SV-BET: Structure Variation Benchmarking and Evaluation Tool with Comparative Analysis of Split Read-Based Approaches

Eman A. Alzaid[1] and Ghada Badr[1,2]
[1]King Saud University, Riyadh, Saudi Arabia
[2]IRI- The City of Scientific Research and Technological Applications, University and Research District, P.O. 21934, New Borg Alarab, Alexandria, Egypt
Email: e.alzaid@yahoo.com, badrghada@hotmail.com

***Abstract*—Several structural variation identification approaches have been developed with various considerations and merits. As a scientist, it is important to choose the suitable tool. Unfortunately, there is a lack of gold standards to benchmark these approaches. A Structural Variation Benchmarking and Evaluation Tool (SV-BET) is proposed, which composed of three main components: benchmark creator, mapper, and evaluator. We use the proposed tool to evaluate the performances of seven available approaches: Pindel, Prism, Delly, Softsearch, Softsv, Socrates, and Manta. These tools are based on split-read-based approaches for detecting structural variations. SVBET is tested using the Escherichia coli K12 genome. Results show the sensitivity and positive-predictive value of detected structural variations and breakpoints for each approach. SV-BET can be used to evaluate the performance of other SV identification algorithms.**

***Index Terms*—structural variation, split-read, breakpoints, next generation-sequencing**

I. Introduction

The Next-Generation Sequencing (NGS) is a term that is used to describe a set of sequencing platforms that sequence DNA quickly and cheaply than traditional approach. Structural Variation (SV) of a Genome is defined as rearrangements that affect at least 50bp of a sequence [1]. Studying SVs facilitates understanding of the phenotypic variations and the genetic diseases. Some SVs are associated with diseases, such as cancers, autism, schizophrenia, Parkinson's disease, and Alzheimer [2]. They also play an important role in personalized medicine studies, which is assumed to be the only way for discovering a cure for cancer in many cases [3].

A genome Structural Version (SV) can fall into several classes: (1) insertion, where a novel sequence is injected into the genome; (2) deletion, where a region of the sequence is removed; (3) translocation, where a region of the sequence is moved to another location; (4) inversion, where a region of the sequence is inverted; and (4) Copy-Number-Variations (CNVs), which involve changing the copies of a region via deletion or duplication (duplication may be tandem or interspersed). Some of the literature also differentiates between translocation and transposition, since two or more chromosomes are involved in translocation, while transposition happens within the same chromosome.

Due to the low cost of genome sequencing, a variety of methods have been developed to identify SV from the whole genome. These approaches use different features of NGS datasets to predict SVs. Alkan [1] divides these approaches into four classes: Paired-End Mapping (PEM), Split-Read (SR), Read-Depth (RD) and de novo assembly-based. The first three approaches start with mapping the reads to a reference genome. PEM, analyzes the distance and orientation of two mapped ends of the paired-end reads [4]. SR approaches analyze unmapped reads by splitting the read and realigning the splits to detect SVs [5]. RD approaches analysis read coverage to detect coverage variations over different genome regions [6]. De novo assembly approaches, however, build longer fragments from the reads first, then, align these fragments (contigs) with the reference genome to detect SVs [7]. In addition to the aforementioned methods, hybrid approaches have been proposed that combine two or more of the techniques classified by Alkan's.

With the lack of gold standard datasets benchmark for SV [8], there is a need for SV benchmark and tool evaluation. In this paper, a Structural Variation Benchmarking and Evaluation Tool (SVBET) is proposed. The tool is used to compare and evaluate seven currently available split read-based approaches for analyzing whole genome sequencing. These approaches are chosen because of their ability in detecting SV at base pair level.

We evaluate the performance of these tools in detecting different SV classes and sizes. The evaluation is based on measuring sensitivity and runtime using different read coverage. Next section describes brie y the tools that are evaluated. Section 3 describes SV-BET. Section 4 includes the experimental results. Finally, Section 5 concludes the paper.

