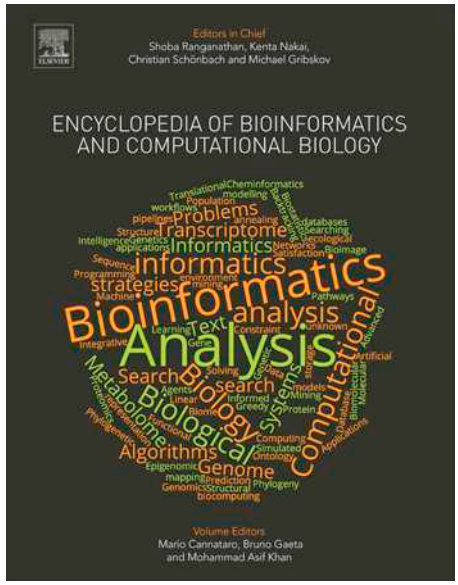

Yosvany López, Piotr J. Kamola, Ronesh Sharma, Daichi Shigemizu, Tatsuhiko Tsunoda and Alok Sharma (2019) Computational Pipelines and Workflows in Bioinformatics. In: Ranganathan, S., Gribskov, M., Nakai, K. and Schönbach, C. (eds.), Encyclopedia of Bioinformatics and Computational Biology, vol. 3, pp. 113–134. Oxford: Elsevier.

Computational Pipelines and Workflows in Bioinformatics

Yosvany López, Genesis Healthcare Co., Tokyo, Japan
Piotr J Kamola, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan and Japan Science and Technology Agency, Tokyo, Japan
Ronesh Sharma, University of the South Pacific, Suva, Fiji and Fiji National University, Suva, Fiji
Daichi Shigemizu, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan; Japan Science and Technology Agency, Tokyo, Japan; RIKEN Cluster for Science and Technology Hub, Yokohama, Japan; National Center for Geriatrics and Gerontology, Obu, Japan; and Tokyo Medical and Dental University, Tokyo, Japan
Tatsuhiko Tsunoda, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan; Japan Science and Technology Agency, Tokyo, Japan; and Tokyo Medical and Dental University, Tokyo, Japan
Alok Sharma, RIKEN Center for Integrative Medical Sciences, Yokohama, Japan; University of the South Pacific, Suva, Fiji; and Griffith University, Brisbane, QLD, Australia

Introduction

The research field of bioinformatics is gaining momentum due to its role in the examination of huge volumes of biological data. Nowadays, technological advancements in sequencer and microscope technology have opened many novel and interesting avenues to study organisms at the molecular level. These advances should be accompanied by efficient computational tools and pipelines, geared towards processing and analysis of the data. This combination will allow for more accurate quantification of molecular interaction, and lead to better understanding of the underlying mechanisms. In the field of molecular biology, next-generation sequencing (NGS) technology is playing a key role already. For instance, studies aimed at assessing DNA–protein interactions often make use of chromatin immunoprecipitation-sequencing (ChIP-seq) data, whereas those focused on trying to understand the genetics behind rare diseases or cancers take advantage of whole genome (WGS) or whole exome sequencing (WES) data. Studies that focus on profiling coding and noncoding RNA on a larger scale have the choice of DNA-chip and RNA-sequencing (RNA-seq) technologies. Complex cellular structures can be studied with microscopes that are capable of generating high-quality images. Given the above circumstances, computational algorithms and methodologies are necessary to face such challenges (Roy *et al.*, 2016, 2018; Metzker, 2010). Bioinformatics pipelines combine a set of software tools, each designed to deal with a specific task, to take full advantage of the biological information.

In this article, we discuss the computational pipelines and workflows used for analyzing a wide range of sequencing and biomedical image information. Accordingly, it is hereafter divided into five sections. Section "Genome Analysis" covers genome analysis pipelines, especially tools for dealing with ChIP-seq and genomic variation. Section "Transcriptome Analysis" describes transcriptome strategies, including differential expression analysis, pathway and network approaches, and machine learning (ML) methods for sample subclassification and biomarker discovery. Section "Workflows in Proteomics" focuses on proteomics, specifically on molecular recognition of features in intrinsically disordered proteins (IDPs). Section "Bioimage Analysis Pipelines" briefly introduces the standard workflow for processing bioimages from cellular structures, while Section "Conclusions" highlights the challenges and future directions.

Genome Analysis

The function of molecular structures is ultimately encoded in the genome sequence, thereby bioinformatics pipelines designed for its analysis are of utmost importance. In this section, we present the workflow and tools available for effectively analyzing ChIP-seq and genomic variation data.

ChIP-Seq Pipeline

A great number of studies have been currently focusing on deciphering the molecular consequences of DNA–protein interactions. In this case, NGS data has played a key role because it allows us to sequence the DNA sequence around binding sites. Subsection "ChIP-Seq Pipeline" introduces the standard workflow (**Fig. 1**) used by many laboratories for such analyses.

Alignment

Because read quality is often affected by artifacts such as adapter contamination and base calling errors, the quality check of reads is sometimes the first procedure before mapping the raw reads to a reference genome. It is here where biases or sequencing errors are identified and low-quality reads are filtered out. Given its flexibility and applicability to many sequencing platforms, the software FastQC (Babraham-Bioinformatics, 2018) is frequently utilized. A more detailed explanation and additional tools can be found in Pabinger *et al.* (2014).

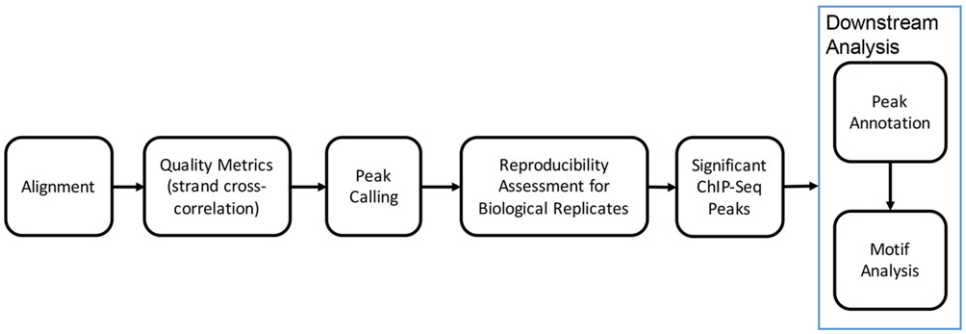

Fig. 1 Standard workflow for the analysis of ChIP-seq data.

After conducting the above filtering, high-quality reads are aligned to the corresponding reference genome using aligners such as Bowtie2 (Langmead and Salzberg, 2012), SOAP (Li *et al.*, 2009b), BWA (Li and Durbin, 2009) and MAQ (Li *et al.*, 2008). It is worth noting that unlike transcriptome analysis pipelines, aligners capable of detecting splice sites are not required whatsoever. The alignment results (the number of uniquely mapped reads) should be carefully checked to validate the success of the mapping procedure.

Quality metrics

When processing ChIP-seq data, the assessment of immunoprecipitation enrichment, sequencing depth or fragment-size selection is a must. This is often referred to as the signal-to-noise ratio and assessed with metrics like strand cross-correlation (Landt *et al.*, 2012) or immunoprecipitation enrichment estimation (Landt *et al.*, 2012). Although the package CHANCE (Diaz *et al.*, 2012) is widely utilized, current peak callers such as MACS (Zhang *et al.*, 2008) already include strand cross-correlation analysis.

Peak calling

Peak calling is one of the most important steps in a ChIP-seq analysis pipeline. It is here where bound DNA regions are detected. Of note, if the improvement of specificity is prioritized, duplicate reads should be removed before peak calling. Moreover, the success of this step will depend on the peak caller and the type of protein being analyzed. For the purpose of analysis, DNA-binding proteins (DBPs) are divided into three groups depending on their binding signals: point-source DBPs, broadly enriched DBPs, and DBPs with mixed characteristics.

Point-source DBPs

These proteins are the most common and therefore many peak callers have been designed to deal with their signals. For instance, peak callers like MACS (Zhang *et al.*, 2008) and SPP (Kharchenko *et al.*, 2008) are able to accurately detect the size of the genomic gap between reads aligned to plus and minus strands. Other software such as BEADS (Cheung *et al.*, 2011) use background models from control samples or GC content for reducing noise. To assess enriched peaks, software such as MACS (Zhang *et al.*, 2008) and CisGenome (Jiang *et al.*, 2010), which rely on statistical distributions, or BayesPeak (Spyrou *et al.*, 2009) and HPeak (Qin *et al.*, 2010), which implement hidden Markov models, are often utilized.

Broadly enriched DBPs

These proteins are usually involved in epigenetic regulation. For their analysis, peak callers such as SICER (Xu *et al.*, 2014) and ZINBA (Rashid *et al.*, 2011) can be utilized to detect broad regions, whereas others such as MACS (Zhang *et al.*, 2008), PeakRanger (Feng *et al.*, 2011), and SPP (Kharchenko *et al.*, 2008) can be used as long as their bandwidth is increased and their peak cut-off decreased.

DBPs with mixed signals

For analyzing molecules with mixed characteristics, for example the RNA Polymerase II, peak callers such as MACS (Zhang *et al.*, 2008), ZINBA (Rashid *et al.*, 2011), and PeakRanger (Feng *et al.*, 2011) can be employed.

Assessment of reproducibility

Because the quality of detected peaks could be affected by the peak caller, sequencing depth, or the number of binding sites, pipelines should always include an assessment of reproducibility. This procedure is intended to check whether the peaks are really reproducible, and should comprise at least two replicates of the same experiment. To do this, it is advisable to filter out those regions with high ChIP signals and then compute the Pearson correlation coefficient of mapped reads at genomic positions. The consistency of peak sets detected in biological replicates is evaluated with the irreproducible discovery rate, and those peaks passing a specific cut-off are finally retrieved. This assessment is often conducted as described in Li *et al.* (2011).

Differential binding analysis

A typical ChIP-seq experiment aims to detect binding differences between biological conditions. Current studies follow two strategies: quantitative and qualitative. The quantitative approach takes into consideration the differential binding between conditions by using the number of reads or the density of reads in peaks. For this, tools such as DBChIP (Liang and Keleş, 2012), which relies on read counts or MAnorm (Shao *et al.*, 2012), which uses read densities, can be utilized. The qualitative approach considers overlapping peaks and should only be regarded for exploration purposes. It is important to point out that reproducible peaks should be computed per condition before regarding one of the above strategies.

Downstream analysis

This step comprises important tasks such as peak annotation and motif analysis. The detection of peaks is often not the ultimate goal, but it is followed by the discovery of functional genomic regions. To do this, the peaks should be first formatted in BED or GFF files so that they can be easily analyzed with a genome browser, or annotated with in-house scripts. Current packages such as BEDTools (Quinlan and Hall, 2010) include functions for annotation purposes and calculate the genomic distance of each peak to relevant genomic features. Other libraries like ChIPpeakAnno (Zhu *et al.*, 2010) can be further used for associating binding sites with gene expression activity. On the other hand, motif analysis could also help to identify those genomic sites bound by a specific regulatory protein, thereby facilitating the validation of the ChIP experiment. To detect motif sequences, the genomic regions of the detected peaks are analyzed with motif-discovery software. These algorithms complement each other so that it is advisable to consider different methods. For instance, algorithms such as MEME-ChIP (Machanick and Bailey, 2011) and peak-motifs (Thomas-Chollier *et al.*, 2012) are often employed. Detailed information on motif analysis can be found in Zambelli *et al.* (2013).

Mutation and Genomic Variation

The low cost of NGS technology has made possible international collaborations such as the 1000 Genomes (Genomes Project *et al.*, 2015) and HapMap (International Hapmap Consortium, 2005) Projects. These efforts have provided detailed catalogs of human genetic variation, which are of vital importance to identify disease-associated variants. Consequently, recent studies have aimed at pinpointing genetic risk factors for common diseases, including rheumatoid arthritis and type 2 diabetes, cancers, and Mendelian diseases such as long QT syndrome and Huntington's disease. Because the accurate detection of variants proves critical for clinical treatments and novel drugs, the main challenges are no longer in the sequencing technology but in the development of suitable bioinformatics pipelines. This section addresses two impacted areas: genome-wide association studies (GWAS), and whole genome and exome analysis.

