File formats used in next generation sequencing: A literature review

Ángel Canal-Alonso1, Pedro Jiménez1 and Noelia Egido1, Javier Prieto[1], Juan Manuel Corchado[1]

[1]Department of Bioinformatics and Computational Biology, AIR Institute, Carbajosa de la Sagrada, Spain

Email: acanal@air-institute.com

Summary

Next-generation sequencing (NGS) has revolutionized the field of genomics, allowing a detailed and precise look at DNA. As this technology advanced, the need arose for standardized file formats to represent, analyze and store the vast data sets produced. In this article, we review the key file formats used in NGS: FASTA, FASTQ, BED, GFF, and VCF.

The FASTA format, one of the oldest, provides a basic representation of genomic and protein sequences, identifiable by unique headers. FASTQ is essential for NGS, as it stores both the sequence and the associated quality information. BED provides a tabular representation of genomic loci, while GFF details the localization and structure of genomic features in reference sequences. Finally, VCF has emerged as the predominant standard for documenting genetic variants, from simple SNPs to complex structural variants.

The adoption and adaptation of these formats have been fundamental for progress in bioinformatics and genomics. They provide a foundation on which to build sophisticated analyses, from gene discovery and function prediction to the identification of disease-associated variants. With a clear understanding of these formats, researchers and practitioners are better equipped to harness the power and potential of next-generation sequencing.

Keywords: Next-Generation sequencing, File format, Data sharing

Introduction

DNA, deoxyribonucleic acid, is the molecule that carries the genetic information that defines and regulates the characteristics of all living beings. It is composed of a specific sequence of nucleotides, which are the basic units of DNA. Each nucleotide consists of one of four bases: adenine (A), thymine (T), cytosine (C), and guanine (G). The order or sequence in which these bases appear in the DNA molecule is what encodes the genetic information. Determining the exact order of nucleotides in a DNA fragment is known as DNA sequencing.

DNA sequencing has played a fundamental role in biology and medicine since its initial development in the 1970s. For decades, the Sanger chain termination sequencing method was the predominant approach. However, as the demand for sequencing grew, new techniques and technologies emerged that made it possible to sequence more quickly and at a lower cost.

In this context, next generation sequencing (NGS) has emerged in the 21st century as a revolutionary technique. Unlike Sanger sequencing, which analyzes one DNA fragment at a time, NGS allows for massively sequencing millions of DNA fragments in parallel. This “high throughput” capability has opened doors to numerous applications, from

cancer genomics to environmental microbiology and evolutionary genetics.

With the rise of NGS, the need to manage, analyze and interpret the vast volume of data generated has also arisen. This has led to the development of various specific file formats to ensure efficient storage, accurate analysis and sharing of genomic data. In this review, we will take a closer look at these formats, providing a comprehensive overview of their structure, utility, and associated software.

Sequencing technologies in NGS

Since the emergence of NGS, several technologies have emerged that have revolutionized the field of genomic sequencing. Below is a summary of the main technologies used in NGS:

1. Illumina (Sequencing by synthesis):
- Principle: Uses the detection of fluorescent nucleotides incorporated during DNA synthesis.
- Characteristics:
- High accuracy.
- Generates large volumes of short readings.
- Widely used in genomics, transcriptomics and epigenomics.

2. Ion Torrent (Semiconductor Sequencing):
- Principle: Detects protons released during the incorporation of nucleotides in DNA synthesis.
- Characteristics:
- Does not require fluorescence.
- Capable of generating medium length readings.
- Suitable for genotyping and targeted sequencing.

3. PacBio (Single molecule sequencing, in real time):
- Principle: Observe the incorporation of nucleotides in real time into individual molecules.
- Characteristics:
- Generates extremely long readings.
- High error rate, but random errors that can be corrected with adequate coverage.
- Ideal for genome assembly and detection of structural variants.

