Biological Sequence Analysis

Usman Saeed[1,2] • Zainab Usman[2]

[1]Dennemeyer Octimine GmbH, München, Germany; [2]Department of Bioinformatics, Technical University Munich, Wissenschaftzentrum Weihenstephan, Freising, Germany

Author for correspondence: Usman Saeed, Dennemeyer Octimine GmbH, Landaubogen 1-3, 81373 München, Germany. Email: usman.saeed08@gmail.com

Doi: http://dx.doi.org/10.15586/computationalbiology.2019.ch4

Abstract: This chapter focuses on several biological sequence analysis techniques used in computational biology and bioinformatics. The first section provides an overview of biological sequences (nucleic acids and proteins). Bioinformatics helps us understand complex biological problems by investigating similarities and differences that exist at sequence levels in poly-nucleic acids or proteins. Alignment algorithms such as dynamic programming, basic local alignment search tool and HHblits are discussed. Artificial intelligence and machine learning methods have been used successfully in analyzing sequence data and have played an important role in elucidating many biological functions, such as protein functional classification, active site recognition, protein structural features identification, and disease prediction outcomes. This chapter discusses both supervised and unsupervised learning, neural networks, and hidden Markov models. Sequence analysis is incomplete without discussing next-generation sequencing (NGS) data. Deep sequencing is highly important due to its ability to address an increasingly diverse range of biological problems such as the ones encountered in therapeutics. A complete NGS workflow to generate a consensus sequence and haplotypes is discussed.

Keywords: dynamic programming; machine learning; next-generation sequencing; pairwise alignment; sequence analysis.

In: *Computational Biology*. Holger Husi (Editor), Codon Publications, Brisbane, Australia.
ISBN: 978-0-9944381-9-5; Doi: http://dx.doi.org/10.15586/computationalbiology.2019

INTRODUCTION

It has been estimated that over 12 million different species exist on the planet (1). The biodiversity across all life forms including plants, animals, and microbes can be attributed to their unique genomic and proteomic composition. Like an instruction manual that guides about all the sequential tasks to be done in the right order to accomplish a process, the biological organisms have all the details in their genes, creating combinations of nucleotides resulting in the diversity that we see in the biological world. There are two types of nucleic acids, DNA (deoxyribonucleic acid) and RNA (ribonucleic acid). In 1953, Watson and Crick proposed that the DNA is made up of two long poly-nucleotide chains comprising of four nucleotides, namely adenine (A), guanine (G), cytosine (C), and thymine (T) (2). In RNA, however, thymine is replaced by the nucleotide uracil (U) as a complementary nucleotide to adenine. The strands in both DNA and RNA have a polyphosphate backbone with adjacent nucleotides forming polyphosphate di-ester bonds. DNA is a double-stranded structure; the two chains are twisted around each other with hydrogen bonds between the base portions of nucleotides holding the two chains together. The sequence of bases in DNA is of crucial importance as it contains the code to the formation of diverse proteins and hence the complexity and diversity of life. The unique order of bases in DNA results in the creation of basic hereditary units called genes. In 2003, the human genome project initially estimated 20,000 genes in the human genome (3, 4), and these estimates were later revised to 25,000–30,000 genes (5). Based on the sequence of DNA, enzymes like RNA polymerase create single-stranded messenger RNA (mRNA) that later translate into proteins. This whole process of decoding the DNA sequence into a protein is referred to as the "central dogma of life" (6). Depending on different organisms, all genes may not code for proteins. Composed of amino acids, proteins are much more complicated than nucleic acids. There are 20 major amino acids which make up proteins, and each protein can have them assembled in different numbers and order. Amino acid sequence of proteins is also of crucial importance as it not only determines the physiochemical properties of proteins but also determines the different conformations they can create in a three-dimensional space (7). These conformational changes result in complicated protein structures that in turn allows them to serve unique biological functions, for example, transport, functional regulation, and homeostasis. Therefore, the importance of nucleotide sequence in DNA/RNA and of amino acids in proteins cannot be overstated.

Sequence comparison of DNA can allow us to compare the differences at gene level across different organisms and species. Comparative genomics is a branch of science that uses bioinformatics techniques extensively to trace the genes across multiple species and study their similarities and differences. Such studies help us infer the functional and structural characteristics of newly found or existing proteins. Programmatically, biological sequence analysis is not much different than comparing strings and text, and thus, developing the concept of alignment is important. Sequences evolving over species and clades through mutations include insertions, deletions (indels), and mismatches. When comparing two biological sequences, an alignment is generated to view differences between the sequences at each position.

PAIRWISE ALIGNMENT AND DYNAMIC PROGRAMMING

Pairwise alignment involves comparing two sequences against each other and finding the best possible alignment between them. The process involves scoring at each position for match, mismatch, and indels. Since matches are preferred over deletions, matches are normally assigned the highest scores, and lowest for insertions. Similarity between two sequences is inversely proportional to the number of mismatches and indels in their alignment. Although the scoring for alignment can be as simple as +1 for match, 0 for mismatch, and –2 for insertion, different scoring models have been developed based on the statistically relevant frequencies of one amino acid changing into another.