Genome-wide association studies

The genetics behind complex diseases has been long studied by focusing on disease-related genes. However, the massive amounts of genotype data have paved the way for broader approaches like GWAS, which regard linkage disequilibrium at the population level. GWAS has proven absolutely necessary for identifying the genetic risk factors for common diseases (Ozaki *et al.*, 2002; Newton-Cheh *et al.*, 2009), and investigating associations between single nucleotide polymorphisms (SNPs) from case/control studies. The most common tool for GWAS analysis is PLINK (Purcell *et al.*, 2007). PLINK is an open-source and freely available software, which also provides quality controls of the data. It requires two standard formats: PED and MAP files, though binary files such as BED, BIM, and FAM are acceptable as well. The PED file is a space or tab delimited file, composed of a family identifier, individual identifier, father and mother identifiers, gender, phenotype, and genotypes. Of note, when creating a PED file, strand orientation should be checked for allele calls (i.e., forward or reverse complement). The TOP/BOT strand and A/B allele coding, developed by Illumina, along with the database of genetic variation dbSNP (Sherry *et al.*, 2001) are widely used for uniformly designating SNP entries. The MAP file consists of a chromosome, identifier, genetic distance, and physical position for each marker. As for the experimental dataset, it should pass different quality checks in which the samples and markers are carefully examined. The samples should be checked for (1) sex inconsistencies (–check-sex[a]), (2) inbreeding coefficient (–het[a] 0.1), (3) genotype missingness (–missing[a] 0.05), (4) kinship coefficient (–genome[a] 0.2), and (5) population stratification. Similarly, the markers should be checked for (1) genotyping efficiency or call rate (–geno[a] 0.99), (2) minor allele frequency (–freq[a] 0.01), and (3) Hardy–Weinberg equilibrium (–hwe[a] 0.001). The remaining dataset can now be used for association analysis (–assoc[a]) whose significance cut-off should be set at $p < 5 \times 10^{-8}$. These association results can be further visualized with Q-Q and Manhattan plots included in the R package *qqman*.

Often, the number of markers in an experimental dataset is not enough for accurate analysis, limiting the power and resolution of the GWAS strategy. To overcome this limitation, genotype imputation at ungenotyped loci is strongly recommended. The most common imputation software include IMPUTE2 (Marchini *et al.*, 2007), BEAGLE (Browning and Browning, 2009), and minimac (Neumann *et al.*, 2010). Of note, reference panels from the HapMap (International Hapmap Consortium, 2005) or 1000 Genomes (Genomes Project *et al.*, 2015) Projects should be used.

Thus far, we have described a typical SNP-based association analysis. However, gene-based association approaches can sometimes turn out effective. These statistical methods are classified into three categories: (1) the burden test (Li and Leal, 2008; Madsen and Browning, 2009; Morgenthaler and Thilly, 2007; Price *et al.*, 2010), (2) the variance component test (Wu *et al.*, 2011),

[a]PLINK parameter

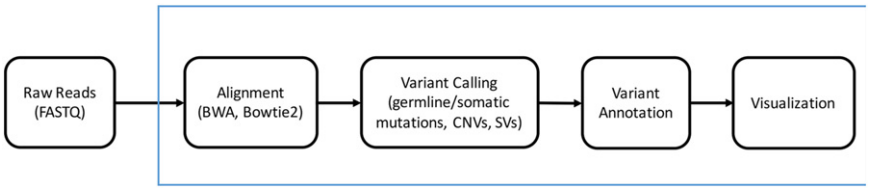

Fig. 2 Standard workflow for whole genome/exome sequencing data analysis.

and (3) the combination methods (Lee *et al.*, 2012). Recently, a method called SMR, which integrates summary-level data of GWAS and expression quantitative trait locus, was developed (Zhu *et al.*, 2016). Additional examples of risk prediction models for several diseases, which regarded disease-associated genes, can be found in Imamura *et al.* (2013); Shigemizu *et al.* (2014).

Genome and exome analysis pipelines

For rare Mendelian diseases, NGS technology (Metzker, 2010; Rusk and Kiermer, 2008) has made WGS and WES possible at an individual level. Extensive studies have revised the pipelines aimed to detect genetic variations by analyzing NGS data (Hwang *et al.*, 2015; Pabinger *et al.*, 2014). A standard pipeline consists of five main steps: quality check of raw reads, alignment to a reference genome, variant identification, variant annotation, and visualization (**Fig. 2**). The first two steps are discussed in Section "Alignment" so that we will focus on the remaining steps.

Variant identification

The identification of genomic variants is the main step of a variant detection pipeline. This step is intended to detect genomic variations between biological samples. Most of the available software require the aligned reads in BAM format and sorted by genomic coordinates. Both requirements can be easily achieved with SAMtools (Li *et al.*, 2009a).

The current variant identification tools can be used for detecting (1) germline mutations, (2) somatic mutations, (3) copy number variations (CNVs), and (4) structural variations (SVs). Among the variant callers for discovering germline mutations are the Genome Analysis Tool Kit (GATK) HaplotypeCaller (McKenna *et al.*, 2010), SAMtools (Li *et al.*, 2009a) and VarScan 2 (Koboldt *et al.*, 2012). The GATK includes a statistical model based on Bayesian genotype likelihood for computing genotypes and allele frequencies. Nevertheless, one disadvantage is that it produces many false positives which have to be filtered for downstream analysis (McKenna *et al.*, 2010). SAMtools can be easily integrated into one single pipeline because of its additional functions for aligning reads, as well as sorting and indexing alignment results (Li *et al.*, 2009a). VarScan 2 implements heuristic and statistical approaches for detecting single nucleotide variants, indels, somatic mutations, and copy number alterations between cancer and control samples (Koboldt *et al.*, 2012). SAMtools and VarScan 2, along with SomaticSniper (Larson *et al.*, 2012) can be also employed for identifying somatic mutations. SomaticSniper statistically compares the genotype likelihoods of cancer and control samples, which could turn out useful for cancer-related studies (Larson *et al.*, 2012). Tools such as CNVnator (Abyzov *et al.*, 2011) and ExomeCNV (Sathirapongsasuti *et al.*, 2011) were mainly developed for identifying CNVs. CNVnator combines the mean-shift approach with multiple-bandwidth partitioning and GC correction for detecting atypical CNVs like *de novo* and multiallelic events. However, it often misses CNVs generated by retrotransposable elements (Abyzov *et al.*, 2011). On the other hand, ExomeCNV was specifically created for identifying loss of heterozygosity in addition to CNVs. This software is able to correctly discover small indels due to the inclusion of B-allele frequencies and depth-of-coverage (Sathirapongsasuti *et al.*, 2011). For detecting SVs such as inversions, translocations, or large indels, the software CLEVER is still widely utilized because of its efficient use of internal segment sizes (Marschall *et al.*, 2012).

According to one study that revised several pipelines for whole exome analysis, the combined use of the aligner BWA and SAMtools, or of any aligner with Freebayes, turns out suitable for SNP calling. This study also claims the software GATK (HaplotypeCaller) as the best variant caller for indels, and Freebayes as the most convenient tool when low-quality variants are previously filtered out (Hwang *et al.*, 2015).

Variant annotation

After identifying variants, the detected variants are annotated to understand their biological implications. This step aims to filter out irrelevant variants and retrieve those disease-related ones. Although the current tools are able to annotate SNPs and indels, the annotation of CNVs has not been fully implemented. The available software includes ANNOVAR (Wang *et al.*, 2010), Vcfanno (Pedersen *et al.*, 2016), and VarMatch (Sun and Medvedev, 2017). ANNOVAR can be utilized for annotating SNVs and indels. It analyzes the function of those genes harboring variants, detects variants in conserved genomic stretches, and predicts cytogenetic bands (Wang *et al.*, 2010). Vcfanno includes a "chromosome sweeping" approach, making possible the annotation of a large number of variants (Pedersen *et al.*, 2016). VarMatch can be used when dealing with dense areas of variants or low complexity genomic regions. It implements an optimization strategy known as edit distance, which contributes to improving robustness (Sun and Medvedev, 2017). Furthermore, the detected variants should be matched against those in mutation databases such as dbSNP (Sherry *et al.*, 2001) or COSMIC (Forbes *et al.*, 2008). Consequently, the variants could be classified as deleterious or accepted variants.

Visualization

Variant visualization is also an essential step because it offers better interpretations of the experiment. There are many genome browsers that facilitate the visualization of genomic regions. Some are useful for visualization and annotation, for example, ABrowse (Kong *et al.*, 2012), Apollo (Lee *et al.*, 2013), Ensembl (Ruffier *et al.*, 2017), and GenomeView (Abeel *et al.*, 2012). Others are specifically designed for visualization, including BamView (Carver *et al.*, 2013), JBrowse (Buels *et al.*, 2016), and MapView (Bao *et al.*, 2009). An extensive review of visualization tools can be found in Pabinger *et al.* (2014).

Transcriptome Analysis

Alterations at the genome level, which can be detected using DNA-sequencing approaches such as WGS, are responsible for a wide range of adverse health phenotypes. Such diseases can often be linked to a single mutation within a crucial gene or accumulation of changes that, when combined, can lead to cancer or age-related illnesses (Podolskiy *et al.*, 2016). However, most complex biological mechanisms arise through the interplay of hundreds of interactome elements, with the disease often being a result of perturbation to those systems and networks (Vidal *et al.*, 2011). The most accessible genomic layer, in terms of easiness of profiling and basic examination, is the transcriptome. A global view of gene activity, which takes place at the RNA level, provides means to perform a wide variety of studies. This includes analysis of cellular functions, identification of predictive biomarkers and diseases subtypes, or pathway and network-oriented studies that can more comprehensively explain changes in the state of biological systems (Brodie *et al.*, 2014). Here we will present several computational approaches to study the transcriptome, and avenues to explore when linking such profiles to actionable results. Emphasis will be placed on popular, open-source tools and libraries and on the fundamentals, which can be easily built upon and extended.

Profiling Platforms

The two most widespread platforms for studying global gene changes are NGS and DNA microarrays. The former, a high-throughput approach that superseded Sanger sequencing, relies on "reading" millions of small RNA fragments in parallel. As each RNA transcript is represented by multiple overlapping fragments, it is possible to combine the sequence pieces together by mapping them to a specie-specific reference (Behjati and Tarpey, 2013). DNA microarray (or DNA-chip) technology, on the other hand, uses a collection of DNA probes attached to a solid surface, with each oligonucleotide probe cluster designed to match a specific short region from a gene of interest. Binding a complementary sequence produces a signal (generated by a fluorophore-labeled antibody), whose strength is used to quantify the abundance of the corresponding target gene (Shalon *et al.*, 1996). While the choice between platforms is often based on personal preference and technical capabilities available within a given group, there are several factors that should be considered when making the decision. Microarrays contain a set of predefined target genes, which makes the data processing simple and straightforward. The methodology has matured to a point where the process can be accomplished using a single package – an overview using a representative tool is provided in Section "DNA-Chip Data Processing." While convenient, the technology does not offer as much flexibility as NGS alternatives (Hurd and Nelson, 2009). If a project would benefit from detecting novel or rare transcripts, fusion genes, or splice variant analysis, or requires higher confidence in the findings (which can be achieved by increasing the sequencing depth of coverage), NGS methods offer an advantage. The analysis of RNA-seq data does, however, require some understanding of the process on the molecular level, as well as familiarity with additional software tools. Each step of the process is described in Section "RNA-Seq Data Processing," and a graphical overview of the whole transcriptome section is shown in **Fig. 3**.