II. Background

The problem of SV identification using NGS can be defined as follows. Given two inputs: (1) reference

Manuscript received February 14, 2016; July 7, 2016.

doi: 10.18178/ijpmbs.5.4.217-221

genome sequence of length g, and (2) donor genome as a set of *n* overlapped sequences (reads) with average read length *l* and coverage $c = n.l/g$. The required output is a set of SVs. Each SV has a type, a location (breakpoints) and at least k reads that support it. The SV type could be deletion, insertion, transposition, inversion, or CNV (tandem duplication, interspersed duplication and deletion). Insertion has one breakpoint. Deletion, inversion and tandem duplication has two breakpoints. Translocation has four breakpoints.

Except for assembly-based approaches, SV detection approaches start with read aligning using available NGS aligner, such as BWA [9] and Bowtie [10]. Then, the read alignments are analyzed to find SVs. In this work, we only consider the algorithms which are based on SR for detecting SVs due to their spurious in specify the breakpoints at low resolution.

A. *Pindel*

Pindel [11] is the first SR method applied for NGS paired-end reads. It applies a pattern growth algorithm to search for unmapped read-end fragments in a limited genome region that is specified mainly by the orientation of perfectly mapped read-end.

B. *Prism*

Prism [12] uses discordant read mapping to reduce search space of split-read mapping. A paired end read has a discordant mapping if either orientation or distance between the two mapped ends are not as expected. Prism aligns the unmapped read-ends to the regions that have discordant read mapping.

C. *SoftSearch*

SoftSearch [13] avoids realignment in PRISM by analyzing overlapped discordant reads with soft-clipped reads. Soft-clipped reads are reads with one part is mapped to the reference genome and the other part unmapped.

D. *Socrates*

Socrates [14] filters the mapped reads to extract long soft clips sequences and realign them to the reference genome. After that, Socrates formed clusters of these reads based on the association of the original alignment and re-aligned region. Then, split-read clusters are formed based on the original and new alignment regions. These clusters are parsed to find cluster pairs that support a potential rearrangement. The last step was matching short soft clips to support unpaired clusters.

E. *SoftSV*

SoftSV [15] analyzes discordant mapping paired-end reads and all split-reads to find the breakpoint sequences. It analyzes paired-end read alignments to define breakpoint regions. It aligns soft-clipped reads that are within breakpoint regions to each other and builds an undirected graph of soft-clipped reads as vertices. Both reads support the same breakpoint and the soft-clip sequences that are match the other read. Then, it searches the graph for maximal clique for each SV.

F. *Delly*

Delly [16] uses PEM as a main approach for specifying discordant reads then apply SR to refine SV calls. To increase sensitivity, Delly supports analyzing SV from different paired-end sequencing libraries with various insert sizes.

G. *Manta*

Manta [17] constructs break-end association graph of breakpoints regions as nodes. The edges connect regions that have adjacency evidence and self-edges for indels. The edges are analyzed to generate SV candidates. In the last step, Manta assembles the SV *regions.*

III. Proposed SV-BET

This tool support several utilities that can be used to evaluate any SV detection approach as shown in Fig. 1. SV-BET consists of three main components: Benchmark creator, mapper, and evaluator. The different components are illustrated in the following subsections.

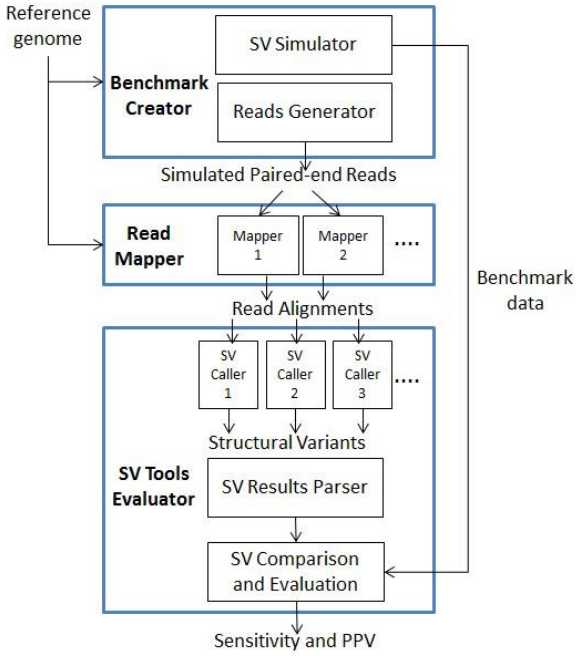

Figure 1. SV-BET main components: Benchmark

A. *Benchmark Creator*

In this part of the tool, a given reference genome is read in FASTA format. The outputs are: Benchmark data marking the SVs, and the simulated paired-end reads for the reference after applying simulated SVs.