Needleman–Wunsch algorithm

Initially developed by Needleman and Wunsch in 1970, the algorithm is based on dynamic programming and allows for global or end-to-end alignment of two sequences (8). The algorithm involves three main steps, namely initialization, calculation, and trace back. A matrix of dimensions *i, j* is initialized, where *i* and *j* are the length of two sequences under comparison. In the second step, *F*(*i, j*) highest score for each comparison at each position is calculated,

$$F(i, j) = max\ \{F(i-1, j-1) + s(x_i, y_i), F(i-1, j) - d, F(i, j-1) - d\}$$

where "$s(x_i, y_i)$" is the match/mismatch score and "*d*" is the penalty for deletion.

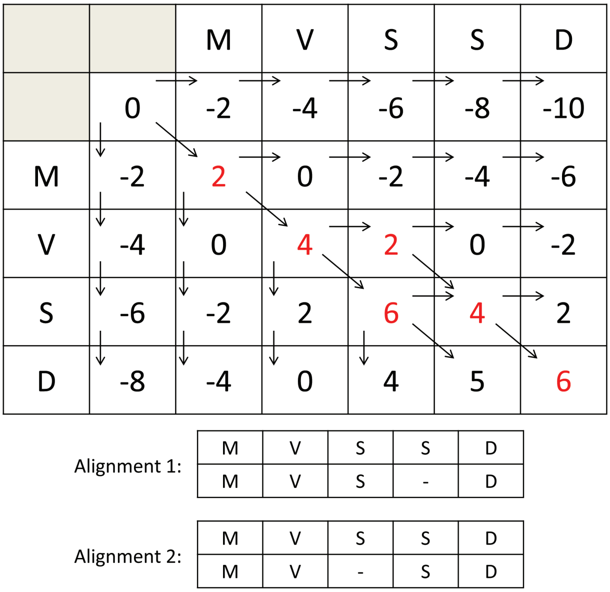
After the maximum score for each position in the matrix is calculated (Figure 1), trace back starts from the last cell (bottom right) in the matrix. Each step involves moving from the current cell to the one from which the value of the current cell was derived. A match or mismatch is assigned if the maximum score was derived from a diagonal cell. Insertion/deletion is assigned if the score was derived from the top or left cell. After the trace back is completed, we have two sequences aligned end to end with each other with an optimal alignment score (9).

Smith–Waterman algorithm

Initially proposed by Smith and Waterman in 1981, the algorithm allows for local sequence alignment and is like the Needleman–Wunsch algorithm (10). Local sequence alignment can be used in situations where it is required to align smaller subsequences of two sequences. In the biological context, such a situation may arise while searching for a domain or motif within larger sequences. The algorithm comprises of the same steps as Needleman–Wunsch; however, there are two main differences. Computation of max score also includes an option of 0:

$$F(i, j) = max\ \{0, F(i-1, j-1) + s(x_i, y_i), F(i-1, j) - d, F(i, j-1) - d\}$$

Assignment of "0" as max score corresponds to starting a new alignment. It allows for alignments to end anywhere within the matrix. The trace back therefore starts from the highest value of *F*(*i, j*) in the matrix and ends where it encounters 0.



		M	V	S	S	D
	0	-2	-4	-6	-8	-10
M	-2	2	0	-2	-4	-6
V	-4	0	4	2	0	-2
S	-6	-2	2	6	4	2
D	-8	-4	0	4	5	6

Alignment 1:

M	V	S	S	D
M	V	S	-	D

Alignment 2:

M	V	S	S	D
M	V	-	S	D

Figure 1 Needleman–Wunsch matrix. The calculation uses scores for match +2, mismatch –1, and gap –2. The arrows show the matrix cell from where the value is generated. Red-coloured cell values show the trace back that creates alignment.

HEURISTIC LOCAL ALIGNMENT

One main challenge in bioinformatics sequence analysis is decoding the vast number and length of sequences. These big data of protein and DNA sequence databases (over 100 million sequences) come from species across the tree of life. Although the local alignment methods based on dynamic programming are quite accurate and guarantee to find an optimally scored alignment, they are slow and not practical for sequence alignments against databases with millions of sequences. The time complexity of dynamic programming algorithms is $O(mn)$, that is, the product of sequence lengths. In the initial attempts to improve the speed for sequence comparisons, heuristic algorithms like BLAST (11), BLAT (12), and FASTA (13, 14) were created. Further advancements in the efficiency of similarity search algorithms came with algorithms like LSCluster (15), Usearch (16), Vsearch (17), Diamond (18) and Ghostx (19). In general, these algorithms search for exact matches and extend the alignment from those matches trying to estimate the optimal scoring alignment.

Basic Local Alignment Search Tool, initially developed by Altschul and colleagues (11), is based on the idea that the best scoring sequence alignment would contain the highest number of identical matches or highly scoring sub-alignments. The algorithm carries out the following steps: (i) reduce the query sequence into small subsequences called seeds, (ii) search these seeds across the database for exact matches, and (iii) extend the exact matches into an un-gapped alignment until a maximal scoring extension is reached. The use of seeds to first search for exact matches greatly increases the whole searching process and the un-gapped alignment misses only a small set of significant matches. The accuracy and sensitivity of BLAST made it amongst the most widely used search algorithm in the biological world. A variant of BLAST named Position-Specific-Iterative BLAST (PSI-BLAST) extends the basic BLAST algorithm (20). PSI-BLAST performs multiple iterations of BLAST and uses the hits found in one iteration as a query for the next iteration. Although slower due to sheer amount of calculations required, PSI-BLAST is considered a reliable tool to find distant homology relationships.