4. Oxford Nanopore Technologies (ONT):
- Principle: Detects changes in electrical current as a DNA molecule passes through a nanometric pore.
- Characteristics:
- Capable of producing the longest readings of all technologies.
- Portability (for example, the MinION device).
- Applications in field genomics, environmental monitoring and real-time diagnosis.
- Higher error rate compared to other technologies, but the length of the readings and improvements in software help in error correction.

5. 10X Genomics:
- Principle: Combines microfluidics to partition cells or DNA fragments and then applies Illumina sequencing.
- Characteristics:
- Provides information about chromosome phases and structures.
- Used for genomics, single cell level transcriptomics and epigenomic analyses.

The choice of NGS technology to use will depend on the objective of the study, the type of information required and the available budget. Each technology has its own advantages, disadvantages and application niches. What is constant is the rapid evolution and improvement of these technologies, which continue to push the boundaries of what is possible in genomic research.

Information in NGS

Introduction to file formats in NGS

In the next generation sequencing (NGS) environment, the production of genomic data has grown exponentially. These data, characterized by their volume and complexity, require specialized file formats that facilitate their storage, analysis and interpretation. These formats not only serve as information containers, but also establish standards that ensure the coherence, interoperability and reproducibility of the analyzes carried out on different platforms and software.

From the initial stages, where raw sequences and their qualities are stored, to the later phases, where alignments, genomic variants or functional annotations are recorded, there are a variety of formats designed specifically for each type of data and purpose.

Some of these formats have become de facto standards in the scientific community due to their versatility and wide adoption. It is essential for any genomics professional to become familiar with these formats, as they form the foundation on which analyzes and interpretations are built in the modern era of genomics.

In the following sections, we will explore in detail the main file formats used in NGS, providing a comprehensive view of their structure, functionality and applications in the world of genomic sequencing.

FASTA:

The FASTA format, often recognized by its `.fasta` or `.fa` extension, is one of the oldest and most widely used formats in bioinformatics. Although in the current NGS context, the FASTQ format is more commonly associated with raw sequencing data, the FASTA format remains essential in many genomic analyses.

History:

The FASTA format is named after the FASTA program, which was developed in the 1980s by David J. Lipman and William R. Pearson. Originally, this program was designed to perform protein sequence searches in databases. As the program gained popularity for sequence alignment and database searching, the associated file format, which was simple and efficient for representing nucleotide or amino acid sequences, also became popular.

The simple structure of the FASTA format was one of the main reasons for its wide adoption. A sequence in FASTA format begins with a description line, preceded by the ">" symbol, followed by lines representing the sequence itself. This simplicity allowed the format to be easily readable by both humans and computers.

With the expansion of genomics and bioinformatics in the following decades, the FASTA format became the de facto standard for representing DNA, RNA, and protein sequences. Larger genomic databases, such as GenBank, EMBL, and the Protein Data Bank, adopted the FASTA format for sequence distribution and searching, further solidifying their position in the scientific community.

Although the sequencing landscape has evolved greatly since the days of the original FASTA program, and new formats such as FASTQ have emerged to meet the specific needs of NGS, the FASTA format remains essential. It is used to represent reference genomes, individual gene sequences, protein assemblies, and much more. Its legacy and relevance endures in the modern era of genomics, underscoring the importance of simple and effective solutions in science.

The FASTA format is appreciated for its simplicity and clarity. Although its basic structure has remained relatively constant over time, it has proven to be flexible and adaptable to various applications in genomics and proteomics.

Basic structure:

1. Header Line: Each entry in a FASTA file begins with a header line, distinguished by the ">" symbol at the beginning. This header line provides a description or identification of the stream. This line often contains a unique identifier for the sequence, and may include additional information such as the source of the organism, database accession number, and so on.

Example:

```
>NM_001301717.2 Homo sapiens actin alpha cardiac muscle 1 (ACTC1), transcript variant 2, mRNA
```

2. Sequence: After the header line, the sequence itself follows, written in consecutive lines. In the case of DNA or RNA sequences, the sequence will be composed of the letters that represent the nucleotides (A, T, C, G, U). For protein sequences, letters representing amino acids are used.

