijn graph. A de Bruijn can substitute every edge by k parallel edges, where k is the number of times the edge is used. If S contains the only sequence S_1, this operation creates k parallel edges for

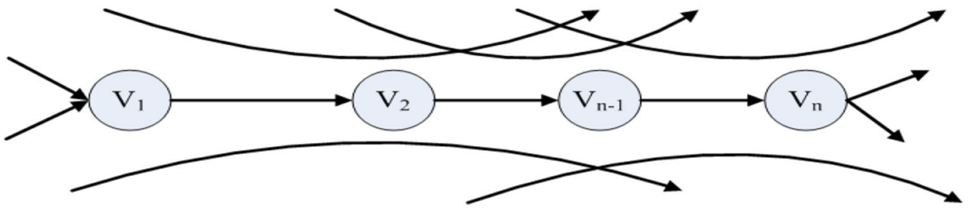

Fig. 6 A repeat v_1…..v_n and arcs are represented by the overlapping the pattern

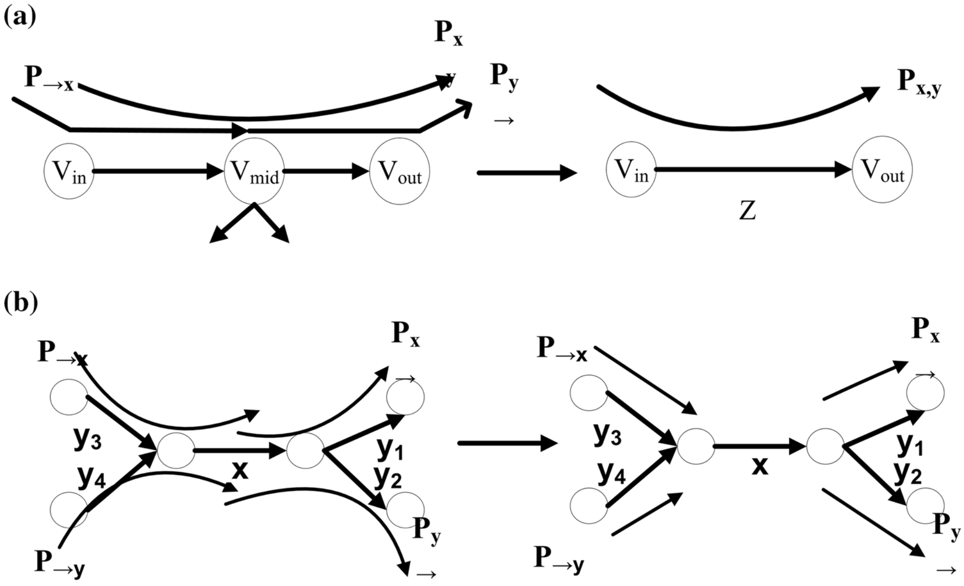

Fig. 7 Equivalent transformation for Euler path; connected edges indicates the overlap paths and disconnected edges indicates equivalent paths. **a** *X*, *Y* –detachment **b** *X*-cut

every l-tuple repeating k times in S_1. Euler path is an efficient algorithm solved in a linear time.

In de Bruijn graph, a vertex v is called a source if indegree(v) $= 0$, a sink if outdegree(v) $= 0$ and branching vertex if indegree(v).outdegree(v) > 1. A path v_1……v_n in the de Bruijn graph is called a repeat pattern if indegree(v_1) > 1, outdegree(v_n) > 1 and indegree(v_i) = outdegree(v_i) $= 1$, for $1 \le i \le n-1$. Repeat pattern starts with v_i node and v_n, are called exits from a repeat (Fig. 6). An Eulerian path visits a repeat some times by visiting entrance and exit nodes. An Eulerian path covers a repeat if it contains an entrance into and exits from repeat by using the end node. Every covering read-path reveals some information about the gene annotation between entrances and exit.

To solve the Euler path for gene annotation, we have transformed both graph G and path P into new graph G_1 with path P_1. This is called equivalence if it exists in one-to-one correspondence in (G,P) and (G_1,P_1).We transform de Bruijn graph into series of equivalence transformation:

$$(G,P) \rightarrow (G_1,P_1) \rightarrow (G_2,P_2) \rightarrow \cdots \rightarrow (G_k,P_k).$$

To combine every solution of sub-graphs represents the solution of gene annotation (exon/introns separation), Euler path solution is used. We describe a simple equivalence transformation that solves the Euler path problem where graph G has no multiple edges. We consider two cases of transformation; one is x–y detachment and the other one is x-cut. Let $x = (v_\text{in}, v_\text{mid})$ and $y = (v_\text{mid}, v_\text{out})$ are two consecutive edges of graph G and $P_{x,y}$ be the collection of all paths of P that includes all sub-paths. $P_{\rightarrow\text{x}}$ defines as a collection of paths from P that end with x and $P_{y\rightarrow}$ as a collection of paths from P that starts with y. Then x,y-detachment is a transformation that adds a new edge $z = (v_\text{in}, v_\text{out})$ and deletes the edges x and y from G (Fig. 7a). This detachment transformation alters the systems of path P as follows:

1. Substitute z for x, y in all path from $P_{x,y}$
2. Substitute z for x in all paths from $P_{\rightarrow x}$.
3. Substitute z for y in all paths from $P_{y\rightarrow}$.

Every detachment reduces the number of edges in G and reduces the complexity of Euler path problem.