Although BLAST and PSI-BLAST are extensively used, recently developed methods offer results with higher accuracy and sensitivity. Hidden Markov models (HMM) have been used efficiently for numerous applications to understand and explore biological data. One such example is HMM–HMM-based lightning fast sequence search (HHblits) introduced in 2012 (21). The tool can be used as an alternative for BLAST and PSI-BLAST and is 50 to 100 times more sensitive. The high sensitivity of the tool can be attributed to the algorithm which relies on comparing the HMM representations of the sequences. Although profile–profile or HMM–HMM alignments are very slow to compute, the prefilter in HHblits reduces the required alignments from millions to thousands, thus giving it a considerable speed advantage. HHblits represents each sequence in the database as a profile HMM. Prefiltering reduces the number of HMM comparisons for similarity search by selecting only those target sequences where the largest un-gapped alignment exists, and a Smith–Waterman based alignment reveals a significant E-value.

MACHINE LEARNING AND SEQUENCE ANALYSIS

Biological data provide amongst the perfect use cases of machine learning and artificial intelligence algorithms. This is the reason that researchers in the field of bioinformatics and computational biology have used statistical analysis and inference since the very beginning. Techniques like maximum likelihood (22) and neighbor joining (23) have been used for comparative genomics. Naïve Bayes and Markov chains have been extensively used for sequence analysis. Logistic regressions, support vector machines, and random forests have been used in numerous applications ranging from prediction of protein sequence or structural elements to classification of proteins into different structural and functional classes. With the development of deep neural networks, we observe an increase in the use of the algorithms like long short-term memory (LSTM) (24) and convolutional neural networks (CNN or ConvNet) (25) to predict the different features and behavior of proteins, for example, protein contact prediction and prediction of post-translational modifications.

Machine learning methods are broadly divided into two types, supervised and un-supervised learning. Based on the inherent features of the data, if it is not labeled and cannot be assigned to any type, then classification is done using unsupervised learning. For instance, the classification of proteins into different groups is done based on their sequence similarity to each other. K-means clustering algorithm (26) and Markov clustering (27) can be used in unsupervised classification. On the other hand, if the data are labeled into different sets, this information can be used to train the computer by showing it positive and negative examples.

Once the training is complete, the accuracy of training can be tested using similar data not used in the training dataset. Any classification technique following training and testing procedures using labeled data is termed supervised machine learning. Examples for this type of learning include SVM, HMM, random forest, and CNN.

Hidden Markov Models

HMM is a statistical method that can be used to predict the probability of occurrence for a future event. HMMs provide the foundations for a range of complex models that can be used for multiple sequence alignment, profile searches or detection of sequence elements. In order to understand the HMMs and their use in biological data, consider the example of binding site recognition on a DNA sequence. There is an observable sequence of nucleotides which in the right order hides underneath a binding site. We can observe the nucleotide sequence, but the presence or absence of a binding site remains hidden to us. HMMs are particularly suited for such problems because they use observed frequencies to calculate emission and transition probabilities to decipher the hidden states. An HMM involves two types of probabilities, transition and emission probabilities. The probability of moving from one state to another is called the transition probability. The probability to observe a variable within a state is called emission or output probability.

Figure 2 shows a schematic HMM with basic architecture and elements. HMMs have been used not only to create sequence profiles but also to create probabilistic model representation of protein clusters. Pfam is an example database that clusters proteins based on their functional elements and represents them with HMM. The downside to HMMs is that they assume a future event depends only on the event that happened immediately before and not in the distant past. This creates a limitation to use standard HMMs in complex cases where sequence elements influence each other that may be close in the three-dimensional space but sequentially lie far from each other. Outside of the biological world, one such example is autocomplete or word suggestions. The words appearing in suggestion are directly dependent on the word that appeared immediately before the present suggestion.

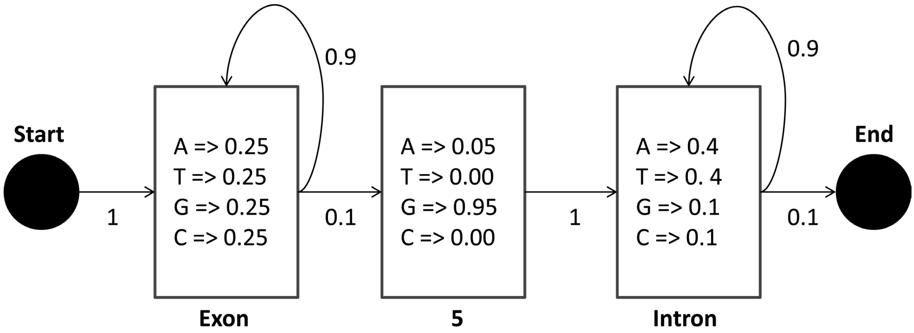


Figure 2 Hidden Markov model. The HMM is designed to predict the G rich splice site. The value inside the boxes show emission probabilities, that is, the probability for each nucleotide to appear while the values outside show transition probabilities to move from one state to the next. HMM representation adapted from (9).