Example:

```
ATGGAGACAGAAGTCTTCACTGCTGAGGAGGAG
GAAGAGGAAGCAGATGGAGAAGAAGCTG
TCAACATCAAGTCTGACCTAATCACTGAGAAGCT
CGGAAGGAAACTGACCGAAGGCAAGA
```

Additional considerations:

- A FASTA file can contain multiple sequences. Each stream will be demarcated by its own header line followed by its stream.
- The length of each line in the sequence is usually restricted to a specific number of characters (for example, 60 or 80) for readability reasons, although this is not a hard and fast rule.
- There is no restriction on the total number of characters in the sequence, so a FASTA file can represent anything from small DNA fragments to entire genomes.
- Although the basic structure of the FASTA format is simple, the specific conventions for the header line may vary depending on the database or resource from which the file comes.

In summary, the FASTA format provides a simple and clear representation of biological sequences. Its intuitive structure has made it easy to be widely adopted and used in various applications, from homology searches to molecular evolution studies. Although it lacks the ability to store quality information required in modern NGS (something the FASTQ format provides), FASTA remains a mainstay in bioinformatics and genomics.

FASTQ:

History

As bioinformatics and genomics advanced in the first decade of the 21st century, so did sequencing technologies. The advent of next-generation sequencing (NGS) marked a turning point, not only in terms of the quantity of data produced but also in the quality and resolution of that data. This change in the technological landscape required an evolution in data formats. It is in this context where the FASTQ format was born and consolidated.

The FASTQ format was initially created to accommodate the demands of early next-generation sequencing platforms, such as Illumina's Solexa technology. These new methods generated, for the first time, individual reads with quality associations for each base sequenced, something that previous techniques, such as Sanger sequencing, did not produce on the same scale.

The name "FASTQ" is derived from the combination of "FASTA", reflecting its structural similarity to the original sequence format, and "phred quality scores", which are the ratings used to represent the quality of the bases. These quality scores, originally developed for Sanger sequencing in the 1990s, were adapted and expanded for use in NGS.

Although the FASTQ format began as a specific solution for Solexa technologies, it was quickly adopted and adapted by other NGS platforms. However, this rapid growth led to certain inconsistencies, especially in the coding of quality scores. For example, while the Solexa platform used a different quality encoding range, later versions of Illumina and other technologies adopted the Phred +33 standard, which is the same system used in Sanger sequencing.

This variability in coding quality and other particularities, while reflecting the rapid evolution and competition among sequencing technologies, also presented challenges for bioinformaticians and software developers. The need to standardize and clarify the format became evident. Fortunately, with time and community effort, a broader consensus has been reached on how the FASTQ format should be used, although it is crucial that users are aware of the specific version and quirks associated with the sequencing platform. that they are using.

From humble beginnings adapting to a new era of sequencing technology, the FASTQ format has grown to become a mainstay in the world of NGS. Their story reflects the dynamic and often challenging nature of the fast-moving field of genomics, and underscores the importance of adaptability and collaboration in science.

Operation and Structure

The FASTQ format is essential in the context of next generation sequencing (NGS). Unlike the FASTA format, which stores only sequences, the FASTQ includes information about the quality of each sequenced base. This quality information is crucial to determine the reliability of the reads and to perform subsequent NGS analyses, such as alignment and variant identification.

Basic structure:

The FASTQ file consists of four lines per record:

1. Header line: It begins with the “@” symbol and usually contains information about the sequencing instrument, the run number, the unique identifier of the reading, among others.

Example:
```
@SEQ_ID MISEQ:7:000000000-A4K08:1:1101:12469:2217 1:N:0:
```

2. Sequence: The next line presents the nucleotide sequence itself, just like in a FASTA file.

Example:
```
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
```

3. Separator: A line that only contains the “+” symbol. Occasionally, this line may repeat information from the header line, although in most cases it simply presents the “+” symbol on its own.