DNA-Chip Data Processing

In this section, we will provide a basic overview of initial DNA microarray data processing using limma package (Ritchie *et al.*, 2015) within R software environment. While these steps can be accomplished using manufacturer provided software solutions, limma offers more built-in functionalities and can be easily integrated with other bioinformatics tools. Furthermore, a similar methodology can be utilized for other types of arrays, such as those profiling protein expression levels. R is a statistical computing environment that is widely used for exploratory data analytics and visualization. It can be installed on any operating systems by following the instructions found on the project's website (see "Relevant Websites" section). Subsequently, a wide range of genomic packages can be downloaded from Bioconductor, a repository for high-throughput genomic analysis tools (see "Relevant Websites" section).

The initial chip processing step is automatically performed by the image analysis software, where the TIFF scan of the array is translated into a list of intensities for each of the small "spots" (i.e., areas with multiple copies of a unique DNA probe). limma provides a read function that loads and converts data (together with sample annotation) from a wide range of microarray formats. Probe annotation, usually stored in GenePix Array List (GAL) format, should be provided separately, in case it is not included in the output files.

Issues with the sample quality can be introduced at any stage of preparation, from low-quality biological material to problems during scanning. It is thus important to detect such problems early during analysis. There are two common quality control

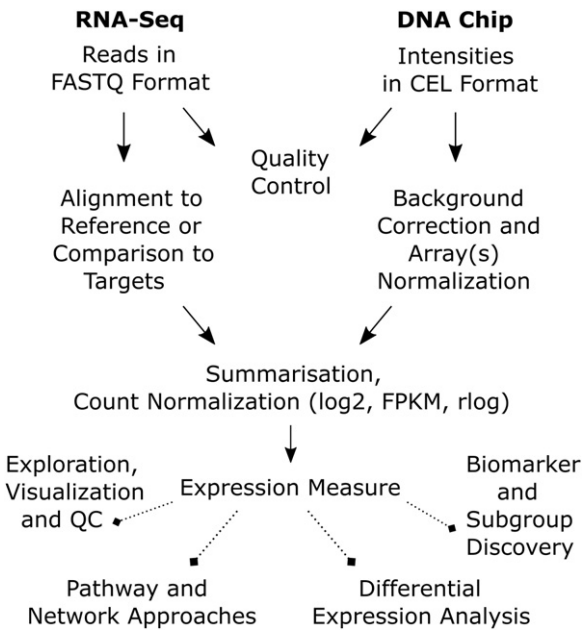

Fig. 3 Graphical overview of the standard pipeline for transcriptome data analysis.

strategies that are applicable to both individual spots within arrays (much higher complexity) and to cross-array comparison – assigning quality weights that are used in a downstream analysis or removing problematic cases (Kauffmann and Huber, 2010). limma has built-in functions to cover the former, which calculate spot weight quality as well as array variance (Ritchie *et al.*, 2006). Lower scores are assigned to spots with quality issues and to arrays that are less reproducible, ensuring that results are not affected by bad samples. Alternatively, arrayQualityMetrics package (Kauffmann, 2009) can be used to calculate and visualize the distance between arrays and exclude those that are significantly different from the rest.

When measuring the fluorescence intensities there are several factors that affect the scan, by introducing ambient noise, both within and between the arrays (Silver *et al.*, 2009). Such signals are adjusted with background correction algorithms, which estimate and remove nonspecific binding or spatial heterogeneity effects (Ritchie *et al.*, 2007). Plotting the foreground against the background intensities can guide the choice of optimal correction algorithm. Lastly, the samples should be normalized to ensure that the biological signal is not "diluted" by technical variation or biases. Depending on the type (single- or two-color) and technology of the array, limma provides several approaches to normalize (i.e., remove some of the variations) within and between the arrays. This ensures that all samples are on the same measuring scale and that technical differences are not influencing the results. The choice of algorithms for background correction and normalization is the only challenging step in the processing of DNA microarrays, and both should be guided by the assessment of the data. As the subsequent steps in the analysis are common between sequencing and DNA-chip platforms, they will be described together in Section "Differentially Expressed Genes and Visualization."

RNA-Sequencing Data Processing

The field of NGS profiling encompasses a wide range of protocols, though from the analysis standpoint the main difference lies in the type of sequence that is enriched during library preparation. These can include small regulatory RNA such as miRNA, total or coding RNA, degraded RNA (German *et al.*, 2008), or RNA amplified from a single cell rather than a population of cells (Eberwine *et al.*, 2014). While there are differences in how each of the types is analyzed, the initial data processing steps and general principles are similar. The section below will focus on processing data from total or polyA selected RNAs, though with small changes to the reference the methodology can be applied to other types of protocols. For the purpose of this section, we will presume that the sequencing was performed on Illumina's HiSeq or MiSeq machine, and that demultiplexing (the process of separating individual libraries that were sequenced on the same lane) was performed during data export to FASTQ files. FASTX-Toolkit (Hannon, 2010) and EMBOSS seqret (Cock *et al.*, 2010) are two useful tools for checking, manipulating, and converting FASTQs to match the input format requirements of other programs.

The quality of reads, or short sequences obtained at the end of NGS protocols, depends on the integrity of the initial sample, proficiency in library construction, and handling during final sequencing stage. FastQC (Babraham-Bioinformatics, 2018) is a quality control tool that generates comprehensive graphical reports on several important characteristics of any sequence data. It should be noted that modern alignment algorithms can deal with cases such as adapter (short DNA sequences used in library preparation and sequencing) contamination or when part of the reads are of lower quality. However, in more severe cases,

trimming the sequences or removing the adapters can increase the quality and number of aligned reads. This can be achieved using tools such as Cutadapt (Martin, 2012), FASTX (Hannon, 2010) or Trimmomatic (Bolger *et al.*, 2014). It is important to repeat the quality control step after each manipulation to ensure that the overall quality of the library has improved. If the entire length of the reads is of very low quality (below the score of 20 for Illumina's version 1.3 or later encoding), adapters make up most of the reads, or there is a significant overexpression of few "background" genes (such as ribosomal RNAs), it is usually advisable to repeat the sample and/or library preparation. The aims of the project and type of data will naturally influence the software stack and parameters used in the data processing. Differential splicing and metatranscriptomics assembly require a more complex setup, though the principles are the same once the fundamentals are understood. For the purpose of this and the next three subsections, we will focus on transcriptome profile comparison between two phenotypes or conditions.

The RNA-Seq reads are small sections of the transcriptome, hence the traditional way of processing them was to align the reads to a reference genome or transcriptome. From there, the reads were "stitched" together, and the overlap with regions of interest was used for abundance estimation. Abundance can be calculated for exons, transcripts, or genes, though gene-level analysis is generally recommended when transcript information is not required. Transcript quantification is more challenging due to a combination of short sequencing length and lack of unique sequences in many splice variants. Increasing the read length through gradual improvements in the technology will solve this problem in the near future. A prime example of the "traditional" approach is STAR aligner (Dobin *et al.*, 2013), where the genome index created prior to mapping is used during alignment. STAR gained widespread adaptation due to high-performance relative to other methods available at its launch. After the alignment is completed, programs such as featureCounts (Liao *et al.*, 2014) are run to count the mapped reads to genomic features, and provide a summary of the alignment. The postalignment report contains information on the percentage of mapped, unique, and duplicated reads, and serves as a final quality check before downstream analysis. A more detailed overview of the alignment can be generated using RNA-SeQC (DeLuca *et al.*, 2012). A recently developed alternative to this software stack is the so-called "pseudoaligners," which instead of aligning to the reference, determine read compatibility with defined targets (Bray *et al.*, 2016). kallisto is an example of such approach, and the algorithm has not only been shown to be very fast but also accurate and more robust to errors (Bray *et al.*, 2016). The transcript level abundance estimates produced by kallisto can be converted to counts using the R package tximport (Soneson *et al.*, 2015), after which they can be analyzed in the same fashion as any other alignment result. Lastly, and similarly to DNA-chip data, the reads should be normalized so that a comparison can be made between samples from different lanes or sequencing runs. Fragments per kilobase of transcript per million mapped reads (FPKM) is a popular expression unit that normalizes for sequencing depth and corresponding gene length. A similar type of normalization can be achieved in tximport using the lengthScaledTPM parameter, which is the final step in the processing of raw sequencing reads.

Differentially Expressed Genes and Visualization

The downstream analysis of normalized counts varies greatly depending on the experimental design, type of data and sample characteristics. While biological replicates and appropriate controls are crucial in many studies, in settings such as clinical bioinformatics, the lack of controls or a low number of samples is still common. Whenever paired samples are available (e.g., control and treated, or normal and diseased tissue from the same patient), differential expression analysis is usually performed first. It is a quantitative measure of expression change that can be used to rank genes based on a degree of under- or overexpression. It is a good practice to remove lowly expressed genes beforehand, which can be accomplished either using sophisticated software models or more simply by removing genes with a count ≤ 10.

Fold change, or the magnitude of change between conditions, is a preferred method for quantification of differential expression. It can be calculated using a variety of packages such as limma, DESeq2 (Love *et al.*, 2014), or sleuth (Pimentel *et al.*, 2017), all of which offer technically advanced though slightly different implementations. For a simpler manual calculation, the genes should first be transformed to account for the presence of extreme values and mean-variance dependency. This is most commonly achieved using log2 transformation, though alternatives such as rlog or VST are also used (Lin *et al.*, 2008). Subsequently, the log2(mean) of all replicates from a healthy control/untreated samples are subtracted from the log2(mean) of all replicates from a disease phenotype/treated samples. For datasets with no paired samples, a comparison can be made between groups with and without a certain condition or phenotype.

Visualization is an important component in bioinformatics as it is challenging to identify patterns in tens of thousands of genes based on text data alone. The most common graphics libraries used for high-quality plotting are ggplot2 in R (Ginestet, 2011) and matplotlib in Python 3 (Hunter, 2007). The built-in statistics and graphics capabilities in R are usually sufficient for initial data exploration purposes. Bioinformatics investigation is often initiated with a thorough check on how closely the samples match the experimental design. This is performed to identify population structures in the data that are previously unknown, or subgroups of samples with different characteristics. Furthermore, outliers, or samples with abnormally low or high values (which could represent technical issues), can be identified and removed. Principal component analysis (PCA) helps to find such characteristics by obtaining a combination of variables, in this case, gene expression values, that best differentiate the samples (Abraham and Inouye, 2014; Sharma and Paliwal, 2007). Any aberrant samples should be removed at this stage so that they do not influence the results. Subsequently, two-group expression comparison can be visualized using MA plots, which show the relationship between logged fold change and mean expression. Such plots are equally suited for diagnosing data-related issues as well as checking the number, magnitude, and pattern of differentially expressed genes. Another way of exploring global changes in gene expression is

the Kolmogorov-Smirnov nonparametric test. It compares the distribution of each gene between two defined groups and is a good approach to visualize changes in mean expression and variability. The above-mentioned plots show general patterns of expression and should be supplemented with more direct visualizations. A very popular and information-rich example is heatmaps, which are two-dimensional grids where a range of values is represented by colors. They are often paired with hierarchical clustering, which sorts rows (which represent gene expression) and columns (which represent samples) in a way that most clearly shows patterns in the data. The combination can reveal commonly regulated genes or sets of samples with distinct expression profiles and characteristics (such as disease subtypes). Another way of using heatmaps is correlation matrices, which show a correlation of expression between multiple genes. More advanced plotting methods often contain a higher density of information within a single figure, utilizing colors, shapes, and opacity in a third-dimensional space.