Consider a fragment of graph G with 5 or 4 paths $y_3 - x$, $y_4 - x$, $x - y_1$ and $x - y_2$ (Fig. 7b). In symmetric situation, x is tangle (repeated pattern) and there is no available information to relate any of paths $y_3 - x$ and $y_4 - x$ to relate other paths $x - y_1$ and $x - y_{2.\text{A}}$n edge $x = (v, w)$ is removable if

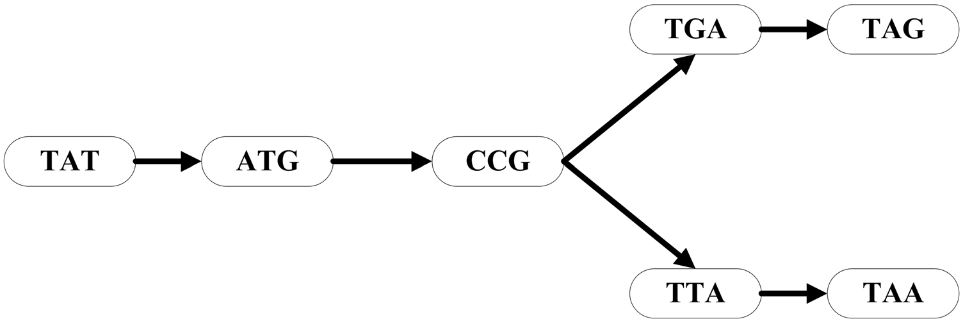

Fig. 8 The length of overlap $k - 1 = 2$. ATG indicates initial site and TAA, TAG node indicates termination site, respectively. *Black line* indicates the order of k-mer

1. It is only incoming edge v and only one outgoing edge w.
2. x is either the initial or the terminal edge for every path P containing x.

An x-cut transformation P into a new system of paths by simply removing x from all paths in $P_{\rightarrow x}$ and $P_{x\rightarrow}$ without affecting graph G itself (Fig. 7b). If x is a removable edge then x-cut is an equivalent transformation. Detachment and x-cut proved to be powerful technique to build a simple de Bruijn graph and reduce fragments for all genomes.

3.5 Annotation with Euler path using de Brujin graph

De Bruijn graph is used for gene annotation (exon intron separation) and next-generation sequence assembler. It reduces the computational effort by breaking read (sort sequence) into smallest part of DNA. Reads are called k-mer where the parameter k denotes the length of bases for these sequences. De Bruijn graph captures exon separation considering exon initial and stop sites using k-mer (Fig. 8). Construct a de Bruijn graph for exon introns where separation consists of the following steps:

1. Structure of k-spectrum: Reads are divided into overlapping sequence of k. Every k-mer consists of a transcription and a stopping site in exon chain.
2. Node generation: Every (k−1) node generates for k-spectrum. In de Bruijn graph, exon initial site is marked as initial node (source), ending node (sink) represents stop site of exon chain and intermediate node serves as donor and acceptor site for exon transcription.
3. Edge construction: A directed edge is created from x node to y node if there exists k-mer such that its prefix is equal to x node to y node. Overlap path is reduced to Euler path simplification which described in Sect. 4. To traverse marked source node to sink node, we had indentified the exon chain from whole DNA sequence.

By reducing whole dataset into k-mer overlaps the de Bruijn graph reduces the high redundancy in short read dataset. In exon annotation, imitation sites start with ATG and termination site with TAA or TAG or TGA. Another donor and acceptor site generates the internal node in de Bruijn graph. Initial and stopping site indicate the exon annotation region (Fig. 8).

By converting the set of reads into edges of the de Bruijn graphs, the annotation problem becomes equivalent to finding an Euler path graph. To reduce the exponential distinct Euler path, heuristics are usually applied to construct the graphs. The graphs are filtered of erroneous occurrences and nodes are unambiguously connected by edge which are merged together (Fig. 7).

3.6 Exon transcription

As exons are not independent, by splicing exons together to assemble a gene one can further eliminate false exon predictions by imposing translatability (i.e., adjacent exons must maintain the open reading frame). The main difficulty in exon assembly is the combinatorial explosion problem: the number of ways N candidate exons may be combined grows exponentially with N. The key idea of computational feasibility comes from dynamic programming (DP), which allows finding “optimal assembly” quickly without having to enumerate all possibilities. We can limit possible errors by assuming each entry with a correct annotation that should satisfy:

- The initiation site is ATG.
- The donor site is GT.
- The acceptor site is AG.
- The stopping site is either TAA or TAG or TGA.
- No stop codon interrupts the open reading frames.
- The length of coding regions is a multiple of three.

In pattern reorganization, the performance of a prediction system can be measured by the following statistics: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The internal exon prediction measurement on the nucleotide base pair level is shown in Figs. 5 and 9.

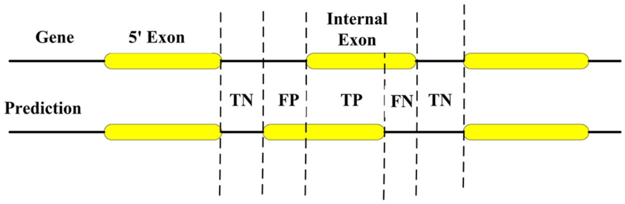

Fig. 9 Measurement parameters for exon prediction

The accuracy of a prediction system is measured by sensitivity (SN), specificity (SP) and F-measure as follows:

$$\mathrm{SN} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6}$$

$$\mathrm{SP} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{7}$$

$$F - \mathrm{measure} = \frac{2 \times \mathrm{SP} \times \mathrm{SN}}{\mathrm{SP} + \mathrm{SN}}. \tag{8}$$

4 Results and discussion

Java environments have been considered for this system design and implementation. During the experiments some key factors such as graph generation, graph reduction and graph optimization have been addressed. Three types of real-world datasets are used for simulation performed here and these dataset are adh22, h178 and sag178 (Table 1). Adh22 is a single sequence of *Drosophila melanogaster* with 2.9 Mb long. Adh22 contains different versions for genome annotation. In the first version adh22 contains 38 genes with 111 exons and the second version consists of 222 genes with 907 exons. H178 has 178 genomic sequences for human that are evaluated from EMBL and GENSCAN. The average sequence length is 716,913 bases. Sag178 is a set of 43 sequences with 178 genes. Graph evaluation and graph generation time are computed for all datasets for different gene sequence lengths. Adh22, h178 and sag178 have different lengths of gene sequences with different number of exons (Table 1).

In Table 1, the first column indicates the three different datasets and rest of the columns indicate the number of gene, exons and base pairs, respectively. Adh22 datasets have maximum genes, exons and base pair than other two datasets. Sag178 has less number of base pairs due to small number of genes. For this metagenomic gene analysis, De Bruijn graph for different data lengths has been generated. Every node contains k-mers with three nucleotides. Comparison between De Bruijn graph and non-De Bruijn graph execution time is measured repeatedly. De Bruijn graph needs less time than non-De Bruijn graph. De Bruijn graph reduces the edge than non-De Bruijn graph. Only the valid directed paths are constructed for exons transcription process, on the other hand non-De Bruijn graph generates multiple edges for exons annotation (Table 2).