Neural networks

Artificial neural networks is another classification technique with numerous applications in computational biology. Neuron is the basic unit of an artificial neural network. Each neuron can have multiple input connections with weights assigned to each of them. The output value from the neurons is calculated according to its activation function. A neural network may consist of multiple layers, with each layer containing multiple neurons. Figure 3 shows a multi-layered neural network with 32 neurons and 192 edges. Neural networks are used in supervised learning and classification. This approach uses labeled data and follows the main steps listed below:

(i) *Dataset*: Divide the data into training sets and testing set (mostly 70–30% split or 60–40% split, respectively).
(ii) *Training*: Use the training data to traverse over the neuron and estimate the output.
(iii) *Iterate*: Based on the difference between the actual and estimated output, calculate the error and adjust the weights accordingly. Repeat step 2.
(iv) *Testing*: After multiple iterations between step 2 and 3, the model is trained and can be tested. Use the test set (unseen data for model) to compute the output. As the actual label is known, the accuracy and sensitivity can be calculated based on the correct (true positives or true negatives) and incorrect classifications (false positives or false negative).
(v) *Validation*: The training- and test-set splits are randomized and new sets are created from the existing dataset. This new test-train split is then used again iterating over steps 2–4. The idea is to create a model independent for generalized datasets. Depending on situations, there can be multiple iterations for this step and hence referred to k-fold cross validation.

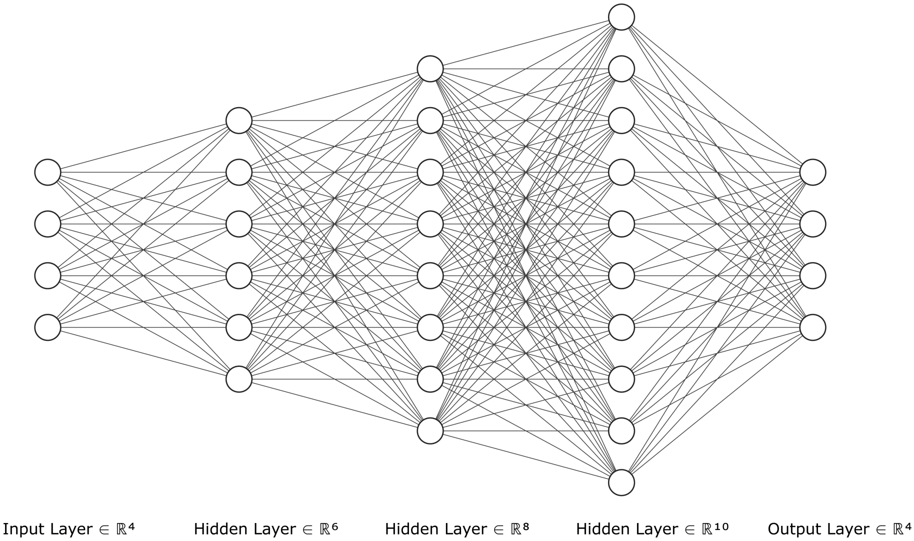

Input Layer $\in \mathbb{R}^4$
Hidden Layer $\in \mathbb{R}^6$
Hidden Layer $\in \mathbb{R}^8$
Hidden Layer $\in \mathbb{R}^{10}$
Output Layer $\in \mathbb{R}^4$

Figure 3 Neural network representation. Each node represents a neuron, and the edges depict weights that connect the neurons between layers. After each iteration, the weights are adjusted to correct for error.

In order to assess the performance of the model, outputs are calculated from different models based on different activation functions or even different neural network architectures. Sensitivity (recall) and accuracy are calculated for each of the models, and the best performing model should have a high recall rate.

The performance of machine learning in general and neural networks in particular depends highly on the quality of the data. A high-quality data would have low noise/junk while having a high homogeneity. Noise in biological data can refer to ambiguous sequence elements or incorrect labels. A high homogeneity results in an equal distribution of diversity in data across different data splits. Assuring the good quality of data before model training is a very important and time-consuming step for data scientists. If the training dataset is not a homogenous representative of the population, it can lead to a biased classification in the models. A bias model can show promising results for the testing dataset but fails in the actual world. This happens because the model is trained to classify only those types of cases that it observed during the training, and a bias sample resulted in a skewed perception of the real-world scenario. The quality of classification from neural networks also depends highly on the training iterations and size of datasets. While the ability for high-powered computation has greatly increased in the last decade, coupled with biological big data, neural networks can be used to train accurate classifiers. Neural networks have now evolved into their more complex form called "Dense Networks" or "Deep learning." These networks (e.g., LSTM) comprise numerous neurons and high number of hidden layers between the input and output layers (hence deep network). Although the depth of a network results in a better-quality model, they are difficult to train due to the requirement of high computing power.

NEXT-GENERATION SEQUENCING

The last three decades have seen a continuous evolution of sequencing technologies. Starting from traditional Sanger sequencing to whole genome shot gun sequencing by Craig Venter and later next-generation sequencing (NGS) (4). The latest amongst these is the "Nanopore," highly compact and efficient sequencing that connects to a computer via USB; it is easily transportable and fits on a small desktop. The technology that initially required thousands of dollars per nucleotide is much cheaper now. An NGS pipeline comprises of two main sections: a wet lab section involves sample preparation, amplification, and sequencing; and the second section involves a bioinformatics workflow that uses the data generated by the wet lab to derive a sequence and other information. It is important to note that the bioinformatics section involves sequence analysis algorithms that are based on statistical and heuristic techniques to analyze and generate sequences. This section focuses on the bioinformatics aspect of NGS since it has evolved an ecosystem of computational algorithms and pipelines around it for accurate and efficient sequencing. NGS is a massively parallel sequencing technology, also referred as high-throughput sequencing, that allows analysis of large fragments of DNA and RNA genomes with high sensitivity, much more quickly and cheaply than the