Functional, Pathway, and Network Analysis

While global profiling of gene (or protein) expression provides a significant amount of information, it is often challenging to link such results to tangible biological insights. Employing annotation regarding a gene's (or its product's) involvement in a certain biological process, signaling pathway, or molecular function is one of the most effective approaches to study RNA-Seq data. Gene set enrichment analysis is one of the most widely used solutions, which verifies if the differentially expressed gene results show an enrichment or depletion of genes associated with a certain biological aspect (Subramanian *et al.*, 2005). This can include genes connected with cancer development, drug response, a wide range of signaling pathways, and more (Curtis *et al.*, 2005). There are numerous approaches for performing gene set analysis (e.g., overrepresentation analysis, pathway topology), most of which are implemented in R or are available as online tools (Tarca *et al.*, 2013). Pathway information can be easily retrieved from online databases such as KEGG (Kanehisa *et al.*, 2016), Reactome (Fabregat *et al.*, 2016), or WikiPathways (Slenter *et al.*, 2018), either manually or using the associated application programming interfaces. A different approach makes use of topological properties of genes within molecular interaction networks. Based on the direct interaction partners, or close proximity neighbors, it is possible to infer novel insights such as biological function (Sharan *et al.*, 2007) or prioritize candidate disease genes (Yu *et al.*, 2013). Cytoscape is the most popular environment for network studies (Shannon *et al.*, 2003), though it is also possible to perform a wide variety of network analyses using igraph in R (Csardi and Nepusz, 2006) or networkx in Python 3 (Hagberg *et al.*, 2008). Weighted gene coexpression network analysis is a great example of a network-based exploratory tool that provides additional data reduction and feature selection capabilities (Langfelder and Horvath, 2008). It is based on correlation patterns between gene expression, which forms the basis for finding clusters of potentially associated genes. It also identifies relationships between multiple such modules. Network approaches are also very useful for integrating different types of data, such as multiomics profiles (e.g., mutation, expression, or methylation datasets) or electronic medical records. Similarity network fusion is a popular example – it constructs a network of samples for each data layer and fuses them into a combined structure (Wang *et al.*, 2014). This approach strengthens strong signals (i.e., similarities between layers) and allows the samples to be grouped into distinctive subtypes.

Machine Learning Approaches in Genomics

The unprecedented growth in volume and complexity of biomedical datasets and literature has made bioinformatics analyses more challenging than ever. The very high ratio of features (e.g., mutation or expression) to the number of samples, and the considerable time investment needed to achieve expert-level knowledge on a particular biological phenomenon translate into the additional level of complexity. This has led to growing popularity of ML approaches, algorithms designed to automatically learn from data without an explicit set of rules to follow. While genomic datasets pose many unique challenges that need to be considered, ML algorithms can be extremely useful, particularly when combined with standard bioinformatics tools and databases. There are three main types of algorithms that are of direct use to genomic studies – unsupervised, supervised, and semisupervised. They are all easily accessible through Python, R, or Java libraries, although it requires familiarity with statistical analysis and programming.

Unsupervised learning is geared towards inferring "hidden structures" in unlabeled data – labels being the output values that in biological context can mean disease subtypes, months of patient survival or type of response to a treatment. Given the previously mentioned high number of features, dimensionality reduction is the first area of interest. PCA has traditionally been used to reduce the dimensionality of the data by geometrically projecting them onto lower dimensions (Lever *et al.*, 2017). This transformation allows for noise removal and easier classification, though recently developed t-distributed stochastic neighbor embedding was shown to outperform it in some biological applications (Fonville *et al.*, 2013). Such methods can be combined with pathway annotation to better explain variations in the phenotype (Ma and Kosorok, 2009). One of the primary aims of personalized medicine efforts is finding subgroups of patients with distinct phenotypes, which would allow for a more tailored prognosis, care, and treatment. To this end, a plethora of clustering approaches are routinely used on genomic data, mainly in hopes of finding novel disease subtypes (Van Rooden *et al.*, 2010). Hierarchical clustering, k-means, and spectral clustering are popular algorithms, though many specialized methods have been created specifically for biomedical application (Sharma *et al.*, 2017a,b,c, 2016a).

When label data is available, supervised learning approaches can be used to identify significant features or build models to predict categorical classes (classification) or continuous values (regression). While the identification of biomarkers for

detecting and monitoring diseases has long been a focus in medicine, in many cases more complex models are needed to make an accurate diagnosis (Kidd *et al.*, 2015). Overfitting is a common challenge when building prediction models – given the high number of features in genomic datasets, it is relatively easy to find a combination of genes that will correlate with outcome labels by pure chance. Such results will not generalize well when tested with an independent dataset. It is thus important to separate the data into training and testing datasets (a popular split is 70%–30%), and only use the latter for sporadic validation of the model. Models with fewer features and lower complexity are always preferred, especially if the features are already known to be predictive of the outcome of interest, or are connected to a biologically relevant mechanism or pathway. The baseline prediction performance can be established using simple annotation (patient information, sample characteristics) or using biomarkers already reported in the literature. Such model can be further expanded or replaced entirely depending on its performance and complexity. Subsequently, feature selection approaches (Sharma *et al.*, 2012a,b,c,d, 2014, 2011; Sharma and Paliwal, 2011) can be applied to find biological components that best explain the phenotype of interest. Simple univariate methods such as t-test or Pearson's correlation were shown to be the optimal methods for feature selection in genomic data (Haury *et al.*, 2011). Alternatively, tree-based algorithms or recursive feature elimination paired with an estimator are popular choices as they allow to rank features based on their importance to the model. Once a subset of features is selected, a model can be constructed using a variety of algorithms – LASSO, ElasticNet, and random forest regressor are popular choices for predicting quantity while support vector machine (SVM), random forest classifier, or k-nearest neighbors (kNN) are popular for predicting categories. More advanced strategies such as boosting, bagging, or stacking can be used to decrease variance or increase accuracy – xgboost, being a representative example, usually provides satisfactory out-of-the-box (i.e., using default parameters) results (Chen and Guestrin, 2016). Neural networks, while popular in many fields such as medical image recognition, are used more sporadically in biomedical applications where decision-making transparency is often crucial. An iterative process of constructing, tuning (i.e., tweaking parameters), and testing the model is usually performed on the training data, using a 10-fold cross-validation (CV) technique. The process partitions the data into 10 equal parts and uses nine parts for training and one for evaluation. This is repeated 10 times, each time using different parts of the data for training and testing, and the final result is calculated as an average from all the runs. The performance metric depends on the type of the label, with accuracy, precision, and recall being popular for classification and root mean squared error and coefficient of determination (R^2) for regression. Once satisfactory results are achieved, the model should be trained on the entire training data (70% in our example) and tested on the remaining test data (the leftover 30%). The performance of a good model should show comparable or better results relative to the 10-fold CV evaluation. While some improvements can be achieved using different ML algorithms or by tweaking the parameters, poor performance usually means that the selected features are not enough to predict the outcome. Feature engineering, or creation of novel features using domain knowledge, is often the best approach to improve the model (Ang *et al.*, 2016). A more recent development in the ML field is semisupervised algorithms, which take advantage of the information contained within unlabeled data in classification problems (Scholkopf and Zien, 2006). Mixing unlabeled and labeled data has been shown to improve the supervised learning accuracy and is of particular interest to biology, where unlabeled data is abundant but labeled data is difficult to obtain.

Workflows in Proteomics

Proteomics is an extensive field of research where a plethora of studies are being conducted. These studies cover the following areas: protein fold recognition, structural class prediction, function analysis, posttranslational modification, and molecular recognition of features (Dehzangi *et al.*, 2018, 2017; López *et al.*, 2018, 2017; Chou, 2017). Various bioinformatics tools and packages have thus been developed for advancing the above research areas. The implementation of modern pipeline frameworks requires detailed guidelines and often offers a command line or workbench interface. In this section, we survey on the pipeline frameworks for identifying molecular recognition features (MoRFs) in IDPs. Specifically, we emphasize the design strategy of MoRF prediction systems and provide practical recommendations based on the requirements of high-throughput bioinformatics analyses.

The Necessity of Molecular Recognition Feature Prediction

In the traditional view of protein structure–function paradigm, proteins fold into a stable three-dimensional structure that ultimately determines their function. However, recent progress in computational and experimental methods has revealed a lack of this stable tertiary structure in certain protein regions (Dyson and Wright, 2005, 2015; Lee *et al.*, 2014; Uversky, 2014). Proteins comprising such regions are called IDPs and reportedly have important biological functions related to signal transduction and cell regulation (Wright and Dyson, 2015; Lee *et al.*, 2014). IDPs often achieve their functions through the loosely structured MoRF region. This region binds to a structured partner and consequently undergoes a disorder-to-order transition to adopt a well-defined conformation (Lee *et al.*, 2014). MoRF length varies up to 70 amino acids. Although there are many documented experimental techniques for identification and analysis of MoRF regions, they are still expensive and time-consuming. To overcome this

challenge, computational approaches have become absolutely necessary for predicting these MoRFs in disordered protein sequences (Sharma *et al.*, 2018a,b, 2016b; Malhis *et al.*, 2016; Disfani *et al.*, 2012; Dosztányi *et al.*, 2009).

State-of-the-Art Molecular Recognition Feature Predictors

An abundance of computational methods and predictors have been already developed to predict MoRFs. These include ANCHOR (Dosztányi *et al.*, 2009), MoRFpred (Disfani *et al.*, 2012), MoRFchibi (Malhis and Gsponer, 2015), MoRFpred-plus (Sharma *et al.*, 2018a), MoRFchibi-web (Malhis *et al.*, 2016), PROMIS, and OPAL (Sharma *et al.*, 2018b). OPAL, the latest proposed predictor, has also proven the most accurate predictor in the literature. The above-mentioned predictors are currently available as online web servers and downloadable software packages. The subsequent sections illustrate the guidelines of MoRFs prediction and how to evaluate the pipeline of MoRF prediction.

Fundamentals of Protein Sequence Analysis

Sequence

The protein sequence is retrieved from an experimental setup where all the amino acids of the protein or peptide are determined. For computational analyses, a large number of protein sequences are at present available in different databanks, which were generated and annotated through high-throughput technologies.

Feature source and feature extraction technique

The features of protein sequences can be captured from different sources of information. These can be chemical, physical and physicochemical properties of amino acids, evolutionary and structural information of protein sequences, functional domains, and gene ontology information of protein sequences.

Some of the existing feature extraction techniques have been used to create feature vectors from protein attributes such as occurrence, composition, transition and distribution, pairwise frequencies, amino acid composition, autocorrelation, and bigram (Sharma *et al.*, 2013, 2015; Liu *et al.*, 2015; Dehzangi *et al.*, 2015; Du *et al.*, 2014). Early studies used to focus on sequential, syntactical, and physicochemical features. However, more recent studies have considered evolutionary-based features because they tend to provide good prediction accuracy for protein-related problems (Sharma *et al.*, 2018a; Yang *et al.*, 2017; Lyons *et al.*, 2016; Dehzangi *et al.*, 2015).

Classifiers

For classification purposes, several classifiers have been widely explored, including SVMs, artificial neural networks, kNN, as well as ensembles of classifiers. Amongst the aforementioned classifiers, SVMs is a well-known classifier that has delivered promising results in recent studies (Sharma *et al.*, 2018b; Malhis and Gsponer, 2015).

Design of a Molecular Recognition Feature Predictor

This section will cover the benchmark dataset and the framework of MoRF prediction.

Benchmark dataset

A wide collection of benchmark databases has been introduced, assembled, and used for developing MoRF predictors. These repositories include MoRFpred, MoRFchibi, MoRFpred-plus, MoRFchibi-web, and OPAL (**Table 1**).

The sequences in TRAIN and TEST464 sets were obtained from Protein Data Bank and filtered to retain protein-peptide regions of 5 to 25 residues in size (Disfani *et al.*, 2012). The sequences in TEST464 share a less than 30% sequence identity with sequences in TRAIN (Disfani *et al.*, 2012). MoRF predictors are always trained using the sequences in TRAIN set and assessed using the sequences of TEST464 set (Sharma *et al.*, 2018a,b; Malhis and Gsponer, 2015; Malhis *et al.*, 2016). According to the principles used to assemble TRAIN and TEST464 sequences, it is not verified whether identified protein-peptides are disordered in isolation and also TEST464 is not free of redundancy as its sequences share more than 30% sequence identity to each other (Disfani *et al.*, 2012). In order to address the above and validate a MoRF predictor, the EXP53 set is often used. The EXP53 set contains 53 nonredundant protein sequences with MoRFs which have been experimentally verified to be disordered in isolation (Malhis *et al.*, 2016; Sharma *et al.*, 2018b).