Example:
```
+
```

4. Qualities: This line encodes the quality of each base in the sequence using ASCII characters. Each character

represents a quality value, which indicates the probability that a particular base has been sequenced incorrectly. NGS analysis programs use these values to weight the reliability of the readings.

Example:
''' !"*((((***+))%%%++)(%%%%).1***-+*"))**55CCF>>>>>> CCCCCCC65 ```

- The FASTQ format allows sequencing reads of different lengths to be represented. Therefore, not all sequences in a FASTQ file will be the same length.
- The quality coding system has varied between versions and sequencing platforms. The most common versions are ASCII code 33 (Sanger) and ASCII code 64 (Illumina 1.3+), and it is vital to know which one is being used to correctly interpret the quality values.
- As with FASTA, a FASTQ file can contain many records, that is, multiple groups of these four lines.
- FASTQ files can be very large, especially in high coverage sequencing studies, so they are often compressed using tools such as gzip.

In summary, the FASTQ format is essential in the era of next-generation sequencing. By providing both sequence and quality information, FASTQ allows researchers and bioinformaticians to evaluate and analyze sequencing data at an unprecedented level of detail. Its structure, although a little more complex than that of FASTA, has become a standard in the field of NGS.

BED:

In the context of genomics and bioinformatics, the BED (Browser Extensible Data) format has been established as a standard for representing regions of interest in genomes and other sequence-related data sets. The simplicity, versatility and extensible capacity of the BED format has made it indispensable in various applications.

Origins of the BED:

The BED format was initially developed by the UCSC Genome Browser team, a widely used online tool for visualizing genomic annotations and features in reference genomic contexts. Since one of the main goals of the Genome Browser is to allow users to explore and visualize specific genomic regions along with a variety of associated data and annotations, there was an obvious need for a format that could efficiently represent these regions and annotations in the genome.

The UCSC team developed the BED format to make it easier to load and display custom data in the browser. In doing so, they created a structure that was not only suitable for their own needs, but also well suited to many other uses in genomics.

Expansion and adoption in the community:

What sets the BED format apart is its simple but highly extensible structure. In its most basic form, a BED file needs only three columns: chromosome, start position, and end position. However, the format can be extended to include additional information in additional columns, such as item names, scores, and string direction, among others.

Given this flexibility, it is not surprising that the BED format has been quickly adopted by the broader genomics community. It has been used in a variety of applications, from the identification of regulatory regions to the annotation of genomic variants. Additionally, many bioinformatics software and tools, such as BEDTools, have been developed specifically to work with BED files, further reinforcing their position as a standard in genomics.

The BED format is an example of how a solution developed for a specific need can, over time, become an industry standard. Since its creation in the UCSC Genome Browser, it has evolved and adapted to a variety of applications in genomics, and its legacy underscores the importance of simplicity and extensibility in data format design.

BED: Structure and Functioning

The BED (Browser Extensible Data) format is a simple and flexible way to represent regions in a genome. Despite its simplicity, it is powerful and widely used in bioinformatics. The basic structure and operation of the BED format is described below.

Basic structure:

A BED file consists of a series of lines, each describing a particular region of the genome. Columns in a BED line are separated by tabs. The number of columns may vary, but the first three are essential and must always be present:

1. chromosome (chrom): The name of the chromosome or reference sequence. For example: `chr1`, `chr2`, `chrX`, etc.

2. start position (chromStart): The start position of the region on the chromosome. It is important to note that BED uses zero-based indexing, meaning that the numbering starts from 0.

3. end position (chromEnd): The final position of the region on the chromosome. Unlike start, this position is exclusive, that is, the base pointed to by chromEnd is not included in the BED region.

These three fields are required in any BED file. However, there are nine additional optional fields that may be present, extending the format:

4. name: The name of the BED region.
5. score: A score between 0 and 1000. Can be used to store any numeric value associated with the region.
6. strand: The strand of the genome. It can be '+' or '-'.
7. thickStart and thickEnd: These fields are useful for representing coding regions in genes.
8. itemRgb: Color to display the item.
9. blockCount: Number of blocks (exons, for example) in the region.
10. blockSizes: Size of the blocks.
11. blockStarts: Block start positions.