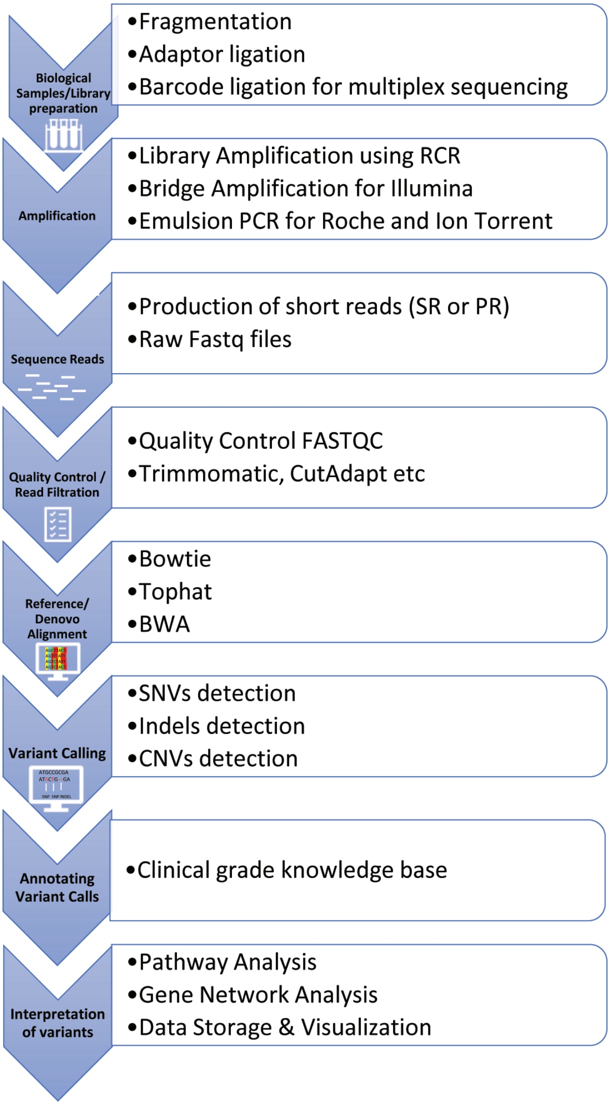

Biological Samples/Library preparation
•Fragmentation
•Adaptor ligation
•Barcode ligation for multiplex sequencing
Amplification
•Library Amplification using RCR
•Bridge Amplification for Illumina
•Emulsion PCR for Roche and Ion Torrent
Sequence Reads
•Production of short reads (SR or PR)
•Raw Fastq files
Quality Control / Read Filtration
•Quality Control FASTQC
•Trimmomatic, CutAdapt etc
Reference/ Denovo Alignment
•Bowtie
•Tophat
•BWA
Variant Calling
•SNVs detection
•Indels detection
•CNVs detection
Annotating Variant Calls
•Clinical grade knowledge base
Interpretation of variants
•Pathway Analysis
•Gene Network Analysis
•Data Storage & Visualization

Figure 4 Overview of NGS data analysis workflow. The steps involved in high-throughput sequencing of biological data: (i) biological samples/library preparation, (ii) amplification, (iii) sequence reads, (iv) quality control/read filtration, (v) alignment, (vi) variant calling, (vii) annotating variant calls, and (viii) interpretation of variants.

previously used Sanger sequencing methodology. In NGS, different platform technologies follow the same eight major steps (Figure 4):

(i) *Library preparation*: The first step in NGS workflow involves preparation of high-quality and high-yield sequence library. The isolated genomic DNA or RNA is sheared into smaller fragments ranging from 150–5000 base pairs (bp) depending on the sequencing platform. The desired library can be created using either of the two fragmentation approaches, mechanical shearing or enzyme-based fragmentation (28, 29). Mechanical shearing methods include acoustic shearing, needle-shear, sonication, and nebulization, whereas enzyme-based methods involve transposons and restriction enzymes (endonucleases) (30). The small fragments known as reads have short overhangs (sticky ends) of 5'-phosphate and 3'-hydroxl groups. These ends are repaired by adenylation at 3' ends resulting in adapter ligation that is important for amplification. During library preparation, unique barcodes can be added to the fragments facilitating multiple sequencing of various samples in the same run (31).

(ii) *Amplification*: The goal of this step is to create thousands of copies for each read. The library is loaded onto the flow-cell, and the fragments are amplified using clonal amplification methods such as emulsion PCR or bridge amplification. In emulsion PCR, the library is amplified within a tiny water droplet floating in an oil solution (32, 33). In bridge amplification, the single-stranded DNA from the library is hybridized to the flow-cell's surface-bound forward and reverse oligos that are complementary to the library adapter sequences. Hybridized at one end, the singe-stranded DNA then folds over to form a bridge and binds to adapter-complementary oligos at the other end. DNA polymerase adds nucleotides to amplify DNA, and a clonal cluster is generated as the original strand is washed away leaving complementary strands of amplified DNA attached to the flow cell. (34).