Table 1 Different gene length, exons and base pair for three datasets

Dataset	Number of genes	Number of exons	Base pair
Adh22	222	907	898,702
H178	178	845	716,913
Sag178	156	756	632,420

Table 2 Execution time of De Bruijn graph and non-De Bruijn graph for different data lengths of adh22 dataset

Data size (bp)	Non-De Bruijn graph generation	De Bruijn graph generation
100,000	2824	1150
200,000	4257	1337
300,000	4980	1552
400,000	6689	2705
500,000	7213	2955
600,000	7934	2974
700,000	8134	3074
800,000	9967	3133
850,000	11,067	3384

Execution time of De Bruijn graph varies for different DNA lengths. Both execution times of De Bruijn graph and non-De Bruijn approaches are gradually increased due to increase in base pairs. De Bruijn graph generation approach required less time than non-De Bruijn graph process due to small size of biological data. When the base pair is 500,000, the execution time of De Bruijn graph generation is 2955 ns and non-De Bruijn graph process is 7213 ns. De Bruijn graph process is $(7213 - 2955)/7213 = 59.03$ % faster than non-De Bruijn graph process. De Bruijn graph process required less time because De Bruijn graph nodes consider only exons transcription k-mers. Graphical representations of the same computing also reflect the impact of both times (Fig. 10).

Figure 10 depicts the execution time for De Bruijn graph and non-De Bruijn graph generation. The execution time of De Bruijn graph generation for adh22 dataset needs less time than non-De Bruijn graph generation. The graph generation of De Bruijn graph and non-De Bruijn graph for h178 and sag178 requires similar execution time as adh22 datasets, though sag178 and h178 have less base pairs than adh22.

We used randomized De Bruijn graph for DNA categorizing in a specific format. Randomized algorithm is used for sampling data in a specific format. Randomized algorithm for De Bruijn graph has two phases: sampling and pre-data analysis. Sampling indicates splitting DNA

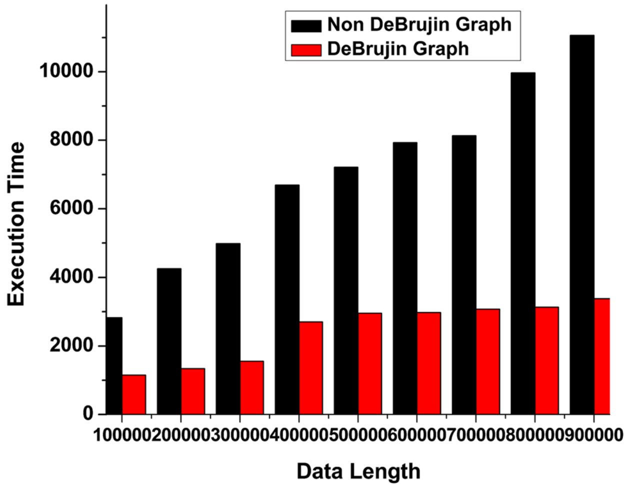

Fig. 10 Execution time for graph generation for De Bruijn graph and non-De Bruijn graph for adh22

Table 3 Execution time of randomized De Bruijn graph algorithm and nonrandomized De Bruijn graph process for different data lengths

Data size (bp)	Nonrandomized De Bruijn graph	Randomized De Bruijn graph
100,000	2825	1375
200,000	6108	2936
300,000	8493	4140
400,000	10,021	5334
500,000	14,084	6974
600,000	14,326	7576
700,000	15,573	8040
800,000	16,110	8354
850,000	16,469	10,147

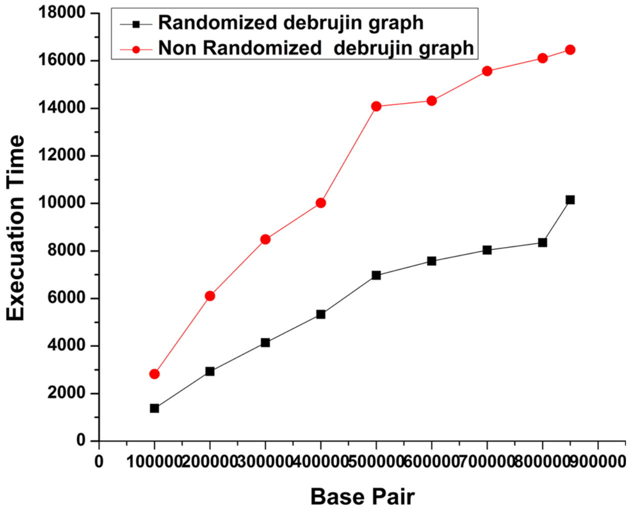

Fig. 11 Execution time analysis of randomized De Bruijn graph and nonrandomized De Bruijn graph approach for DNA data sampling

sequences. It is an important step for sampling distribution because without proper subdivision, it is difficult to handle large DNA dataset. Weights are assigned for finding DNA factors. Weights are considered for threshold value for DNA sampling. DNA sampling data are selected based on threshold value. Metagenomic data analysis is more accurate after data sampling by randomized algorithm. At first we are sampling the DNA sequence for graph generation. We measure the execution time for randomized De Bruijn graph and non-randomized De Bruijn graph data sampling (Table 3).

Execution time of non-randomized De Bruijn graph process and randomized De Bruijn graph process varies for different base pairs. Both execution times of randomized algorithm and non-randomized approaches are gradually increased due to base pairs increased. Randomized approach required less time than non-randomized process due to sample size of sampling data. When the base pair is 600,000, the execution time of randomized approach is 7576 ns and non-randomized process is 12,326 ns. The randomized process is $(12{,}326 - 7576)/12{,}326 = 38.54\ \%$ faster than nonrandomized De Bruijn graph process. Randomized De Bruijn graph process required less time because DNA data are smaller and precise for subdivision and data preprocessing.