Table 1 Description of OPAL benchmark dataset

Datasets		*No. of sequences*	*Total residues*	*No. of molecular recognition features (MoRF) residues*	*No. of non-MoRF residues*
Training set	TRAIN	421	245,984	5396	240,588
Test sets	TEST464	464	296,362	5779	290,583
	EXP53	53	25,186	2432	22,754

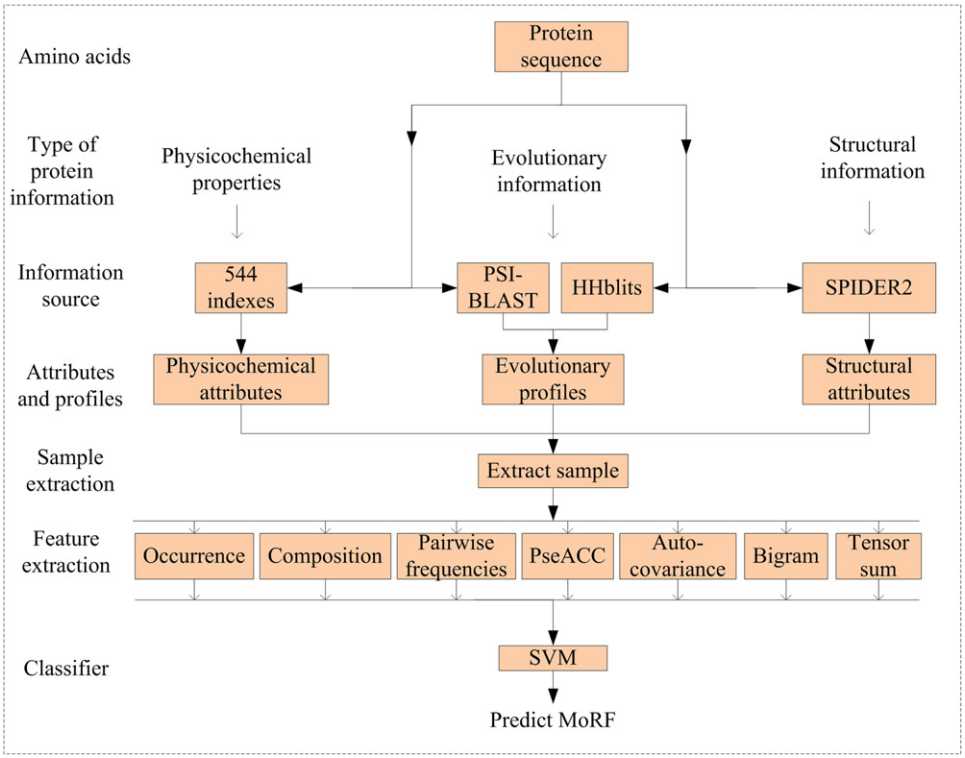

Fig. 4 Framework of molecular recognition feature (MoRF) prediction.

Framework of Molecular Recognition Feature Prediction

Fig. 4 shows the framework of MoRF prediction, whose input is the protein sequence. For prediction purposes, protein sequences are commonly represented using different features. These characteristics comprise physicochemical properties of amino acids such as 544 common physicochemical indexes (Kawashima *et al.*, 2008), evolutionary profiles such as position specific scoring matrix generated with PSI-BLAST (Altschul *et al.*, 1997) and hidden Markov model profiles generated with HHblits (Remmert *et al.*, 2011), as well as structural attributes such as secondary structure, accessible surface area and dihedral torsion angles predicted by tools as SPIDER2 (Yang *et al.*, 2017). To predict each residue in the protein as MoRF and non-MoRF, a sample is first extracted, described as a feature vector, and finally predicted with a suitable classifier.

Fig. 5(A) outlines the bioinformatics tools/scripts and the data format required for implementing a framework of MoRF prediction. These tools are generally run on a Linux operating system and protein sequences are organized in a FASTA format file. Tools such as PSI-BLAST, HHblits, and SPIDER2 can be installed and run via command line. On the other hand, scripts are written with software tools such as MATLAB, OCTAVE, and C-code. Finally, Fig. 5(B) shows the procedure of training and scoring a MoRF predictor.

Implementation Steps of a Molecular Recognition Feature Predictor

To identify MoRFs in a protein sequence, each residue should be initially scored. This score will help predict the residues as MoRF or non-MoRF. Two of the methods often used in a MoRF classification scheme are ResidueMoRF and RegionMoRF. These methods are used to extract samples from protein sequences during the training and test steps. The following subsections will discuss the training and test steps, which are of utmost importance to construct a MoRF predictor.

Training step

In this step, features are extracted from MoRFs and non-MoRFs and grouped in a training set. Let us assume the segment F, which represents a MoRF with flanks, and the segment G, which describes a non-MoRF with flanks. Both segments F and G are from a protein sequence in the training set. Let us also suppose that a protein sequence P_i is given as

$$P_i = A_1 A_2 \ldots A_j \ldots A_{n_i} (i = 1, 2, \ldots, T) \quad (1)$$

where A_j is the jth amino acid in the sequence, T is the total number of protein sequences in the training set, and n_i is the length of the protein sequence P_i. For instance, let us consider a protein sequence P_1 whose length $n_1 = 150$ and which contains one MoRF located between A_{30} and A_{40}. The segments F and G for this protein P_1 are illustrated in Fig. 6. For illustration purposes, a flank size of 20 amino acids has been considered. Of note, the flank size should be selected during training and evaluation.

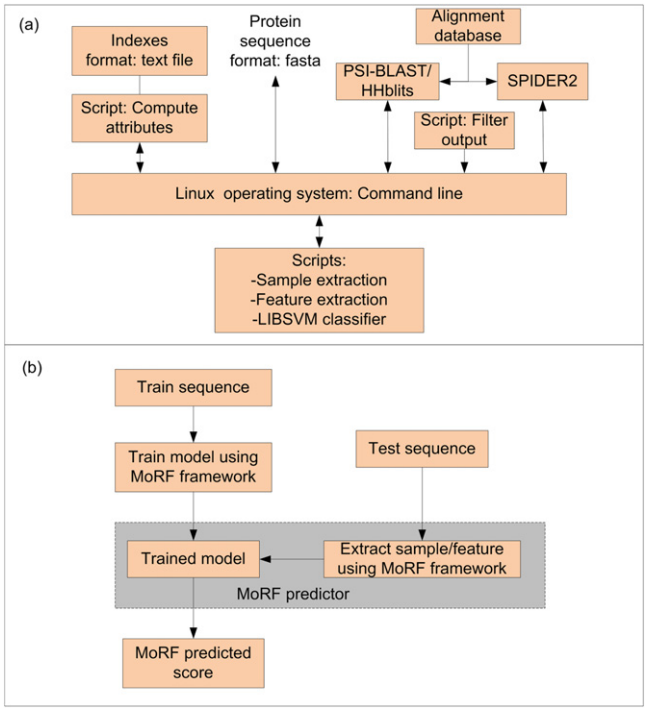

Fig. 5 Pipeline of molecular recognition feature (MoRF) predictor implementation. (A) details of bioinformatics tools/scripts and data formats, (B) procedure of training and testing a MoRF predictor.

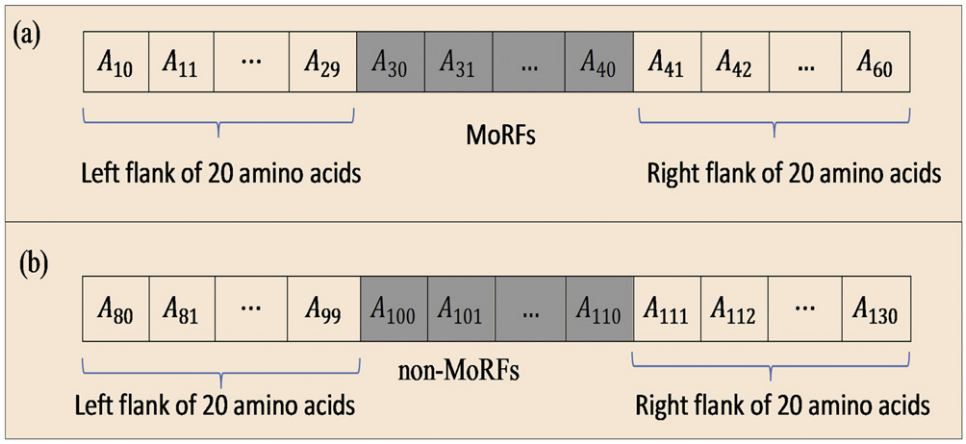

Fig. 6 Schematic illustration of (a) segment *F* and (b) segment *G*.

MoRFs can have a variable length of 5 to 25 amino acids. Additionally, zeros are padded to create flanks of 20 amino acids if the MoRF is present at the beginning or end of the protein sequence. Positive and negative samples representing MoRFs and non-MoRFs are extracted from segments *F* and *G*, respectively.

For sample extraction, two methods named ResidueMoRF and RegionMoRF, which are detailed below, are commonly used.

ResidueMoRF method

With this method, a window size of 41 ((flank size $\times$ 2) + 1) amino acids is regarded for extracting samples from a segment. For extracting positive samples, the center of the window is placed on the MoRF residue of segment *F*, with amino acids upstream and downstream of the MoRF residue. For example, the sample S_α for a MoRF residue is defined as

$$S_\alpha = \{A_{\alpha-20}, \cdots\cdots A_{\alpha-2}, A_{\alpha-1}, A_\alpha, A_{\alpha+1}, A_{\alpha+2} \cdots\cdots, A_{\alpha+20}\} \quad (2)$$

where A_α is the MoRF residue (i.e., in **Fig. 7(A)**, $\propto = 30$). Positive samples extracted from a segment *F* using Eq. (2) are interpreted as

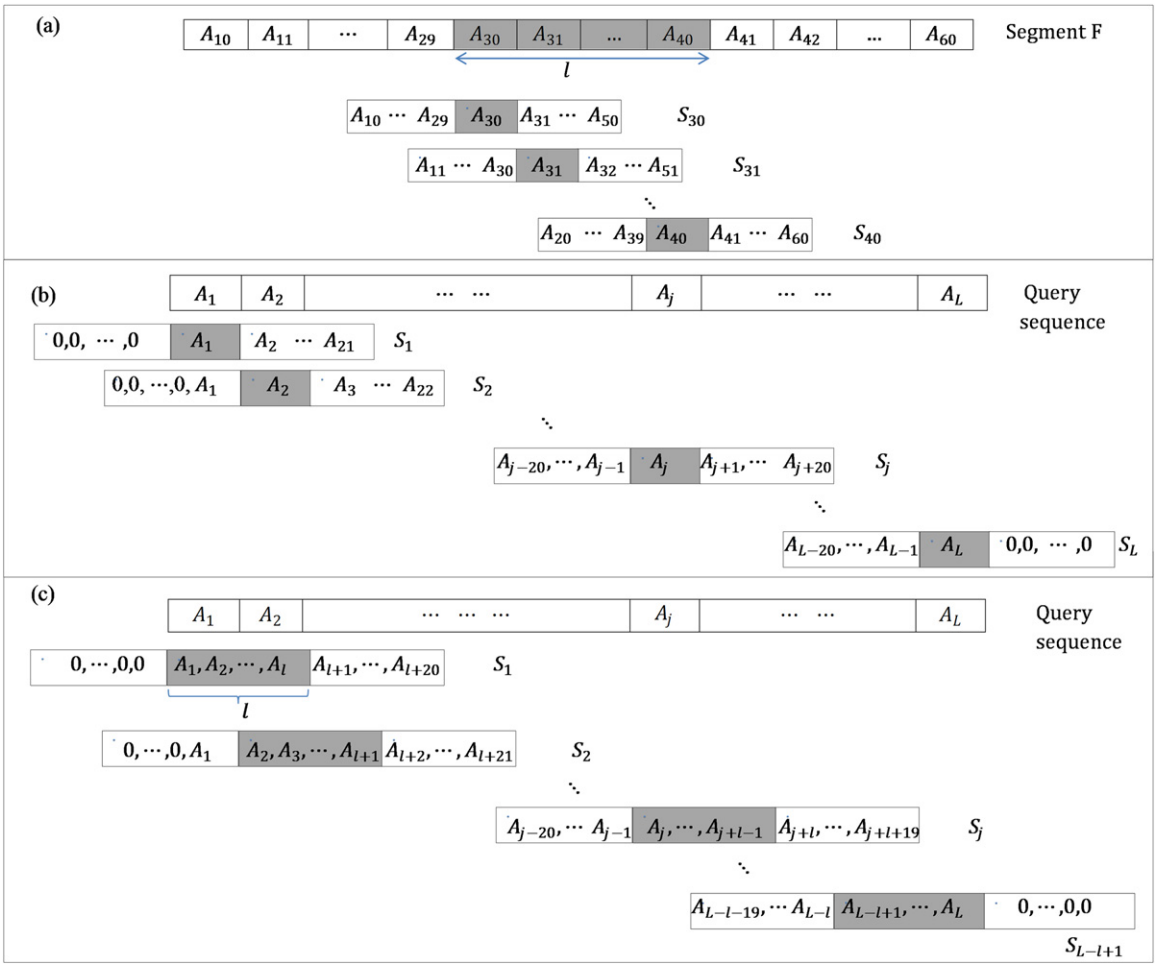

Fig. 7 Graphical illustration of (a) sample extraction from segment *F*, (b) sample extraction for scoring a query protein sequence of length *L* (A_j is the *j*th amino acid in the sequence), and (c) sample extraction from a query sequence of length *L*.