(iii) *Sequencing*: The amplified individual sequences are sequenced using different platforms and sequencing technologies that include Illumina (Solexa) sequencing, Roche 454 sequencing, and Ion Torrent (Proton/PGM sequencing). Illumina (Solexa) sequencing works by simultaneously identifying DNA bases (A, T, C or G), and each base emits a unique fluorescent signal as it is added to the nucleic acid chain. Illumina sequencing involves 100–150 bp read length. Illumina has some variations that mainly differ in the amount of DNA sequenced in one run (Table 1). Roche 454 sequencing is based on pyrosequencing; a technique that detects pyrophosphate release, again

TABLE 1 **Comparison of Illumina sequencing platforms**

Sequencing platforms	Run time	Max output (Gb)	Max read number (million)	Max read length (bp)
iSeq Series	9–17.5 hours	1.2	4	2 × 150
MiniSeq Series	4–24 hours	7.5	25	2 × 150
MiSeq Series	4–55 hours	15	25	2 × 300
NextSeq Series	13–20 hours	120	400	2 × 150
HiSeq Series	<1–3.5 days	1500	5000	2 × 150
HiSeq X Series	<3 days	1800	6000	2 × 150

Different attributes and key features of different Illumina platforms include run time, maximum output, maximum read number, and maximum read length.

using fluorescence, after nucleotides are incorporated by polymerase to a new strand of DNA. Roche 454 sequencing produces sequence reads of up to 1000 bp in length. Like Illumina, it does this by sequencing multiple reads at once by reading optical signals as bases are added. Ion Torrent (Proton / PGM sequencing) measures the direct release of H+ (protons) from the incorporation of individual bases by DNA polymerase and therefore differs from the previous two methods as it does not measure light. As in other kinds of NGS, the input DNA or RNA is fragmented, this time ~200 bp. These sequencing technologies result in raw sequencing reads (20 to 1000 bp) stored in the FASTQ format which contains both the nucleotide sequence and its corresponding quality scores. These reads can be either "single-ended" or "paired-ended." Paired-end reads are produced when the fragment size used in the sequencing process is much longer (typically 250–500 bp long).

(iv) *Quality control and read filtration*: After sequencing is complete, the read data are in electronic form and can be processed to generate a whole genome or a specific gene sequence using a bioinformatics NGS pipeline. Although quality control and filtration is the fourth step in generating a full analyzable sequence, it is the first step in a bioinformatics NGS pipeline. Read filtration involves removing low confidence and erroneous reads from the dataset. The amplified raw reads pass through quality control check using FastQC (35) that can produce a detailed report on the number, quality, and coverage of reads. These methods mostly work on sequence analysis techniques like clustering short reads to calculate their frequency and quality scores. It is followed by read filtration, clipping of adapters and low-quality base pairs from 3' and 5' ends using software such as CutAdapt (36), trimmomatic (37) and others.

(v) *Alignment*: Once the read quality is acceptable, millions of raw sequence reads (single-end or paired-end) are mapped and aligned using either a reference based assembly (in which reference sequence is available) or de novo assembly (in the absence of a reference sequence). The sequence reads of variable lengths are aligned using different bioinformatics alignment tools such as BWA (38), Bowtie (39), and TopHat (40). These heuristic-based aligners allow fast sequence alignment and generate a consensus sequence from the alignment by searching the overlapping portions of the reads and merging them into longer reads in order to construct a region of interest, that is, genes or a whole genome. The main aim of this step is to generate a consensus sequence from the millions of reads. A consensus sequence shows the genetic makeup at the time of the sample collection. This step marks the completion of sequence generation for a partial or a whole genome. The following steps are important for an in-depth analysis beyond generation of only a single sequence.

(vi) *Variant identification*: NGS is not only time efficient but also provides the data for an in-depth sequence analysis. Variant analysis uses the reads file to determine the conserved and variable nucleotides at specific positions. As this process involves statistical calculations spanning over millions of reads, it is both a time and computationally intensive process. Bootstrap resampling of reads can be used to assess the quality of variant calling scores. The variations within the genomic sequences such as single-nucleotide polymorphisms (SNPs), single-nucleotide variants (SNV), and indels (insertions and deletions) are detected using software such as SAMtools (41), Genome Analysis Toolkit (GATK) (42), and VarScan (43, 44). Both SAMtools and GATK use the Bayesian probabilistic approach to identify true variants from alignment errors, whereas VarScan uses a heuristic approach. Most NGS methods for SNV detection are designed to detect germline variations in an individual's genome, whereas the variations that are identified within a population are referred as SNPs.

(vii) *Annotation*: The genetic variants detected are annotated based on the published peer-reviewed literature and public genetic variant databases.

(viii) *Interpretation of variants*: Lastly, medical professionals will interpret these variants and obtain the patient's clinical history in order to establish a most accurate diagnosis. This includes examining different disease pathways and gene network analysis and identifying actual mutations causing a disease.

APPLICATIONS OF NGS IN CLINICAL PRACTICE

The NGS technologies have several applications in research to solve a diverse range of biological problems. Comprehensive analysis of NGS data includes whole-genome sequencing, gene expression determination, transcriptome profiling, and epigenetics. NGS has enabled the researches to sequence large segments of the genome (i.e., whole-genome sequencing) and provides insights into identifying and understanding the genetic variants such as SNPs, insertions, and deletions of DNA, and rearrangements such as translocation and inversions associated with diseases for further targeted studies (45). Researchers use RNA sequencing (RNASeq) to uncover genome-wide transcriptome characterization and profiling (46). Analysis involving genome-wide gene expression (i.e., gene transcription, post-transitional modifications, and translation) and the molecular pathway analysis provide a deeper understanding of gene regulation in neurological, immunological, and other complex diseases. Other applications include studying heritable changes in gene regulation that occur without a change in the DNA sequence. Epigenetics play a significant role in growth, development, and disease progression. The studies on epigenetic changes in cancer provide insight into important tumorigenic pathways (47, 48).