Execution time of randomized and nonrandomized graph is increased linearly (Fig. 11). Figure 11 depicts a line graph that indicates the execution time of randomized and nonrandomized De Bruijn graph for different base lengths.

Figure 11 depicts execution time for randomized and nonrandomized approach for DNA data sampling. Randomized process reduced the data length for metagenomic data analysis that takes less execution time than nonrandomized DNA data. The execution time of randomized approach needs more when DNA sequence length is increased. Data subdivision process of randomized algorithm required more time when it generates more splitted portions. In graph reduction phase, sign graphs are optimized for graph simplification. Simplified graph reduces the execution time for exon finding. In the graph reduction phase, nodes of the graph that contain *k*-mers are reduced. This reduced process simplified the graph. When the graph is simplified, exon-finding operation becomes easier by using reduced *k*-mers nodes. Graph-reducing approach reduces execution time than non-reduction graph (Table 3). Non-reduction graph consists of multiple and redundant nodes that are responsible for graph and time complexity (Table 4).

Execution time of non-reduction and graph reduction processes varies for different base pairs. Both execution times of optimized De Bruijn graph and De Bruijn graph

Table 4 Execution time of optimized De Bruijn graph and De Bruijn graph process for different data lengths

Data size (bp)	De Bruijn graph	Optimized De Bruijn graph
100,000	2957	1507
200,000	6240	3068
300,000	8625	4272
400,000	10,153	5466
500,000	14,216	7106
600,000	12,458	7708
700,000	15,705	8172
800,000	16,242	8486
850,000	16,601	10,279

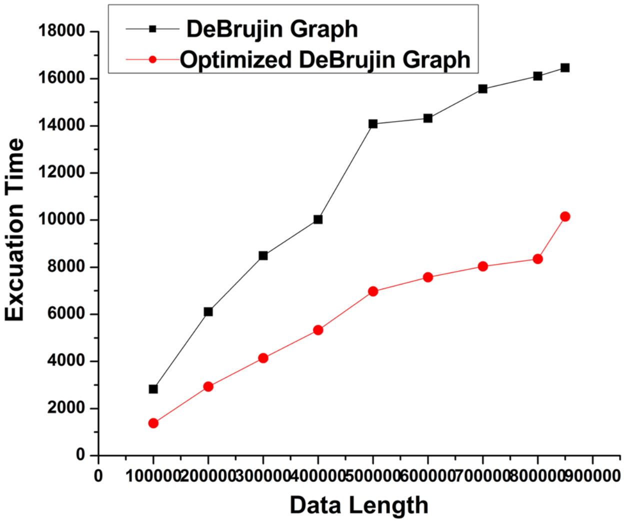

Fig. 12 Execution time analysis of graph reduction and non-reduction approach for different base pairs

approaches are gradually increased due to increase in base pairs. Optimized De Bruijn graph needs lesser time than simple De Bruijn graph process. When the base pair is 500,000, the execution time of optimized De Bruijn graph is 7106 ns and De Bruijn graph process is 14,216 ns. Optimized De Bruijn graph process is nearly about two times faster than simple De Bruijn graph process. Optimized De Bruijn graph reduces the unnecessary nodes that consist of *k*-mers for exon annotation. In simple De Bruijn graph process, multiple nodes have to traverse for exon finding that does not provide optimal solution and need more execution time (Fig. 12).

We measured the accuracy, sensitivity and specificity for exons prediction and gene annotation. Predicted exons are correct if splice sites are at the annotation position. Predicted gene is correct if all exons predicted are correctly predicted. We also measure false positive that indicates when some exons are partially predicted. For each data set gene prediction and exons prediction are measured globally. Euler path approach is used in optimized De Bruijn graph for exon prediction and gene annotation. We compare our result with another gene annotation approach, GENESCAN and GENEID. GENSCAN as it is the most commonly used gene annotation program for human's genome. Optimized graph-based approach of Euler path provides more accurate output than GENESCAN. GENEID (version 1.1) is an exon-finding approach more suitable for Drosophila. GENESCAN is a program that identifies the gene structure. It is a GHMM-based program that can be used to predict gene annotation and exon introns boundaries (Mochizuki et al. 2011). GENESCAN performs two-phase gene prediction structure: statistical pattern identification and sequence similarity comparison. GENEID is a gene prediction program with a hierarchical structure (Parra et al. 2000). GENEID used position weight matrices (PWMS) that build the exon generation site. We also compare our approach with GENEID for exon finding with different base pairs. We calculate sensitivity and specificity for exon prediction for GENEID and optimized De Bruijn graph.

In Table 5, the first column indicates the three datasets and second column indicates the prediction criteria. Optimized De Bruijn graph is more accurate than GENESCAN and GENEID for base pair analysis, exon prediction and gene annotation. Sensitivity and specificity are higher for base pair and exon prediction than gene annotation process. Sensitivity and specificity are low for gene annotation, because it is difficult to predict all exon predictions accurately. Optimized De Bruijn graph analysis measures sensitivity of base analysis, exon prediction and gene annotation 91.4, 60.3 and 35.82 %, respectively, for adh22 datasets. For adh22 dataset, optimized De Bruijn graph measures 62.3 % sensitivity for exon prediction, whereas GENESCAN and GENEID measure 61.1 and 57.8 % sensitivity for exon prediction. Optimized De Bruijn graph measures higher sensitivity and specificity for h178 datasets than adh22. It is difficult for large datasets to predict exon and introns splices for whole gene annotation that measure the less sensitivity and specificity than other criteria. We measure better result for h178 dataset for every base pair analysis, exon prediction and gene annotation. Sag178 datasets predict less sensitivity and specificity for exon prediction and gene annotation. On the other hand, GENESCAN measures better result for every dataset for human genome analysis than GENEID process.