Typical operation and applications:

The BED format is primarily used to define and manipulate sets of genomic coordinates. It is especially useful for:

- View genomic regions in genome browsers, such as the UCSC Genome Browser.
- Annotate regions of interest, such as protein binding sites or differentially methylated regions.
- Manipulate and analyze sets of coordinates, such as intersections, unions and complements, especially using tools such as BEDTools.
- Define coding or non-coding regions in genes.

Example:

Assuming we want to represent a gene located on chromosome 1, starting at base 100 and ending at base 500, with the name "MiGen", a score of 960, on the positive strand, might look like this:

```
chr1  99    500    MiGen  960    +
```

The BED format is an essential tool in genomics and bioinformatics due to its simplicity and versatility. Although its basic structure is simple, its extensible design allows it to represent a wide variety of genomic information, making it a popular choice for many applications in the field. It is crucial to understand zero-based indexing and other format-specific details to ensure correct use and accurate interpretation of data.

GFF

The General Feature Format (GFF) has been an essential tool in genomics and bioinformatics for the past decades. It provides a mechanism for representing gene structure and other annotations in genomes. Let's see how the GFF has evolved and how it has served the scientific community since its inception.

Origins of the GFF:

The first version of GFF was developed within the scope of the WormBase project, a database dedicated to the annotation and genomic analysis of the nematode Caenorhabditis elegans, one of the first organisms to be completely sequenced. With the rise of sequencing projects in the 1990s, the need for a standardized format to describe the location and structure of genes, exons, introns and other relevant genomic features in reference sequences was evident.

Evolution and adaptations:

The original GFF format, now known as GFF1, was simple and met basic needs. However, as genomics advanced and more genomic features and annotation types were identified, the format needed to be adapted and expanded.

This drive led to the development of GFF2, which introduced changes and improvements, but still had limitations in representing complex relationships between traits, such as hierarchies in gene structures.

Therefore, version 3, GFF3, was released to address these and other issues. GFF3 introduced a richer structure, allowing relationships between features to be represented using identifiers and references. Additionally, stricter rules were incorporated to ensure uniformity and consistency in annotations.

Adoption and uses in the community:

Since its introduction, the GFF format has been widely adopted in genomics. Many genomic databases and genome browsers, such as UCSC and Ensembl, have offered the ability to download or upload data in GFF format. It has also been

essential for gene annotation tools and genomic analysis pipelines.

Over time, variants and related formats, such as GTF (Gene Transfer Format), emerged to meet the specific needs of certain projects or platforms. Although these derived formats share many similarities with GFF, they also possess key differences in their structure and specifications.

The history of the GFF is a testament to the dynamism and adaptability of the genomics community. As science advanced, the format evolved in response to the increasing demands and complexities of the field. Today, the GFF and its variants remain indispensable tools, reflecting their value and relevance in a field that continues to evolve rapidly.

GFF: Operation and Structure

The General Feature Format (GFF) is a standard used to describe the location and structure of genomic features in reference sequences. In its most recent version, GFF3, this format has been designed to provide a detailed and extensible representation of genomic annotations. Next, we explore its structure and operation.

Basic structure:

The GFF is a text-based format with tab-delimited columns. Each line in the file represents a genomic feature, such as a gene, exon, intron, etc. The columns in GFF are:

1. seqid: The identifier of the reference sequence (for example, a chromosome or contig) where the feature is located.
2. source: The source or program that generated this feature. This can be a gene prediction program, an annotation database, etc.
3. type: The type of the feature (e.g. gene, mRNA, exon, CDS, etc.).
4. start: Start position of the feature in the reference sequence.
5. end: End position of the feature.
6. score: A score associated with the characteristic. If there is no punctuation, a period ('.') is used as a placeholder.
7. strand: The strand on which the feature is located. It can be '+', '-' or '.' (if the thread is unknown or not applicable).
8. phase: For "CDS" type features, indicates where the next full encoding begins. It can be '0', '1', '2' or '.' (if not applicable).
9. attributes: A list of key-value pairs separated by semicolons. These attributes can include unique identifiers, relationships to other features (such as an exon associated with a specific mRNA), and other metadata.