CONCLUSION

Sequence analysis is a broad area of research with sub-domains. Alignment of sequences can reveal important information concerning the structural and functional sites within sequences. It is used to explore the evolutionary path of sequences by identifying the sequence orthologs and homologs. Sequence analysis also involves the use of machine learning techniques for classification and prediction of sequence elements. Statistical methods are used to create sequence profiles and identify other distantly related sequences with a higher precision. Advancement of sequencing technologies has resulted in a next-generation era that opened the doors to personalized medicine and haplotype/quasi-species detection. With correctly organized NGS pipelines, it is possible to analyze the effects of drugs directly at the sequence level.

Acknowledgement: Usman Saeed acknowledges the funding support by Dennemeyer Octimine GmbH, Landaubogen 1-3, 81373 München, Germany.

Conflict of interest: The authors declare no potential conflict of interest with respect to research, authorship, and/or publication of this chapter.

Copyright and permission statement: To the best of our knowledge, the materials included in this chapter do not violate copyright laws. All original sources have been appropriately acknowledged and/or referenced. Where relevant, appropriate permissions have been obtained from the original copyright holder(s).

REFERENCES

1. Locey KJ, Lennon JT. Scaling laws predict global microbial diversity. Proc Natl Acad Sci U S A. 2016;113(21):5970–5. http://dx.doi.org/10.1073/pnas.1521291113
2. Watson JD, Crick FH. Molecular structure of nucleic acids; A structure for deoxyribose nucleic acid. Nature. 1953;171(4356):737–8. http://dx.doi.org/10.1038/171737a0
3. Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, Baldwin J, et al. Initial sequencing and analysis of the human genome. Nature. 2001;409(6822):860–921. http://dx.doi.org/10.1038/35057062
4. Venter JC, Adams MD, Myers EW, Li PW, Mural RJ, Sutton GG, et al. The sequence of the human genome. Science. 2001;291(5507):1304–51. http://dx.doi.org/10.1126/science.1058040
5. Pennisi E. Human genome. A low number wins the GeneSweep Pool. Science. 2003;300(5625):1484. http://dx.doi.org/10.1126/science.300.5625.1484b
6. Crick F. Central dogma of molecular biology. Nature. 1970;227(5258):561–3. http://dx.doi.org/10.1038/227561a0
7. Anfinsen CB. Principles that govern the folding of protein chains. Science. 1973;181(4096):223–30. http://dx.doi.org/10.1126/science.181.4096.223
8. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol. 1970;48(3):443–53. http://dx.doi.org/10.1016/0022-2836(70)90057-4
9. Durbin R, Eddy SR, Krogh A, Mitchison GJ. Biological sequence analysis: Probabilistic models of proteins and nucleic acids. Cambridge: Cambridge University Press; 1998.
10. Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol. 1981;147(1):195–7. http://dx.doi.org/10.1016/0022-2836(81)90087-5
11. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403–10. http://dx.doi.org/10.1016/S0022-2836(05)80360-2
12. Kent WJ. BLAT—The BLAST-like alignment tool. Genome Res. 2002;12(4):656–64. http://dx.doi.org/10.1101/gr.229202
13. Lipman DJ, Pearson WR. Rapid and sensitive protein similarity searches. Science. 1985;227(4693):1435–41. http://dx.doi.org/10.1126/science.2983426
14. Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. Proc Natl Acad Sci U S A. 1988;85(8):2444–8. http://dx.doi.org/10.1073/pnas.85.8.2444
15. Husi H, Skipworth RJ, Fearon KC, Ross JA. LSCluster, a large-scale sequence clustering and aligning software for use in partial identity mapping and splice-variant analysis. J Proteomics. 2013;84:185–9. http://dx.doi.org/10.1016/j.jprot.2013.04.006
16. Edgar RC. Search and clustering orders of magnitude faster than BLAST. Bioinformatics. 2010;26(19):2460–1. http://dx.doi.org/10.1093/bioinformatics/btq461
17. Rognes T, Flouri T, Nichols B, Quince C, Mahe F. VSEARCH: A versatile open source tool for metagenomics. PeerJ. 2016;4:e2584. http://dx.doi.org/10.7717/peerj.2584
18. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nat Methods. 2015;12(1):59–60. http://dx.doi.org/10.1038/nmeth.3176