Figure 13a depicts the measurement of specificity and sensitivity of exon prediction for our collected dataset. Our approach is to more accurately measure the exon pattern from whole genome sequence than GENESCAN and GENEID for exon prediction. Optimized De Bruijn graph operates only those *k*-mers that are responsible for exon

Table 5 Sensitivity and specificity measurement for gene annotation and exon prediction

Datasets	Prediction criteria	Methods		
		Optimized De Bruijn graph (%)	GENESCAN (%)	GENEID (%)
Adh22	Base			
	SP	96.3	96.2	95.2
	SN	91.4	91.2	90.3
	Exon			
	SP	72.4	71.1	69.4
	SN	62.3	61.1	57.8
	Gene			
	SP	49.12	45.58	38.41
	SN	35.82	31.58	27.54
H178	Base			
	SP	97.12	95.55	87.32
	SN	88.12	84.57	89.22
	Exon			
	SP	83.82	81.77	64.97
	SN	75.62	73.32	73.58
	Gene			
	SP	39.62	38.53	12.55
	SN	38.12	34.58	13.61
Sag178	Base			
	SP	94.12	92.76	87.66
	SN	84.32	62.77	76.27
	Exon			
	SP	68.52	66.90	65.22
	SN	62.12	43.11	58.31
	Gene			
	SP	21.12	16.13	15.34
	SN	23.12	12.58	15.58

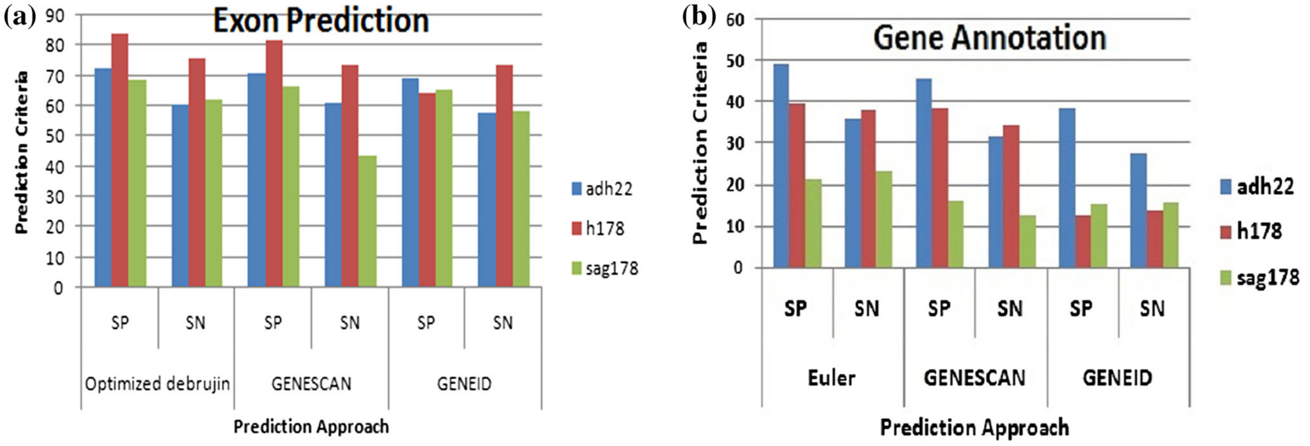

Fig. 13 Sensitivity and specificity prediction for **a** exon prediction, **b** gene annotation using optimized De Bruijn graph, GENESCAN and GENEID approach

generation. By finding optimal analysis, optimized De Bruijn graph accurately measures the exon pattern. GENESCAN more accurately measures the exon pattern than GENEID approach for human genome. GENEID approach has failed for whole exon pattern for whole human large DNA sequences. We also measured the gene annotation

(Fig. 13b) for different datasets. Optimized De Bruijn graph more accurately measures the gene than the other two approaches. Optimized De Bruijn graph approach, (49.12 − 38.41)/49.12 = 21.80 %, has more specificity than GENEID approach for adh22 dataset. GENEID measures less accurate result for gene annotation for human genome, but this approach is accurate for *Drosophila* (Parra et al. 2000). GENESCAN approach measures more accurately for gene than GENEID for all datasets. By combining both measures, we use the *f*-measure and accuracy for exon and gene prediction. A good indictor indicates false positive and accuracy that are measured by specificity and sensitivity. *F*-measure indicates that exon pattern is wrongly predicted and accuracy indicates the rate of accurate measurement of the exon pattern and gene annotation.

Table 6 indicates the *f*-measure and accuracy rate for gene annotation and exon prediction for three datasets. When accuracy rate is increased, *f*-measure decreases; high accuracy indicates maximum correct gene annotation process. Our optimized De Bruijn graph approach measures high accuracy and less *f*-measure for gene annotation than other two prediction approaches. For adh22 dataset, optimized De Bruijn graph approach accuracy is

Table 6 F-measure and accuracy for gene annotation and exon prediction

Datasets	Prediction criteria	Method		
		Optimized De Bruijn graph (%)	GENESCAN (%)	GENEID (%)
Adh22	Exon			
	F-measure	12.7	14.8	16.6
	Accuracy	87.3	85.2	83.4
	Gene			
	F-measure	23.2	26.4	31.2
	Accuracy	76.8	73.6	68.8
H178	Exon			
	F-measure	11.7	13.7	17.6
	Accuracy	88.3	86.3	82.4
	Gene			
	F-measure	25.2	27.8	30.2
	Accuracy	74.8	72.2	69.8
Sag178	Exon			
	F-measure	17.7	17.6	22.6
	Accuracy	82.3	82.4	77.4
	Gene			
	F-measure	22.1	28.4	34.2
	Accuracy	77.9	71.6	65.8