The Benchmark data is generated by introducing SVs in a copy of the reference genome based on a user de fined parameters: number of simulated genomes, maximum number of SVs per genome, read length, read depth, standard deviation and error rate. Several SV types and sizes are supported. A SV type can be deletion, insertion, inversion, duplication, and translocation. Duplication events can be either tandem or interspersed. Translocations have two forms intertranslocation and intra-translocation (known as transposition) The SV sizes

are classified into tiny (5-50bp), small (100-200bp), medium (500-1000bp), large (2,000-10,000) and extra-large (20,000-100,000). To prevent the overlapping of SVs, The genome is divided into a non-overlapping parts. The number of SV per genome, SV locations (part number and location in the part), SV types, SV sizes are all chosen randomly. In the benchmark data, the original genome name, length of original genome, number of SVs, length of the simulated genome, SVs types, sizes, and locations are recorded for each simulated genome. This benchmark data will be fed into the evaluator in order to measure the performance of the SV detection tools.

To generate simulated paired-end mapping reads, the wgsim from SAMTools [18] is used. The output is paired-end reads that can be used as input for the Mapper tool.

B. Mapper

In this stage, already known NGS aligners are applied to the reference genome and the simulated reads. We use 3 aligners BWA [9], Bwa-mem [19], and Bowtie 2 [10]. For each SV caller, we use the same alignment tool that was used in its original paper, see Table I.

TABLE I. SPLIT READ-BASED APPROACHES (DEL: DELETION, INS: INSERTION, INV: INVERSION, DUP: DUPLICATION, TRA: TRANSLOCATION, VCF: VARIANT CALL FORMAT [15], TSV: TAB SEPARATED VALUES)

Tool	Input	Read aligner	SV classes	Output
Pindel	BAM	Bwa	DEL, INS, INV, DUP, TRA	VCF
Socrates	BAM	Bowtie2	Breakpoints only	TSF
Delly	BAM	Bwa	DEL, DUP, INV, TRA	VCF
Prism	SAM	Bwa	DEL, INS, INV, DUP	TSF
SoftSearch	BAM	Bwa	DEL, INS, INV, DUP, TRA	VCF
SoftSV	BAM	Bwa-mem	DEL, INV, DUP, TRA	TSF
Manta	BAM, CRAM	Bwa-mem	DEL, INS, INV, DUP, TRA	VCF

C. Evaluator

The evaluator tool takes read alignments and the corresponding reference index as inputs. It then: 1) call SV detection tools, 2) parse their output to a unique format, 3) compare and evaluate with respect to the created benchmark.

The SV detection tools are executed on the read alignments, which are generated by the same aligner used in the original work of each tool. However, configuring SV detection tools and parsing the output of SV calls is often not trivial. Some tools require prerequisite tools, for example Softsv requires BAMTools (https://github.com/pezmaster31/bamtools). Others require configuration files to be set such as Pindel and Manta.

The output of each SV prediction tool is different as shown in Table I. Although, some of SV callers use Variant Call Format (VCF) [20], the way of recording variants are different. Thus, in our tool, the output of the SV callers for each tool are parsed to count the SVs that have been identified correctly and incorrectly into a unique format that allow us to compare with our generated benchmark data. This unique format records reference genome name, SV types and the corresponding breakpoints.

In order to compare and evaluate a given SV tool, two measures are used: the sensitivity and Positive Predictive Value (PPV) as in the equations 1 and 2 where TP is the number of true positives, FN is the number of false negatives and FP is the number of false positives.. The tool evaluates the breakpoint accuracy and the SV accuracy. A threshold is used to define the regions of the breakpoints. This threshold can be tuned according to the required distance. A breakpoint prediction is a true positive if it is at a real breakpoint region without considering the correctness of SV type. PPV and sensitivity are used to evaluate SV type prediction. The SV type is true positive if and only if (1) the region of the predicted variant is intersect with a real variant region; and (2) the SV type is predicted correctly.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \quad (1)$$