Example of a GFF3 entry:

```
chr1 . gene 1000 5000 . + . ID=gene0001;Name=my_gene
chr1 . mRNA 1000 5000 . + . ID=mRNA0001;Parent=gene0001;Name=my_transcript
chr1 . exon 1000 1500 . + . ID=exon0001;Parent=mRNA0001
chr1 . exon 2000 2500 . + . ID=exon0002;Parent=mRNA0001
chr1 . CDS 1200 1500 . + 0 ID=cds0001;Parent=mRNA0001
```

Functioning:

- The GFF allows representing hierarchical genomic annotations. For example, a gene may have multiple transcripts (mRNAs), and each transcript may have multiple exons and coding regions (CDS). These relationships are represented using the "ID" and "Parent" attributes in the attributes column.

- It is common for GFF3 files to be accompanied by FASTA files containing the sequences of the described characteristics. This may be at the end of the GFF3 file, separated by a "##FASTA" line.

The GFF3 format, being versatile and detailed, is essential to represent the complexity inherent in genomic annotations. Although it may require some learning curve to understand and use efficiently, its framework is designed to accurately capture genomic relationships and features in the context of reference sequences. It is crucial for many analytical processes and bioinformatics tools and is a cornerstone in modern genomics.

VCF

The VCF (Variant Call Format) format has revolutionized the way we represent and analyze genetic variants in high-depth sequencing. Since its creation, it has been a key standard for documenting variants such as SNPs, indels and more complex structural variants. Let's look at how VCF emerged and evolved in response to the emerging needs of genomics.

With the onset of the next-generation sequencing (NGS) era in the mid-2000s, the scientific community began generating large volumes of sequence data. As more human

and other species genomes were sequenced, it became evident that there was massive genetic diversity among individuals. These variations could range from simple single nucleotide changes (SNPs) to deletions, insertions or chromosomal rearrangements. A standardized format was necessary to document these variants in a coherent and structured way.

Birth of the VCF:

The VCF format was first introduced around 2008, primarily by the 1000 Genomes Project, an international consortium that aimed to sequence the genomes of a wide range of individuals to catalog human genetic variation. VCF was designed to be flexible, allowing representation of variants across different genomes and reference assemblies.

Evolution and improvements:

The original VCF (VCFv1) was functional but limited in its ability to represent the increasing complexity of genomic data. This led to the development of VCFv2 and later VCFv3. Finally, VCFv4 (and its subversions) incorporated additional features, including the ability to represent more complex structural variants and detailed metadata on variant calls.

A key improvement in later versions of VCF was the introduction of INFO and FORMAT fields, which provide additional metadata about variants and allow greater customization in the description of genetic variants.

Adoption and uses in the community:

Since its introduction, the VCF format has been widely adopted in the genomics community and has become the de facto standard for genetic variant representation. It has been essential for large genomic projects, variant databases and bioinformatics tools.

The history of the VCF reflects the rapid evolution and adaptability of the genomic community. It was developed in response to an urgent need and has continued to adapt to the changing demands of the field. Today, VCF remains a fundamental tool in genomics, evidence of its robustness and versatility. Its development and mass adoption is testament to the collaboration and collective vision of the scientific community to address emerging challenges in genomics.

VCF (Variant Call Format): Operation and Structure

The VCF is a text format used to describe genetic variants in the context of reference sequences. It is structured to be clear and readable, but at the same time provides a detailed level of information about the variants detected. Let's break down the structure and operation of the VCF:

Header:

The VCF file begins with a header, which allows users and programs to properly interpret the contents of the file. The header lines begin with "##" and provide metadata about the file. This metadata can include versions of the VCF format, information about the programs or pipelines used to generate the variants, INFO and FORMAT field definitions, and more. There is also a header line that specifies the columns in the file, starting with "#CHROM".