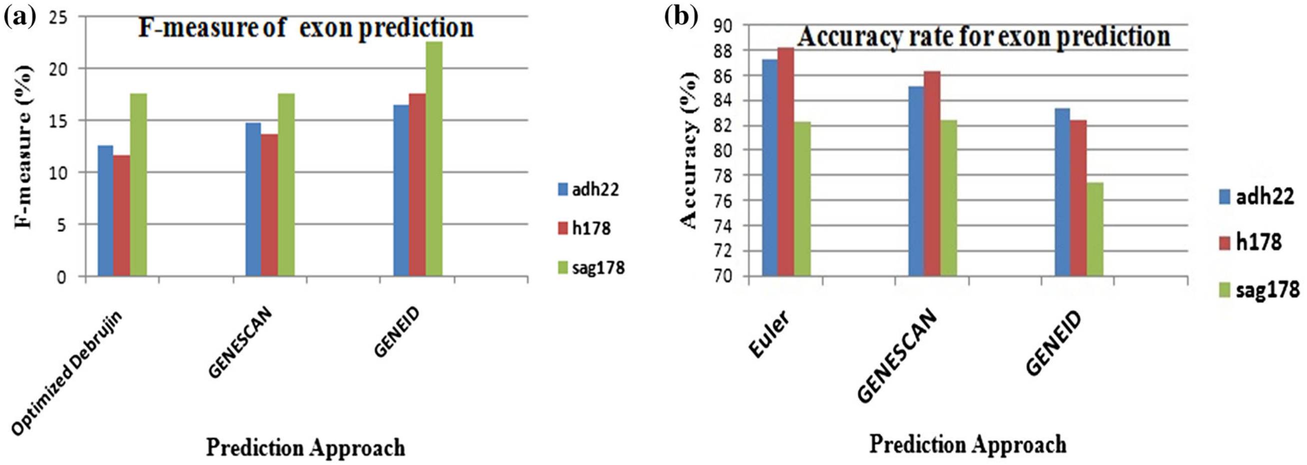

Fig. 14 **a** *F*-measurement and **b** accuracy prediction for exon prediction using optimized De Bruijn graph, GENESCAN and GENEID approach

(87.3 − 85.2)/87.3 = 2.41, higher than GENESCAN for exon prediction. For similar datasets our approach is 4.47 %; more accurate exon prediction approach than GENEID. For h178 datasets our optimized De Bruijn graph measures less *f*-measure than other two approaches which means our approach is more accurate than the others. Optimized De Bruijn graph measures less *f*-measurement for sag178 datasets, because it has less base pairs, that is, nearly about 650,000 base pairs.

All possible exon prediction is important for accurate gene annotation. Internal exons also have flanking splicing boundaries: the acceptor splicing sites at the 5′ end and the donor sites at the 3′ end. In the optimized De Bruijn graph, Euler approach selects donor and acceptor region of exon prediction more efficiently than GENESCAN and GENEID. Potentially all of the selected donor site and acceptor site candidates can be paired to form exon boundaries. The number of internal exons in a gene is one less than the number of introns.

Accurate gene annotation depends on perfect exon prediction. Our optimized De Bruijn graph approach predicts the exon pattern better than other prediction algorithms (Fig. 14b). Our optimized De Bruijn graph is more accurate and has less *f*-measure for adh22, sag178 and h178. Our approach accuracy is higher than GENEID and GENESCAN for exon prediction. Optimized De Bruijn graph method also provides optimal solutions for gene annotation.

5 Conclusion

In this research, we have observed that graph theory gives better accuracy than the other two models.

Our method is robust that continuously free the memory storage. In fact, our simulation result indicates that it is more accurate for a large dataset. It performs relatively well on the task of assembling exons to genes, because programs with a similar exon-level accuracy often have a lower gene-level accuracy. This means those programs more often combine the exons to a wrong gene structure, for example by splitting or joining genes. With the growing number of sequenced species, the possibilities of finding approximate possible exons by cross-species alignments of homologous genomic sequences also increase. This leaves the task of assembling possible exons to genes. In future, we shall consider this concept for finding possible intron analysis.

References

Abubucker S et al (2012) Metabolic reconstruction for metagenomic data and its application to the human microbiome. PLoS Comput Biol 8:e1002358

Ayyala DN, Lin S (2015) GrammR: graphical representation and modeling of count data with application in metagenomics. Bioinformatics 31(10):1648–1654

Basford KE, McLachlan GJ, Rathnayake SI (2013) On the classification of microarray gene-expression data. Brief Bioinform 14(4):402–410

Bazinet A, Cummings M (2012) A comparative evaluation of sequence classification programs. BMC Bioinform 13:1–13

Besemer J, Borodovsky M (1999) Heuristic approach to deriving models for gene finding. Nucleic Acids Res 27(19): 3911–3920

Bicego M, Lovato P, Perina A, Fasoli M, Delledonne M, Pezzotti M et al (2012) Investigating topic models' capabilities in expression microarray data classification. IEEE/ACM Trans Comput Biol Bioinform (TCBB) 9(6):1831–1836

Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30:2114–2120

Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A (2012) An ensemble of filters and classifiers for microarray data classification. Pattern Recogn 45(1):531–539

Brown CT (2015) Strain recovery from metagenomes. Nat Biotechnol 33:1041–1043

Brown CT, Hug LA, Thomas BC, Sharon I, Castelle CJ, Singh A et al (2015) Unusual biology across a group comprising more than 15% of domain bacteria. Nature 523:208–211

Brum JR, Ignacio-Espinoza JC, Roux S, Doulcier G, Acinas SG, Alberti A, Chaffron S, Cruaud C, de Vargas C, Gasol JM et al (2015) Ocean plankton. Patterns and ecological drivers of ocean viral communities. Science 348:1261498

Chang Z et al (2015a) Bridger: a new framework for de novo transcriptome assembly using RNA-seq data. Genome Biol 16:30

Chang Z, Li G, Li J, Zhang Y, Ashby C, Liu D, Cramer C, Huang X (2015b) Bridger: a new framework for de novo transcriptome assembly using RNA-seq data. Genome Biol 16:30

Chopra P, Lee J, Kang J, Lee S (2010) Improving cancer classification accuracy using gene pairs. PLoS One 5(12):e14305

De Cruz P, Kang S, Wagner J, Buckley M, Sim WH, Prideaux L et al (2015) Association between specific mucosa-associated microbiota in Crohn's disease at the time of resection and subsequent disease recurrence: a pilot study. J Gastroenterol Hepatol 30:268–278

De Vargas C, Audic S, Henry N, Decelle J, Mahé F, Logares R, Lara E, Berney C, Le Bescot N, Probert I et al (2015) Ocean plankton. Eukaryotic plankton diversity in the sunlit ocean. Science 348:1261605