$$PPV = \frac{TP}{TP + FP} \quad (2)$$

IV. RESULTS

The Escherichia coli K12 is used as a reference genome, available from the UCSC website (http://microbes.ucsc.edu/cgi-bin/hgGateway?db=eschColi_K12). This genome is chosen because it has a manageable size (about 4.6 Mpb). Two datasets are created as shown in the Table II. The first dataset is used to measure SV breakpoints accuracy and runtime of the SV callers at different read coverage. It has 4 deletions, 0 insertions, 1 inversions, 3 tandem duplications 3 and 2 translocations. The second Dataset is used to compute PPV and sensitivity of SV type predictions. The SV events in dataset 2 are 43 deletions, 63 insertions, 44 inversions, 43 tandem duplications, 57 interspersed duplication, and 52 translocations. The maximum number of SV per genome used in this experiment is 5. In generating paired-end reads, we use 0.01% error rate and standard deviation equals 30. The SV callers are called using default parameters. A 64-bit operating system (Ubuntu 14.04 LTS) with 3.7GiB of memory and Intel Core (i5-4210U CPU, 1.7GHz x4) is used to run the all SV callers, except Manta. Due to the limited memory of this system, Manta was executed on another system with higher CPU and memory.

TABLE II. STATISTICS SETUP FOR GENERATING BENCHMARK USING ESCHERICHIA COLI K12.

Dataset	No. genomes	No. SVs	Coverage	Read Length
1	5	13	[7,15,30]	100
2	100	302	7	100

Dataset 1 is used to evaluate the breakpoints sensitivity Using 3 different coverage values 7, 15 and 30. The SV caller, in our case, is based on split-read analysis that

should be able to detect the SV breakpoints with few deviation from the real breakpoints. Therefore, in this experiment, the threshold was tuned to 6bp. Fig. 2 shows sensitivity results of breakpoints for three coverage values, 7, 15, and 30. The graph illustrates that Pindel and Softsv have the highest sensitivity in detecting breakpoints. The highest sensitivity of Pindel and Softsv are at the read coverage values 15 and 30 respectively. However, there is no significant effect of increasing read coverage on other tools.

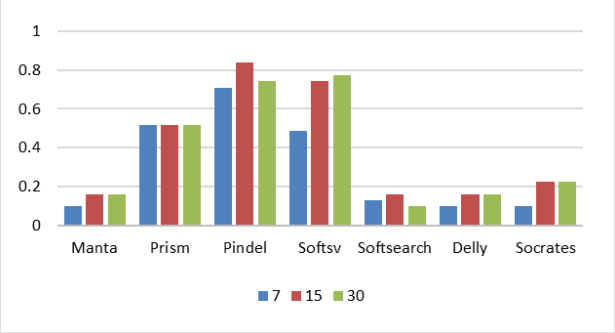

Figure 2. Breakpoints detection sensitivity at coverages (7, 15, and 30) and the maximum distance from real breakpoint is 6bp.