$$\gamma_{tr} = \begin{cases} S_{\alpha} \\ S_{\alpha+1} \\ \vdots \\ \vdots \\ \vdots \\ S_{\alpha+l-1} \end{cases} \quad \text{for } 5 \le l \le 25 \tag{3}$$

where l is the MoRF length (i.e., in Fig. 7(A), $l=11$). Fig. 7(A) shows the graphical illustration of sample extraction from the segment *F*. In a similar way, negative samples are extracted from segment *G*.

After applying either of the above models, the feature vector is finally computed from the sample and used for model training.

RegionMoRF method

With this method, the entire segment *F* with a MoRF region of length l (as depicted in Fig. 7(A)) is taken as a positive sample. Likewise, the entire segment *G* is taken as a negative sample.

Test step

The scoring of each residue in a query protein sequence also requires the extraction of samples. In this case, the sample extraction procedure makes use of the above methods ResidueMoRF and RegionMoRF as well.

ResidueMoRF method

Like the procedure described in the above section, a window size of 41 ((flank size $\times$ 2) + 1) amino acids is used to extract a sample for each query residue. The sample S_j represents a query residue defined as

$$S_j = \{A_{j-20}, \cdots\cdots\cdots A_{j-2}, A_{j-1}, A_j, A_{j+1}, A_{j+2} \cdots\cdots\cdots, A_{j+20}\} \tag{4}$$

where A_j is the residue in the query sequence and $j=1,2,\ldots,L$. Samples extracted using Eq. (4) for a query sequence of length L are described as

$$\gamma_{ts} = \begin{cases} S_1 \\ S_2 \\ \vdots \\ \vdots \\ \vdots \\ S_L \end{cases} \tag{5}$$

Fig. 7(B) shows the graphical illustration of sample extraction from a query sequence. Like the procedure followed for training, zeros are padded to make the length of flanks equal to 20 if query residues are at the beginning or end of the sequence. The feature vector is then computed from the sample and used for scoring.

RegionMoRF method

With this method, a window of size ($2 \times$ flank size $+ l$) is used to extract samples from a query sequence. The sample S_j for a query sequence is defined as

$$S_j = \{A_{j-20}, \cdots\cdots\cdots, A_{j-2}, A_{j-1}, A_j, A_{j+1}, A_{j+2}, \cdots\cdots\cdots, A_{j+(l-1)+20}\} \tag{6}$$

where A_j is the residue in the query sequence and $j=1,2,\ldots,L-l+1$. Samples extracted using Eq. (6) for a query sequence of length L are interpreted as

$$\gamma_{ts} = \begin{cases} S_1 \\ S_2 \\ \vdots \\ \vdots \\ S_{L-l+1} \end{cases} \tag{7}$$

where $l=6,7,\ldots,30$. A graphical illustration of sample extraction can be observed in Fig. 7(C). Eq. (8) defines the samples used for scoring each of the residues as

$$A_i \rightarrow \begin{cases} \{S_1, S_2, \cdots, S_i\}, 1 \le i < l \\ \{S_{i-l+1}, S_{i-l+2}, \cdots, S_i\}, l \le i \le L-l+1 \\ \{S_{i-l+1}, S_{i-l+2}, \cdots, S_{L-l+1}\}, i > L-l+1 \end{cases} \tag{8}$$

where $i=1,2,\ldots,L$. By varying l from 6 to 30, 450 $(6+7+\text{- - -}+29+30)$ samples are obtained for each residue, except for the residues at the beginning and end of the query sequence. The feature vector is computed from the sample and used for scoring. For each residue, the maximum score of the samples defined in Eq. (8) is set as final propensity score.

Summary of Molecular Recognition Feature Prediction

For the prediction of MoRFs in protein sequences, state-of-the-art predictors such as ANCHOR, MoRFpred, MoRchibi, MoRFchibi-web, and OPAL are currently available as web servers. Thus, end users or scientists can easily use a graphical web interface to predict MoRF and non-MoRF residues in a protein sequence. Except for MoRFpred, the other predictors can be further installed and configured locally (in personal computers). The performance of the above-mentioned predictors is shown in Table 2. As it clearly shows, the OPAL predictor outperforms the other approaches, achieving a significant accuracy and confirming its practical use in real scenarios.

To select a target protein for further experimental analysis, the sequences of large databases need to be scored. To do this, the end user can write SQL query scripts to obtain such scores from the MoRF predictor (Fig. 8(A)).

As Table 2 clearly indicates, the prediction accuracies are still limited when it comes to selecting a target protein for further experimental analysis for drug design. However, these limitations can be further improved by considering certain strategies. There is a need for high-performance workbenches, which could be accessible via the web to obtain protein sequence alignments. In the current MoRF web servers, the alignment tools are locally installed and run instead of using available alignment web servers. This is partly because alignment tools require a huge amount of time to provide alignment results. In addition, future MoRF pipelines similar to the one in Fig. 8(B) are expected. In other words, a core database server should store the experimental annotation from research labs whereas the computational annotation should be guaranteed by high-throughput bioinformatics analyses.

Table 2 Performance of state-of-the-art MoRF predictors. The results are reported for the TEST464 and EXP53 datasets using the area under the curve (AUC) as the performance metric

Predictor/method	*TEST464*	*EXP53*
ANCHOR	0.605	0.615
MoRFpred	0.675	0.620
MoRFchibi	0.743	0.712
PROMIS	0.790	0.818
MoRFchibi-web	0.805	0.797
OPAL	0.816	0.836

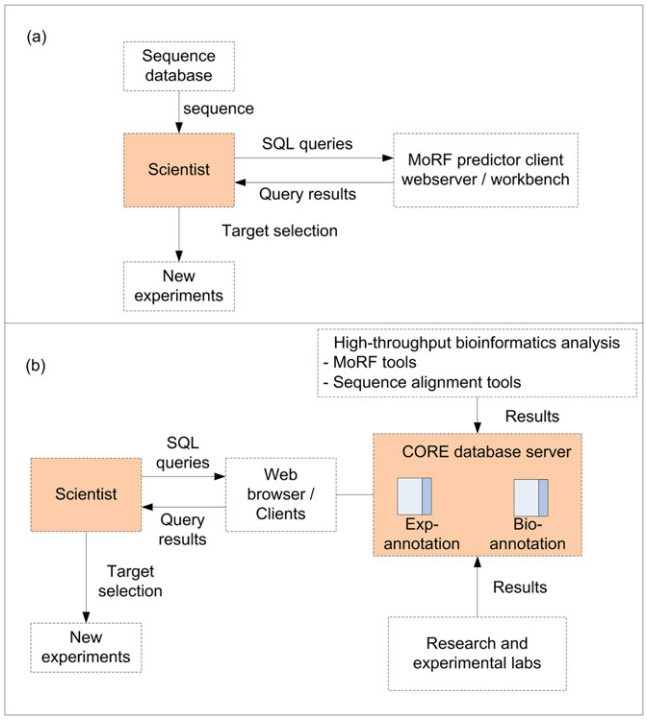

Fig. 8 Requirements for an improvement to molecular recognition feature (MoRF) prediction. (a) procedure of target protein selection for further experimentation, and (b) future pipeline for MoRF analysis.

Bioimage Analysis Pipelines

One of the areas in bioinformatics which is experiencing an enormous transformation is related to bioimage analysis. This is because a great number of bioinformatics applications are relying on image data for assessing different cellular and molecular mechanisms and thus validating the biological hypothesis. More and more images of biological mechanisms are generated by microscopes every single day, making necessary the design of accurate computational approaches.

In this direction, distinct imaging techniques such as PALM (Betzig *et al.*, 2006), STORM (Rust *et al.*, 2006), and STED (Hell, 2003) are currently able to detect individual proteins separated a few nanometers. Moreover, images related to biological mechanisms like RNA interference and chemical compounds (Echeverri and Perrimon, 2006; Moffat *et al.*, 2006; Sepp *et al.*, 2008) are absolutely necessary to shed light on scientific problems such as the differentiation of cancer cell phenotypes (Long *et al.*, 2007a).

However, the complexity of bioimages makes it extremely difficult to apply conventional medical image analysis workflows because the objects of interest sometimes possess different morphologies and intensities. Therefore, it is necessary to start paying attention to this emerging subfield of bioinformatics, which is not widely covered in the scientific literature.

The development of bioimage analysis pipelines often consists of the following steps (**Fig. 9**): preprocessing in which noise is reduced, image registration (Qu *et al.*, 2015) where images are aligned for pixel-by-pixel comparisons, image segmentation (Meijering, 2012) in which the objects of interest are separated from the background, feature extraction where object descriptors such as shape (Pincus and Theriot, 2007) and texture (Depeursinge *et al.*, 2014) are computed, classification where objects are detected by ML techniques (Shamir *et al.*, 2010), and visualization (Walter *et al.*, 2010).

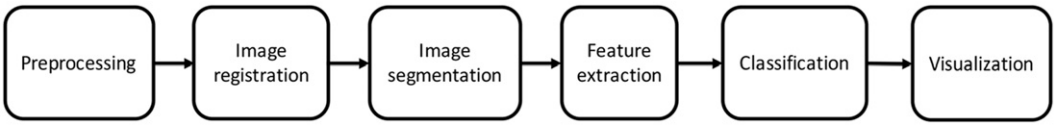

Fig. 9 Bioimage analysis workflow.

In this section, we will go through each of the above steps and explain the different methods.

Preprocessing

Preprocessing should be the first step of any pipeline. Its main objective is to improve the appearance of a bioimage or highlight additional information about the objects of interest, which might not be clearly observable. Many enhancement techniques and filters are often used to enhance the contrast and borders of the different objects, accentuate space frequencies, and remove or attenuate noise. One of these contrast enhancement techniques is histogram adjust, which produces a histogram with a specific form for the output image. Histogram equalization achieves good approximations towards a uniform distribution of the values of gray scale levels, giving the same occurrence probability to all gray levels (Pratt, 2014). To overcome the limitations of standard histogram equalization, the contrast-limited adaptive histogram equalization (CLAHE) technique is sometimes used. It divides the images into contextual regions and applies the histogram equalization to each of them (Pratt, 2014). This evens out the distribution of used gray values, making the hidden features of the image more visible. Another technique is the wavelet transform, which increases the image contrast and facilitates well-localized decomposition schemes (Salvado and Roque, 2005). In addition, digital filters are also used for enhancing bioimages by reducing noise and preserving the borders of those objects of interest. Some of these filters are Prewitt and Canny operators, as well as median, Gaussian, and average filters.