Deng X, Naccache SN, Ng T, Federman S, Li L, Chiu CY et al (2015) An ensemble strategy that significantly improves de novo assembly of microbial genomes from metagenomic next-generation sequencing data. Nucleic Acids Res 43(7):e46

Eikmeyer FG, Rademacher A, Hanreich A, Hennig M, Jaenicke S, Maus I, Wibberg D, Zakrzewski M, Pühler A, Klocke M (2013) Detailed analysis of metagenome datasets obtained from biogas-producing microbial communities residing in biogas reactors does not indicate the presence of putative pathogenic microorganisms. Biotechnol Biofuels 6(1):49

Forster SC, Lawley TD (2015) Systematic discovery of probiotics. Nat Biotechnol 33:47–49

Franzosa EA et al (2014) Relating the metatranscriptome and metagenome of the human gut. Proc Natl Acad Sci USA 111:E2329–E2338

Gibbons SM, Schwartz T, Fouquier J, Mitchell M, Sangwan N, Gilbert JA et al (2015) Ecological succession and viability of human-associated microbiota on restroom surfaces. Appl Environ Microbiol 81:765–773

Gilbert JA, Jansson JK, Knight R (2014) The Earth Microbiome project: successes and aspirations. BMC Biol 12:69. doi:10.1186/s12915-014-0069-1

Giugno R, Pulvirenti A, Cascione L, Pigola G, Ferro A (2013) MIDClass: microarray data classification by association rules and gene expression intervals. PLoS One 8(8):e69873

Hernandez D (2008) De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer. Genome Res 18:802–809

Hoff KJ, Lingner T, Meinicke P, Tech M (2009) Orphelia: predicting genes in metagenomic sequencing reads. Nucleic Acids Res 37:W101–W105 **(Web Server)**

Hsiao A, Ahmed AM, Subramanian S, Griffin NW, Drewry LL, Petri WA Jr, Haque R, Ahmed T, Gordon JI (2014) Members of the human gut microbiota involved in recovery from *Vibrio* cholerae infection. Nature 515:423–426

Huang K, Brady A, Mahurkar A, White O, Gevers D, Huttenhower C, Segata N (2014) MetaRef: a pan-genomic database for comparative and community microbial genomics. Nucleic Acids Res 42:D617–D624

Hultman J, Waldrop MP, Mackelprang R, David MM, McFarland J, Blazewicz SJ et al (2015) Multi-omics of permafrost, active layer and thermokarst bog soil microbiomes. Nature 521:208–212

Hunter S, Corbett M, Denise H, Fraser M, Gonzalez-Beltran A, Hunter C, Jones P, Leinonen R, McAnulla C, Maguire E et al (2014) EBI metagenomics—a new resource for the analysis and archiving of metagenomic data. Nucleic Acids Res 42:D600–D606

Huson DH et al (2011) Integrative analysis of environmental sequences using MEGAN4. Genome Res 21:1552–1560

Ives Z, Alon Y, Mork P, Tatarinov I (2004) Piazza: mediation and integration infrastructure for semantic web data. J Web Sem 1(2):155–175

Jing X-Y, Zhang D, Tang Y-Y (2004) An improved LDA approach. IEEE Trans Syst Man Cybern B Cybern 34(5):1942–1951

Kang DD, Froula J, Egan R, Wang Z (2015) MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. PeerJ 3:e1165

Kopf A, Bicak M, Kottmann R, Schnetzer J, Kostadinov I, Lehmann K, Fernandez-Guerra A, Jeanthon C, Rahav E, Ullrich M et al (2015) The ocean sampling day consortium. Gigascience 4:27

Leung Y, Hung Y (2010) A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. IEEE/ACM Trans Comput Biol Bioinform (TCBB) 7(1):108–117

Leimena MM et al (2013) A comprehensive metatranscriptome analysis pipeline and its validation using human small intestine microbiota datasets. BMC Genom 14:530

Lima-Mendez G, Faust K, Henry N, Decelle J, Colin S, Carcillo F, Chaffron S, Ignacio-Espinosa JC, Roux S, Vincent F et al (2015) Ocean plankton. Determinants of community structure in the global plankton interactome. Science 348(6237):1262073

Liu H, Liu L, Zhang H (2010a) Ensemble gene selection by grouping for microarray data classification. J Biomed Inform 43(1):81–87

Liu H, Liu L, Zhang H (2010b) Ensemble gene selection for cancer classification. Pattern Recogn 43(8):2763–2772

Lozupone C, Lladser ME, Knights D, Stombaugh J, Knight R (2011) UniFrac: an effective distance metric for microbial community comparison. ISME J 5(2):169–172

Lenzerini M (2002) Data integration: a theoretical perspective. Proc ACM PODS, Madison, WI, pp 233–246

Lu H, Qian G, Ren Z et al (2015) Alterations of *Bacteroides* sp., *Neisseria* sp., *Actinomyces* sp., and *Streptococcus* sp. populations in the oropharyngeal microbiome are associated with liver cirrhosis and pneumonia. BMC Infect Dis 15(1):239

Markowitz VM, Chen IM, Palaniappan K, Chu K, Szeto E, Pillay M, Ratner A, Huang J, Woyke T, Huntemann M et al (2014) IMG 4 version of the integrated microbial genomes comparative analysis system. Nucleic Acids Res 42:D560–D567

Maurice CF, Haiser HJ, Turnbaugh PJ (2013) Xenobiotics shape the physiology and gene expression of the active human gut microbiome. Cell 152(1–2):39–50

McNulty NP et al (2011) The impact of a consortium of fermented milk strains on the gut microbiome of gnotobiotic mice and monozygotic twins. Sci Transl Med 3(106):ra106

Meyer F et al (2008) The metagenomics RAST server—a public resource for the automatic phylogenetic and functional analysis of metagenomes. BMC Bioinform 9:386

Mitchell A, Chang H-Y, Daugherty L, Fraser M, Hunter S, Lopez R, McAnulla C, McMenamin C, Nuka G, Pesseat S et al (2015) The InterPro protein families database: the classification resource after 15 years. Nucleic Acids Res 43:D213–D221