19. Suzuki S, Kakuta M, Ishida T, Akiyama Y. GHOSTX: An improved sequence homology search algorithm using a query suffix array and a database suffix array. PLoS One. 2014;9(8):e103833. http://dx.doi.org/10.1371/journal.pone.0103833
20. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, et al. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Res. 1997;25(17):3389–402. http://dx.doi.org/10.1093/nar/25.17.3389
21. Remmert M, Biegert A, Hauser A, Soding J. HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat Methods. 2011;9(2):173–5. http://dx.doi.org/10.1038/nmeth.1818
22. Guindon S, Gascuel O. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst Biol. 2003;52(5):696–704. http://dx.doi.org/10.1080/10635150390235520
23. Saitou N, Nei M. The neighbor-joining method: A new method for reconstructing phylogenetic trees. Mol Biol Evol. 1987;4(4):406–25.
24. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. http://dx.doi.org/10.1162/neco.1997.9.8.1735
25. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017;60(6):84–90. http://dx.doi.org/10.1145/3065386
26. Forgy EW. Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. Biometrics. 1965;21(3):768–9.
27. Van Dongen S. Graph clustering by flow simulation. Utrecht: University of Utrecht, 2000.
28. Head SR, Komori HK, LaMere SA, Whisenant T, Van Nieuwerburgh F, Salomon DR, et al. Library construction for next-generation sequencing: Overviews and challenges. Biotechniques. 2014;56(2):61–4, 6, 8, passim. http://dx.doi.org/10.2144/000114133
29. Knierim E, Lucke B, Schwarz JM, Schuelke M, Seelow D. Systematic comparison of three methods for fragmentation of long-range PCR products for next generation sequencing. PLoS One. 2011;6(11):e28240. http://dx.doi.org/10.1371/journal.pone.0028240
30. Marine R, Polson SW, Ravel J, Hatfull G, Russell D, Sullivan M, et al. Evaluation of a transposase protocol for rapid generation of shotgun high-throughput sequencing libraries from nanogram quantities of DNA. Appl Environ Microbiol. 2011;77(22):8071–9. http://dx.doi.org/10.1128/AEM.05610-11
31. Dodt M, Roehr JT, Ahmed R, Dieterich C. FLEXBAR—Flexible Barcode and Adapter Processing for Next-Generation Sequencing Platforms. Biology (Basel). 2012;1(3):895–905. http://dx.doi.org/10.3390/biology1030895
32. Dressman D, Yan H, Traverso G, Kinzler KW, Vogelstein B. Transforming single DNA molecules into fluorescent magnetic particles for detection and enumeration of genetic variations. Proc Natl Acad Sci U S A. 2003;100(15):8817–22. http://dx.doi.org/10.1073/pnas.1133470100
33. Nakano M, Komatsu J, Matsuura S, Takashima K, Katsura S, Mizuno A. Single-molecule PCR using water-in-oil emulsion. J Biotechnol. 2003;102(2):117–24. http://dx.doi.org/10.1016/S0168-1656(03)00023-3
34. Pemov A, Modi H, Chandler DP, Bavykin S. DNA analysis with multiplex microarray-enhanced PCR. Nucleic Acids Res. 2005;33(2):e11. http://dx.doi.org/10.1093/nar/gnh184
35. Andrews, S. FastQC: a quality control tool for high throughput sequence data. 2010. http://www.bioinformatics.babraham.ac.uk/projects/fastqc
36. Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet J. 2011;17:10–2. http://dx.doi.org/10.14806/ej.17.1.200
37. Bolger AM, Lohse M, Usadel B. Trimmomatic: A flexible trimmer for Illumina sequence data. Bioinformatics. 2014;30(15):2114–20. http://dx.doi.org/10.1093/bioinformatics/btu170
38. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. Cambridge: Broad Institute of Harvard and MIT; 2013.
39. Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. 2009;10(3):R25. http://dx.doi.org/10.1186/gb-2009-10-3-r25
40. Trapnell C, Pachter L, Salzberg SL. TopHat: Discovering splice junctions with RNA-Seq. Bioinformatics. 2009;25(9):1105–11. http://dx.doi.org/10.1093/bioinformatics/btp120

41. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The sequence alignment/map format and SAMtools. Bioinformatics. 2009;25(16):2078–9. http://dx.doi.org/10.1093/bioinformatics/btp352

42. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, et al. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res. 2010;20(9):1297–303. http://dx.doi.org/10.1101/gr.107524.110

43. Koboldt DC, Chen K, Wylie T, Larson DE, McLellan MD, Mardis ER, et al. VarScan: Variant detection in massively parallel sequencing of individual and pooled samples. Bioinformatics. 2009;25(17):2283–5. http://dx.doi.org/10.1093/bioinformatics/btp373

44. Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, et al. VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Res. 2012;22(3):568–76. http://dx.doi.org/10.1101/gr.129684.111

45. Peng L, Bian XW, Li DK, Xu C, Wang GM, Xia QY, et al. Large-scale RNA-Seq Transcriptome Analysis of 4043 Cancers and 548 Normal Tissue Controls across 12 TCGA Cancer Types. Sci Rep. 2015;5:13413. http://dx.doi.org/10.1038/srep13413

46. Koh Y, Park I, Sun CH, Lee S, Yun H, Park CK, et al. Detection of a distinctive genomic signature in Rhabdoid Glioblastoma, A rare disease entity identified by whole exome sequencing and whole transcriptome sequencing. Transl Oncol. 2015;8(4):279–87. http://dx.doi.org/10.1016/j.tranon.2015.05.003

47. Kinsella M, Harismendy O, Nakano M, Frazer KA, Bafna V. Sensitive gene fusion detection using ambiguously mapping RNA-Seq read pairs. Bioinformatics. 2011;27(8):1068–75. http://dx.doi.org/10.1093/bioinformatics/btr085

48. McPherson A, Hormozdiari F, Zayed A, Giuliany R, Ha G, Sun MG, et al. deFuse: An algorithm for gene fusion discovery in tumor RNA-Seq data. PLoS Comput Biol. 2011;7(5):e1001138. http://dx.doi.org/10.1371/journal.pcbi.1001138