Main columns:

1. CHROM: The name of the chromosome or the reference sequence.
2. POS: The starting position of the variant on the chromosome or reference sequence.
3. ID: Identifier of the variant, usually from databases such as dbSNP.
4. REF: The reference allele at that position.
5. ALT: The alternative allele(s) observed for the variant.
6. QUAL: A value that represents the quality of the variant call.
7. FILTER: Indicates if the variant has passed the established filters. If so, "PASS" is usually displayed.
8. INFO: Contains additional metadata about the variant, defined in the header.
9. FORMAT (optional): Specifies the format of the individual data in the following columns.
10. Sample columns (if present): Provide sample-specific information about the variant, such as genotypes, read depth, etc.

Example of a VCF entry:

```
#CHROM  POS     ID      REF     ALT     QUAL
        FILTER INFO     FORMAT          Sample1
        Sample2
chr1    12345   rs1234  TO      g       99      PASS
        AF=0.5  GT:DP   0/1:20  1/1:18
```

Functioning:

- Each line after the header represents a genetic variant at a specific position. It can be a SNP, an indel, or even a structural variant, depending on the REF and ALT entries and the information in the INFO column.

- The INFO column provides additional details about the variant, such as allele frequency (AF), functional impacts, evidence of disease association, among others. The fields within INFO are defined in the header and may vary depending on the tool or database.

- Sample columns, when present, provide genotypic information and other data related to individual samples. For example, "0/1" could indicate a heterozygous genotype, while "1/1" indicates a homozygous for the alternative allele.

Final considerations:

The VCF is a powerful tool that encapsulates a wide range of genetic information in a standardized format. It is fundamental to many aspects of genomics, from basic research to clinical applications. Its modular and extensible design has made it suitable to handle the increasing complexity and volume of data in modern genomics. By understanding its structure and functioning, researchers and clinicians can effectively extract, compare, and analyze the rich information contained in its inputs.

References

Garcia-Retuerta D, Canal-Alonso A, Casado-Vara R, Rey AM, Panuccio G, Corchado JM. Bidirectional-Pass Algorithm for Interictal Event Detection. In Practical Applications of Computational Biology & Bioinformatics, 14th International Conference (PACBB 2020). PACBB 2020. Advances in Intelligent Systems and Computing, vol 1240. Springer, Cham. https://doi.org/10.1007/978-3-030-54568-0_20

Castillo Ossa LF, Chamoso P, Arango-López J, Pinto-Santos F, Isaza GA, Santa-Cruz-González C, Ceballos-Marquez A, Hernández G, Corchado JM. A Hybrid Model for COVID-19 Monitoring and Prediction. Electronics. 2021; 10(7):799.
https://doi.org/10.3390/electronics10070799

Intelligent Platform Based on Smart PPE for Safety in Workplaces. Márquez-Sánchez S, Campero-Jurado I, Herrera-Santos J, Rodríguez S, Corchado JM. Sensors (Basel). 2021 Jul 7;21(14):4652
https://doi.org/10.3390/s21144652

A. Canal-Alonso, R. Casado-Vara and J. Manuel Corchado, "An affordable implantable VNS for use in animal research," 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2020, pp. 1-4,
doi: 10.1109/ICECS49266.2020.9294958

An Agent-Based Clustering Approach for Gene Selection in Gene Expression Microarray. Ramos J, Castellanos-Garzón JA, González-Briones A, de Paz JF, Corchado JM. Interdiscip Sci. 2017 Mar;9(1):1-13
DOI 10.1007/s12539-017-0219-6

Thanks

This study has been funded by the AIR Genomics project (with file number CCTT3/20/SA/0003), through the call 2020 R&D PROJECTS ORIENTED TO THE EXCELLENCE AND COMPETITIVE IMPROVEMENT OF THE CCTT by the Institute of Business Competitiveness of Castilla y León and FEDER funds