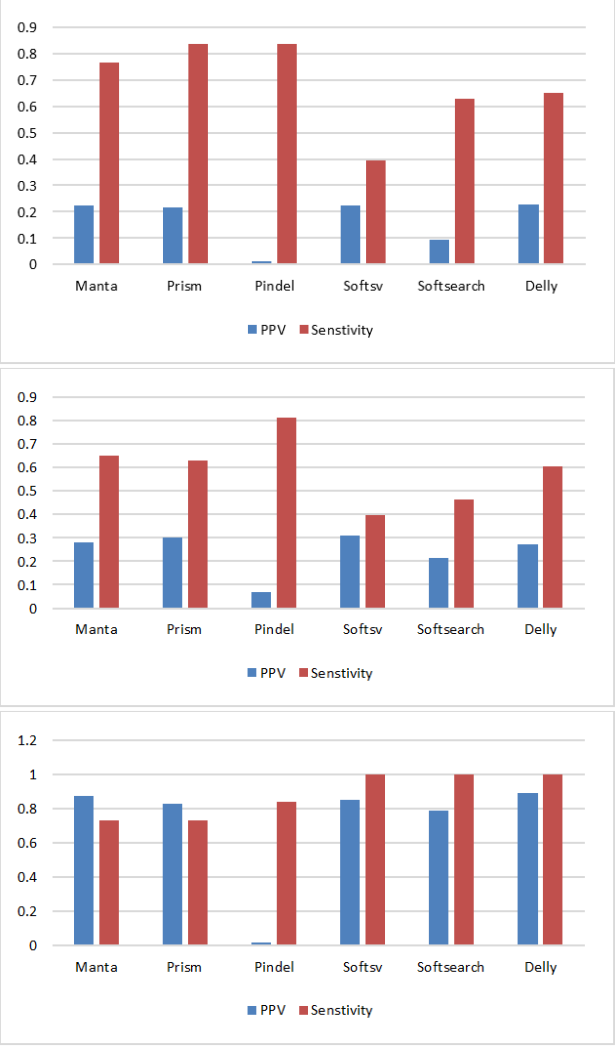

Figure 3. The sensitivity and PPV for deletions (top), duplications (middle) and inversions (bottom).

Dataset 2 was used to evaluate SV callers in detecting three SV types: deletion, inversion and tandem duplication, see Table II. Fig. 3 shows the PPV and sensitivity for each SV type. Prism and manta have the highest sensitivity in detecting deletions. For detecting inversions, Softsv, Softsearch and Delly have 100% sensitivity. Pindel has the highest sensitivity in detecting duplications with the lowest PPV. Fig. 4 summarizes the overall sensitivity of detecting all three events.

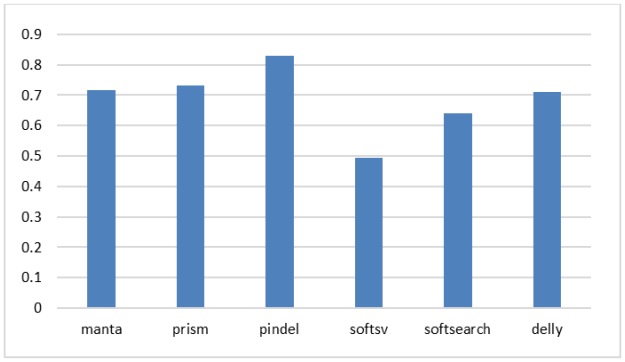

Figure 4. The overall SV callers sensitivity for deletions, inversions, and duplications.

To estimate runtime, we run the SV callers on Dataset 2. Table III shows the runtime of the SV callers for different coverage values, 7, 15, and 30. The runtime of Manta is not recorded here, because of the memory limitation of the system.

TABLE III. RUNTIME IN SECONDS FOR EXECUTING SV CALLERS ON DATASET 1

Tool	Coverage		
	7	15	30
Prism	29.91	63.331	123.195
Pindel	**916.99**	**2015.623**	**60336**
Softsearch	18.236	36.458	68.892
Softsv	**1.638**	**3.212**	**6.249**
Socrates	28.007	35.292	52.822
Delly	11.772	22.818	42.421

V. CONCLUSION

The lack of the gold standard benchmark makes the comparison of SV detection tools tedious. In this paper, SV-BET is proposed as a tool that can be used for creating benchmarks and evaluating different SV callers. It composes of three main components benchmark creator, mapper, and evaluator. We also applied the proposed SV-BET to evaluate and compare seven split- read based SV callers. With using the default settings for SV callers, the initial results show that Pindel is the highest sensitivity in detecting SV breakpoints and predicting SV types. However, Pindel scores the lowest PPV. The current implementation of SV-BET evaluates deletion, inversion and duplication calls for split-read based approaches. SV-BET can be easily extended to evaluate other SV classes and other SV callers based on the SV length.

REFERENCES

[1] C. Alkan, B. P. Coe, and E. E Eichler, "Genome structural variation discovery and genotyping," *Nature Reviews Genetics*, vol. 12, no. 5, pp. 363-376, 2011.

[2] P. Stankiewicz and J. R. Lupski, "Structural variation in the human genome and its role in disease," *Annual Review of Medicine*, vol. 61, pp. 437-455, 2010.

[3] C. Lee and C. C. Morton, "Structural genomic variation and personalized medicine," *The New England Journal of Medicine*, vol. 358, no. 7, p. 740, 2008.