Image Registration

This step is often included in pipelines intended to compare images of different biological conditions such as those for building brain atlas (Ng *et al.*, 2007), and for comparing cell morphology and gene expression patterns. Here, methods such as mutual information registration (Viola and Wells, 1997), spline-based elastic registration (Rohr *et al.*, 2003), and congealing registration (Learned-Miller, 2006) can be utilized. Image registration has been used for blastoderm embryos in *D. Melanogaster* where each nucleus is described using a point in the 3D space (Fowlkes *et al.*, 2008).

Image Segmentation

The main aim of the segmentation step is to separate the objects of interest from the background. It is sometimes considered as a classification process of the objects present in the image because it can easily locate the different objects. Segmentation methods can vary depending on the specific application and image type. However, a segmentation method able to reach acceptable results for all types of bioimages has not been developed thus far. For a better explanation, we will present the segmentation methods in two groups: classical segmentation and clustering methods.

Classical segmentation methods

Thresholding is a method that segments scalar images, creating a binary partition of the image intensities. Once thresholding determines an intensity value, segmentation is achieved by grouping all the pixels based on intensity into two different classes. Its main limitation is that it only considers the intensity instead of other relationships between the pixels. One of the most reported methods is the Otsu's method, which selects the optimal threshold by maximizing the between-class variance with an exhaustive search. Although there are different methods to find a threshold, most of them are not satisfactory when analyzing real images, such as bioimages, due to the presence of noise, plain histograms, or an inadequate illumination.

Another classical method is deformable models, which are based on physical motivations, and used to delineate the borders of regions using curves or closed parametric surfaces deformed under the influence of external and internal forces. Two important deformable models are snakes (Kass *et al.*, 1988) which segments objects whose morphology is variable, and intelligent scissors (Liang *et al.*, 2006) which allows us to segment with minimal user interaction.

Clustering methods

Clustering techniques are also important in biomedical analysis pipelines because they achieve a correct separation of the regions of interest from the image background. Two of the frequently used clustering techniques are fuzzy *C*-means (Dulyakarn and Rangsanseri, 2001; Yang *et al.*, 2004) and hierarchical clustering, including single-linkage, complete-linkage, and average-linkage.

The segmentation of bioimages depends on which features are needed. For instance, texture features can be extracted for chromatin compositions whereas concavity features could be employed for nuclear morphology. When the images are of cell-based assays where nuclear compartments are colored, methods such as globular template segmentation, watershed segmentation, or active contour/snake methods are recommended.

Nevertheless, when the images contain nonglobular objects such as neurons, the neuroanatomical analysis software Neurolucida (Glaser and Glaser, 1990) is often used. Alternatively, directional kernels are also considered for searching neuronal structures in confocal images (Al-Kofahi *et al.*, 2003). Finally, the tool ImageJ plugin NeuronJ (Meijering *et al.*, 2004) is advisable as well.

Feature extraction/selection

The feature extraction/selection step is of vital importance when we aim to classify objects in bioimages. In this step, representative feature vectors are usually obtained. To do this, dimensionality reduction techniques like PCA are necessary. In addition, in-depth knowledge of the research domain is needed because it will allow us to select those features that can help discriminate between objects of interest. For example, if the aim is to analyze dynamic fluorescent images, texture features should be used (Dorn *et al.*, 2008). For clustering gene expression patterns, global decomposition by eigen-embryo analysis is recommended (Peng *et al.*, 2006) whereas wavelet features able to detect global and local properties are advisable to recognize and annotate gene expression patterns (Zhou and Peng, 2007). When a collection of image characteristics is extracted, we would then recommend the use of minimum-redundant maximum-relevant feature selection algorithms (Ding and Peng, 2005; Peng *et al.*, 2005b) so that redundant features can be eliminated.

Classification

The classification step is also of vital importance in bioimage analysis pipelines because it is here where the features of the object of interest will be used for classification. Standard ML techniques are often employed to achieve this goal. Some of these classifiers are the following:

- Bayesian classifiers, which rely on the Bayes' theorem, can be used in binary classification problems. The naïve Bayes classifier is often employed for classifying objects of interest because it requires a small amount of training data for parameter estimation.
- The kNN rule (Kuncheva, 2014) does not require knowledge of a priori probabilities whereas it assigns an unknown object to the class of its nearest neighbor in the measurement space. Moreover, it does not require patterns to be represented in a suitable vector space but only a similarity measure or distance function.
- Decision trees (Kuncheva, 2014) do not use comparison functions to infer general properties of the training matrix. Nor do they use all the features for object classification but each class can be inferred using its own feature set.
- SVMs (Hastie *et al.*, 2001b) search the hyperplane that separates a set of training objects while maximizing the margin between their classes. This method aims to minimize the bound on the generalization error rather than minimize the mean squared error over the dataset.

When the training set consists of few bioimages, an ensemble of machines is advisable. This strategy builds several basic classifiers and combines their outputs for improving classification. In such cases, a diverse ensemble in which the classifiers have adequately different decision boundaries is needed. To achieve this diversity, strategies such as bagging, boosting, AdaBoost, stacked generalization, and a mixture of experts are often considered (Kuncheva, 2014; Polikar, 2006). Bagging (Kuncheva, 2014; Polikar, 2006) draws different training subsets with replacement and is particularly recommended when the number of images is limited. Boosting (Kuncheva, 2014; Polikar, 2006) makes new machines pay more attention to those examples in which previous machines produced errors. AdaBoost (Kuncheva, 2014; Polikar, 2006) generates a set of hypotheses by training weak machines and making sure that the cases misclassified by previous classifiers are included in the training data of the next classifier.

One of the methods used for clustering mRNA expression patterns of coexpressed genes was the minimum spanning tree cut (Peng *et al.*, 2006) whereas for determining cell identities, the absolute three-dimensional location of the cell along with its location patterns was employed (Long *et al.*, 2008). However, for assigning bioimage patterns to different annotation terms, parallel classifiers are highly recommended (Zhou and Peng, 2007).

Visualization

As in other bioinformatics pipelines, visualization is vital for result verification. Among the techniques often used in bioimage analyses are surface and flow visualization. Tools for the visualization of images of protein surfaces and gene expression can be utilized in addition to immersive visualization systems such as NCMIR's ATLAS and ImmersaDeskTM (Ai *et al.*, 2005).

Conclusions

The spectrum of bioinformatics tools and methods is currently too wide for an exhaustive review. The computational approaches grow in sophistication alongside technological improvements in the corresponding experimental protocols. New requirements, which become more evident with increase in volume and complexity of biological data, will further push the development of

efficient and comprehensive solutions. This article covers four main research areas: genomics, transcriptomics, proteomics, and bioimage analysis.

Within the genomics area, we presented current efforts that focus on accurate mapping of sequences to the reference and identifying regions that show variation. As the number of quantifiable changes will continue to rise, new frameworks geared towards integrating this information and linking them to phenotypic changes will be required.

In the transcriptome section, we examined several computational data processing approaches and covered linking expression profiles to actionable results. We summarized popular tools and libraries that can be easily built upon and extended. As our understanding of pathways, networks and omics interaction grows, so will the scope of the analyses. Future solutions will look at RNA changes more in the context of global interaction network, rather than few genes with higher or lower expression.

For proteomics, we have provided a detailed guideline for MoRF prediction, including data description, examples of state-of-the-art tools, MoRF classification schemes, and pipelines of MoRF downstream analysis. Any reader can easily follow the steps to develop a new MoRF predictor, and thus analyze MoRFs in protein sequences using the presented methodology.

Finally, within the bioimage analysis area, we described the standard workflow for analyzing digital images, which includes preprocessing, image registration and segmentation, feature extraction, classification, and visualization. With the capture of high-resolution images of different cellular mechanisms, future bioinformatics approaches will definitely be necessary to keep up with the technological advances.

Author Contributions

AS and PJK defined the structure of the article. YL, AS and PJK wrote the abstract, introduction, and conclusions. YL and DS covered Section "Genome Analysis" (Genome). PJK covered Section "Transcriptome Analysis" (Transcriptome). RS and AS covered Section "Workflows in Proteomics" (Proteome). YL covered Section "Bioimage Analysis Pipelines" (Bioimages). AS and TT oversaw and managed article creation.

See also: Computing for Bioinformatics. Natural Language Processing Approaches in Bioinformatics

References

Abeel, T., Parys, T.V., Saeys, Y., Galagan, J., Peer, Y.V.D., 2012. GenomeView: A next-generation genome browser. Nucleic Acids Research 40, e12.
Abraham, G., Inouye, M., 2014. Fast principal component analysis of large-scale genome-wide data. PLoS ONE 9. e93766.
Abyzov, A., Urban, A.E., Snyder, M., Gerstein, M., 2011. CNVnator: An approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. Genome Research 21, 974–984.
Ai, Z., Chen, X., Rasmussen, M., Folberg, R., 2005. Reconstruction and exploration of three-dimensional confocal microscopy data in an immersive virtual environment. Computerized Medical Imaging and Graphics 29, 313–318.
Al-Kofahi, K.A., Can, A., Lasek, S., *et al.*, 2003. Median-based robust algorithms for tracing neurons from noisy confocal microscope images. IEEE Transactions on Information Technology in Biomedicine 7, 302–317.
Altschul, S.F., Madden, T.L., Schaffer, A.A., *et al.*, 1997. Gapped blast and psi-blast: A new generation of protein database search programs. Nucleic Acids Research 17, 3389–3402.
Ang, J.C., Mirzal, A., Haron, H., Hamed, H.N.A., 2016. Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection. IEEE/ACM Transactions on Computational Biology and Bioinformatics 13, 971–989.
Babraham-Bioinformatics, 2018. A quality control tool for high throughput sequence data.Available at: http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc.
Bao, H., Guo, H., Wang, J., *et al.*, 2009. MapView: Visualization of short reads alignment on a desktop computer. Bioinformatics 25, 1554–1555.
Behjati, S., Tarpey, P.S., 2013. What is next-generation sequencing? Archives of Disease in Childhood - Education and Practice 98, 236–238.
Betzig, E., Patterson, G.H., Sougrat, R., *et al.*, 2006. Imaging intracellular fluorescent proteins at nanometer resolution. Science 313, 1642–1645.
Bolger, A.M., Lohse, M., Usadel, B., 2014. Trimmomatic: A flexible trimmer for Illumina sequence data. Bioinformatics 30, 2114–2120.
Bray, N.L., Pimentel, H., Melsted, P., Pachter, L., 2016. Near-optimal probabilistic RNA-seq quantification. Nature Biotechnology 34, 525–527.
Brodie, A., Tovia-Brodie, O., Ofran, Y., 2014. Large scale analysis of phenotype-pathway relationships based on GWAS results. PLoS ONE. e100887.
Browning, B.L., Browning, S.R., 2009. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. American Journal of Human Genetics 84, 210–223.
Buels, R., Yao, E., Diesh, C.M., *et al.*, 2016. JBrowse: A dynamic web platform for genome visualization and analysis. Genome Biology, 17. p. 66.
Carver, T., Harris, S.R., Otto, T.D., *et al.*, 2013. BamView: Visualizing and interpretation of next-generation sequencing read alignments. Briefings in Bioinformatics 14, 203–212.
Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: KDD'16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California pp. 785–794.
Cheung, M.-S., Down, T.A., Latorre, I., Ahringer, J., 2011. Systematic bias in high-throughput sequencing data and its correction by BEADS. Nucleic Acids Research 39, e103.
Chou, K.C., 2017. An unprecedented revolution in medicinal chemistry driven by the progress of biological science. Current Topics in Medicinal Chemistry 17, 2337–2358.
Cock, P.J.A., Fields, C.J., Goto, N., Heuer, M.L., Rice, P.M., 2010. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 38, 1767–1771.
Csardi, G., Nepusz, T., 2006. The igraph software package for complex network research. Inter Journal Complex Systems. 1695.
Curtis, R.K., Oresic, M., Vidal-Puig, A., 2005. Pathways to the analysis of microarray data. Trends in Biotechnology 23, 429–435.
Dehzangi, A., Heffterman, R., Sharma, A., *et al.*, 2015. Gram-positive and Gram-negative protein subcellular localization by incorporating evolutionary-based descriptors into Chou's general PseAAC. Journal of Theoretical Biology 364, 284–294.