Mochizuki H, Nakamura K, Sato H, Goto-Koshino Y, Sato M, Takahashi M, Fujino Y, Ohno K (2011) Multiplex PCR and Genescan analysis to detect immunoglobulin heavy chain gene rearrangement in feline B-cell neoplasms. Vet Immunol Immunopathol 143(2011):38–45

Noguchi H, Park J, Takagi T (2006) MetaGene: prokaryotic gene finding from environmental genome shotgun sequences. Nucleic Acids Res 34(19):5623–5630

Noguchi H, Taniguchi T, Itoh T (2008) Meta gene annotator: detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. DNA Res 15(6):387–396

Li P, Yang C, Xie J et al (2015) *Acinetobacter calcoaceticus* from a fatal case of pneumonia harboring blaNDM-1 on a widely distributed plasmid. BMC Infect Dis 15(131)

Parra G, Blanco E, Guigo R (2000) GeneID in *Drosophila*. Genome Res 10:511–515

Carreira P, Helena G (2004) Execution of data mappers. Proc ACM SIGMOD workshop IQIS, Paris, France, pp 2–9

Pylro VS, Roesch L, Ortega JM, do Amaral AM (2014) Brazilian microbiome project: revealing the unexplored microbial diversity challenges and prospects. Microb Ecol 67:237–241. doi:10.1007/s00248-013-0302-4

Raman V, Joseph MH (2001) Potter's Wheel: an interactive data cleaning system. Proc VLDB Conf, Roma, Italy, pp 381–390

Reboiro-Jato M, Arrais JP, Oliveira JL, Fdez-Riverola F (2014) geneCommittee: a web-based tool for extensively testing the discriminatory power of biologically relevant gene sets in microarray data classification. BMC Bioinform 15(1):31

Reddy TBK, Thomas AD, Stamatis D, Bertsch J, Isbandi M, Jansson J, Mallajosyula J, Pagani I, Lobos EA, Kyrpides NC (2015) The Genomes OnLine Database (GOLD) v. 5: a metadata management system based on a four level (meta)genome project classification. Nucleic Acids Res 43:D1099–D1106

Rusch DB, Halpern AL, Sutton G, Heidelberg KB, Williamson S, Yooseph S, Wu D, Eisen JA, Hoffman JM, Remington K et al (2007) The Sorcerer II global ocean sampling expedition: northwest Atlantic through eastern tropical Pacific. PLoS Biol 5:e77

Sangwan N, Xia F, Gilbert JA (2016) Recovering complete and draft population genomes from metagenome datasets. Microbiome 4:8

Sato K, Sakakibara Y (2015) MetaVelvet-SL: an extension of the Velvet assembler to a de novo metagenomic assembler utilizing supervised learning. DNA Res 22(1):69–77

Schlüter A, Bekel T, Diaz NN, Dondrup M, Eichenlaub R, Gartemann K-H, Krahn I, Krause L, Krömeke H, Kruse O (2008) The metagenome of a biogas-producing microbial community of a production-scale biogas plant fermenter analysed by the 454-pyrosequencing technology. J Biotechnol 136(1):77–90

Sharma VK, Kumar N, Prakash T, Taylor TD (2010) MetaBioME: a database to explore commercially useful enzymes in metagenomic datasets. Nucleic Acids Res 38:D468–D472

Silvester N, Alako B, Amid C, Cerdeno-Tarraga A, Cleland I, Gibson R, Goodgame N, Ten Hoopen P, Kay S, Leinonen R et al (2015) Content discovery and retrieval services at the European Nucleotide Archive. Nucleic Acids Res 43:D23–D29

Sunagawa S, Coelho LP, Chaffron S, Kultima JR, Labadie K, Salazar G, Djahanschiri B, Zeller G, Mende DR, Alberti A et al (2015) Ocean plankton. Structure and function of the global ocean microbiome. Science 348:1261359

Freitas TAK, Li PE, Scholz MB, Chain PSG (2015) Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. Nucleic Acids Res 1. doi:10.1093/nar/gkv180

Ten Hoopen P, Pesant S, Kottmann R, Kopf A, Bicak M, Claus S, Deneudt K, Borremans C, Thijsse P, Dekeyzer S et al (2015) Marine microbial biodiversity, bioinformatics and biotechnology (M2B3) data reporting and service standards. Stand Genomic Sci. 10:20

Villar E, Farrant GK, Follows M, Garczarek L, Speich S, Audic S, Bittner L, Blanke B, Brum JR, Brunet C et al (2015) Ocean plankton. Environmental characteristics of Agulhas rings affect interocean plankton transport. Science 348:1261447

Wang S, Cho H, Zhai CX, Berger B, Peng J (2015) Exploiting ontology graph for predicting sparsely annotated gene function. Bioinformatics 31:i357–i364

Wirth R, Kovács E, Maróti G, Bagi Z, Rákhely G, Kovács KL (2012) Characterization of a biogas-producing microbial community by short-read next generation DNA sequencing. Biotechnol Biofuels 5(1):41

Wu MY, Dai DQ, Shi Y, Yan H, Zhang XF (2012) Biomarker identification and cancer classification based on microarray data using laplace naive Bayes model with mean shrinkage. IEEE/ACM Trans Comput Biol Bioinform (TCBB) 9(6):1649–1662

Wu Y-W, Simmons BA, Singer SW (2016) MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. Bioinformatics 32(4):605–607

Xu K, Cui J, Olman V, Yang Q, Puett D, Xu Y (2010) A comparative analysis of gene-expression data of multiple cancer types. PLoS One 5(10):e13696

Rahm E, Philip A (2001) A survey of approaches to automatic schema matching. VLDB J 10(4):334–350

Wang Y, Li R, Zhou Y, Ling Z, Guo X, Xie L, Liu L (2016) Motif-based text mining of microbial metagenome redundancy profiling data for disease classification. BioMed Res Int 2016: 11 pages **(Article ID 6598307)**

Yinan W, Renner DW, Albert I, Szpara ML (2015) VirAmp: a galaxy-based viral genome assembly pipeline. GigaScience 4:19

Yuzhen Y, Haixu T (2015) Utilizing de Bruijn graph of metagenome assembly for metatranscriptome analysis. Bioinformatics 32(7):1001–1008