[4] E. Tuzun, *et al.*, "Fine-Scale structural variation of the human genome," *Nature Genetics*, vol. 37, no. 7, pp. 727-732, 2005.

[5] R. E. Mills, *et al.*, "An initial map of insertion and deletion (INDEL) variation in the human genome," *Genome Research*, vol. 16, no. 9, pp. 1182-1190, 2006.

[6] C. Xie and M. T. Tammi, "Cnv-seq, a new method to detect copy number variation using high-throughput sequencing," *BMC Bioinformatics*, vol. 10, no. 1, p. 80, 2009.

[7] J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. M. Jones, and I. Birol, "Abyss: A parallel assembler for short read sequence data," *Genome Research*, vol. 19, no. 6, pp. 1117-1123, 2009.

[8] K. Lin, S. Smit, G. Bonnema, G. Sanchez-Perez, and D. D. Ridder, "Making the difference: Integrating structural variation detection tools," *Briefings in Bioinformatics*, vol. 31, no. 2, 2014.

[9] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows-wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754-1760, 2009.

[10] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with bowtie 2," *Nature Methods*, vol. 9, no. 4, pp. 357-359, 2012.

[11] K. Ye, M. H. Schulz, Q. Long, R. Apweiler, and Z. Ning, "Pindel: A pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads," *Bioinformatics*, vol. 25, no. 21, pp. 2865-2871, 2009.

[12] Y. Jiang, Y. Wang, and M. Brudno, "Prism: Pair-Read informed split-read mapping for base-pair level detection of insertion, deletion and structural variants," *Bioinformatics*, vol. 28, no. 20, pp. 2576-2583, 2012.

[13] S. N. Hart, *et al.*, "Softsearch: Integration of multiple sequence features to identify breakpoints of structural variations," *PloS One*, vol. 8, no. 12, p. e83356, 2013.

[14] J. Schroder, *et al.*, "Socrates: Identification of genomic rearrangements in tumour genomes by re-aligning soft clipped reads," *Bioinformatics*, vol. 30, no. 8, pp. 1064-1072, 2014.

[15] C. Bartenhagen and M. Dugas, "Robust and exact structural variation detection with paired-end and soft-clipped alignments: Softsv compared with eight algorithms," *Briefings in Bioinformatics*, vol. 4226, no. 11, 2015.

[16] T. Rausch, T. Zichner, A. Schlattl, A. M Stutz, V. Benes, and J. O. Korbel, "Delly: Structural variant discovery by integrated paired-end and split-read analysis," *Bioinformatics*, vol. 28, no. 18, pp. i333-i339, 2012.

[17] X. Chen, *et al.*, "Manta: Rapid detection of structural variants and indels for clinical sequencing applications," *Bioinformatics*, pp. 1-2, 2015.

[18] H. Li, *et al.*, "The sequence alignment/map format and samtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078-2079, 2009.

[19] H. Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM," arXiv preprint arXiv:1303.3997, 2013.

[20] P. Danecek, *et al.*, "The variant call format and vcftools," *Bioinformatics*, vol. 27, no. 15, pp. 2156-2158, 2011.

Eman Alzaid is a lecturer in the Department of Computer Science at Imam Muhammad Ibn Saud Islamic University (Saudi Arabia). She received a MSc degree in Advanced computer science from Sheffield University (UK) in 2007. She is currently pursuing the Ph.D. degree at King Saud University (Saudi Arabia). Areas of her research interest include bioinformatics, data mining and software engineering.

Ghada Badr completed Ph.D. in 2006 in Computer Science at Carleton University, School of Computer Science, Ottawa, Canada, in Information Retrieval and Syntactic Pattern Recognition. She was the winner of the Senate Medal for outstanding research achievements in her Ph.D. studies. In 2007, she worked as a research associative in National Research of Canada, Gatineau, Canada. In 2008 and for three years, she worked as a Postdoctoral Fellow in the University of Ottawa, Ottawa, Canada. She is currently an Assistant Professor at the College of Computer and Information Sciences, King Saud University. At KSU, she established the Bioinformatics Research group (BioInG), where she is the coordinator for the group since Fall 2012. Her research interests include bioinformatics, data mining, big data analysis, advanced data structures, information retrieval, and machine learning.