Dehzangi, A., López, Y., Lal, S.P., *et al.*, 2017. PSSM-Suc: Accurately predicting succinylation using position specific scoring matrix into bigram for feature extraction. Journal of Theoretical Biology 425, 97–102.
Dehzangi, A., López, Y., Lal, S., *et al.*, 2018. Improving succinylation prediction accuracy by incorporating the secondary structure via helix, strand and coil, and evolutionary information from profile bigrams. PLOS ONE 13, e0191900.
DeLuca, D.S., Levin, J.Z., Sivachenko, A., *et al.*, 2012. RNA-SeQC: RNA-seq metrics for quality control and process optimization. Bioinformatics 28, 1530–1532.
Depeursinge, A., Foncubierta-Rodriguez, A., Ville, D.V.D., Müller, H., 2014. Three-dimensional solid texture analysis in biomedical imaging: Review and opportunities. Medical Image Analysis 18, 176–196.
Diaz, A., Nellore, A., Song, J.S., 2012. CHANCE: Comprehensive software for quality control and validation of ChIP-seq data. Genome Biology, 13. . p. R98.
Ding, C., Peng, H., 2005. Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology 3, 185–205.
Disfani, F.M., Hsu, W.L., Mizianty, M.J., *et al.*, 2012. MoRFpred, a computational tool for sequence-based prediction and characterization of short disorder-to-order transitioning binding regions in proteins. Bioinformatics 28, i75–i83.
Dobin, A., Davis, C.A., Schlesinger, F., *et al.*, 2013. STAR: Ultrafast universal RNA-seq aligner. Bioinformatics 29, 15–21.
Dosztányi, Z., Mészáros, B., Simon, I., 2009. ANCHOR: Web server for predicting protein binding regions in disordered proteins. Bioinformatics 25, 2745–2746.
Dulyakarn, P., Rangsanseri, Y., 2001. Fuzzy C-means clustering using spatial information with application to remote sensing. In: Proceedings of the 22nd Asian Conference on Remote Sensing.
Du, P., Gu, S., Jiao, Y., 2014. PseAAC-General: Fast building various modes of general form of Chou's pseudo-amino acid composition for large-scale protein datasets. International Journal of Molecular Sciences 15, 3495–3506.
Dyson, H.J., Wright, E.P., 2005. Intrinsically unstructured proteins and their functions. Nature Reviews Molecular Cell Biology 6, 197–208.
Eberwine, J., Sul, J.Y., Bartfai, T., Kim, J., 2014. The promise of single-cell sequencing. Nature Methods 11, 25–27.
Echeverri, C.J., Perrimon, N., 2006. High-throughput RNAi screening in cultured cells: A user's guide. Nature Reviews Genetics 7, 373–384.
F.Dorn, J., Danuser, G., Yang, G., 2008. Computational processing and analysis of dynamic fluorescence image data. Methods in Cell Biology 85, 497–538.
Fabregat, A., Sidiropoulos, K., Garapati, P., *et al.*, 2016. The reactome pathway knowledgebase. Nucleic Acids Research 44, D481–D487.
Feng, X., Grossman, R., Stein, L., 2011. PeakRanger: A cloud-enabled peak caller for ChIP-seq data. BMC Bioinformatics 12, 139.
Fonville, J.M., Carter, C.L., Pizarro, L., *et al.*, 2013. Hyperspectral visualization of mass spectrometry imaging data. Analytical Chemistry 85, 1415–1423.
Forbes, S.A., Bhamra, G., Bamford, S., *et al.*, 2008. The Catalogue of Somatic Mutations in Cancer (COSMIC). In: HAINES, J.L. (Ed.), Current Protocols In Human Genetics.
Fowlkes, C.C., Hendriks, C.L.L., Keränen, S.V.E., *et al.*, 2008. A quantitative spatiotemporal atlas of gene expression in the *Drosophila* blastoderm. Cell 133, 364–374.
Auton, A., Brooks, L.D., Durbin, R.M., *et al.*, 2015. A global reference for human genetic variation. Nature 526, 68–74.
German, M.A., Pillay, M., Jeong, D.H., *et al.*, 2008. Global identification of microRNA-target RNA pairs by parallel analysis of RNA ends. Nature Biotechnology 26, 941–946.
Ginestet, C., 2011. ggplot2: Elegant graphics for data analysis. Journal of the Royal Statistical Society Series A 174, 245.
Glaser, J.R., Glaser, E.M., 1990. Neuron imaging with neurolucida – A PC-based system for image combining microscopy. Computerized Medical Imaging and Graphics 14, 307–317.
Hagberg, A., Swart, P.J., Chult, D.S., 2008. Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conference.
Hannon, 2010. FASTX-Toolkit: FASTQ/A short-reads pre-processing tools. Available at: http://hannonlab.cshl.edu/fastx_toolkit/.
Hastie, T., Friedman, J., Tibshirani, R., 2001b. Support vector machines and flexible discriminants. The Elements of Statistical Learning. New York: Springer.
Haury, A.C., Gestraud, P., Vert, J.P., 2011. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. PLoS ONE 6, e28210.
Hell, S.W., 2003. Toward fluorescence nanoscopy. Nature Biotechnology 21, 1347–1355.
Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Computing in Science & Engineering 9, 90–95.
Hurd, P.J., Nelson, C.J., 2009. Advantages of next-generation sequencing versus the microarray in epigenetic research. Briefings in Functional Genomics & Proteomics 8, 174–183.
Hwang, S., Kim, E., Lee, I., Marcotte, E.M., 2015. Systematic comparison of variant calling pipelines using gold standard personal exome variants. Scientific Reports 5, 17875.
Imamura, M., Shigemizu, D., Tsunoda, T., *et al.*, 2013. Assessing the clinical utility of a genetic risk score constructed using 49 susceptibility alleles for type 2 diabetes in a Japanese population. The Journal of Clinical Endocrinology and Metabolism 98, E1667–E1673.
International HapMap Consortium, 2005. A haplotype map of the human genome. Nature 437, 1299–1320.
Jiang, H., Wang, F., Dyer, N.P., Wong, W.H., 2010. CisGenome Browser: A flexible tool for genomic data visualization. Bioinformatics 26, 1781–1782.
Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., Tanabe, M., 2016. KEGG as a reference resource for gene and protein annotation. Nucleic Acids Research 44, D457–D462.
Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. International Journal of Computer Vision 1, 321–331.
Kauffmann, A., 2009. arrayQualityMetrics – A bioconductor package for quality assessment of microarray data. Bioinformatics 25, 415–416.
Kauffmann, A., Huber, W., 2010. Microarray data quality control improves the detection of differentially expressed genes. Genomics 95, 138–142.
Kawashima, S., Pokarowski, P., Pokarowska, M., *et al.*, 2008. AAindex: Amino acid index database, progress report 2008. Nucleic Acids Research 36, D202–D205.
Kharchenko, P.V., Tolstorukov, M.Y., Park, P.J., 2008. Design and analysis of ChIP-seq experiments for DNA-binding proteins. Nature Biotechnology 26, 1351–1359.
Kidd, B.A., Readhead, B.P., Eden, C., Parekh, S., Dudley, J.T., 2015. Integrative network modeling approaches to personalized cancer medicine. Personalized Medicine 12, 245–257.
Koboldt, D.C., Zhang, Q., Larson, D.E., *et al.*, 2012. VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Research 22, 568–576.
Kong, L., Wang, J., Zhao, S., *et al.*, 2012. ABrowse – a customizable next-generation genome browser framework. BMC Bioinformatics 13, 2.
Kuncheva, L.I., 2014. Combining Pattern Classifiers: Methods and Algorithms. New Jersey: John Wiley & Sons.
Landt, S.G., Marinov, G.K., Kundaje, A., *et al.*, 2012. ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. Genome Research 22, 1813–1831.
Langfelder, P., Horvath, S., 2008. WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics 9, 559.
Langmead, B., Salzberg, S.L., 2012. Fast gapped-read alignment with Bowtie 2. Nature Methods 9, 357–359.
Larson, D.E., Harris, C.C., Chen, K., *et al.*, 2012. SomaticSniper: Identification of somatic point mutations in whole genome sequencing data. Bioinformatics 28, 311–317.
Learned-Miller, E., 2006. Data driven image models through continuous joint alignment. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 236–250.
Lee, R.V.D., Buljan, M., Lang, B., *et al.*, 2014. Classification of intrinsically disordered regions and proteins. Chemical Reviews 114, 6589–6631.
Lee, S., Emond, M.J., Bamshad, M.J., *et al.*, 2012. Optimal unified approach for rare-variant association testing with application to small-sample case-control whole-exome sequencing studies. American Journal of Human Genetics 91, 224–237.
Lee, E., Helt, G.A., Reese, J.T., *et al.*, 2013. Web Apollo: A web-based genomic annotation editing platform. Genome Biology 14, R93.
Lever, J., Krzywinski, M., Altman, N., 2017. Points of Significance: Principal component analysis. Nature Methods 14, 641–642.
Liang, K., Keleş, S., 2012. Detecting differential binding of transcription factors with ChIP-seq. Bioinformatics 28, 121–122.
Liang, J., Mcinerney, T., Terzopoulos, D., 2006. United Snakes. Medical Image Analysis 10, 215–233.
Liao, Y., Smyth, G.K., Shi, W., 2014. featureCounts: An efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics 30, 923–930.
Lin, S.M., Du, P., Huber, W., Kibbe, W.A., 2008. Model-based variance-stabilizing transformation for Illumina microarray data. Nucleic Acids Research 36, e11.
Liu, B., Liu, F., Wang, X., Chen, J., 2015. Pse-in-One: A web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nucleic Acids Research 43, W65–W71.

Li, Q., Brown, J.B., Huang, H., Bickel, P.J., 2011. Measuring reproducibility of high-throughput experiments. The Annals of Applied Statistics 5, 1752–1779.
Li, H., Durbin, R., 2009. Fast and accurate short read alignment with Burrows–Wheeler transform. Bioinformatics 25, 1754–1760.
Li, H., Handsaker, B., Wysoker, A., *et al.*, 2009a. The Sequence Alignment/Map format and SAMtools. Bioinformatics 25, 2078–2079.
Li, B., Leal, S.M., 2008. Methods for detecting associations with rare variants for common diseases: Application to analysis of sequence data. American Journal of Human Genetics 83, 311–321.
Li, H., Ruan, J., Durbin, R., 2008. Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Research 18, 1851–1858.
Li, R., Yu, C., Li, Y., *et al.*, 2009b. SOAP2: An improved ultrafast tool for short read alignment. Bioinformatics 25, 1966–1967.
Long, F., Peng, H., Liu, X., Kim, S., Myers, G., 2008. Automatic Recognition of Cells (ARC) for 3D Images of *C. elegans*. In: Vingron, M., Wong, L. (Eds.), Research in Computational Molecular Biology. Berlin, Heidelberg: Springer.
Long, F., Peng, H., Sudar, D., Lelièvre, S.A., Knowles, D.W., 2007a. Phenotype clustering of breast epithelial cells in confocal images based on nuclear protein distribution analysis. BMC Cell Biology 8, S3.
López, Y., Dehzangi, A., Lal, S.P., *et al.*, 2017. SucStruct: Prediction of succinylated lysine residues by using structural properties of amino acids. Analytical Biochemistry 527, 24–32.
López, Y., Sharma, A., Dehzangi, A., *et al.*, 2018. Success: Evolutionary and structural properties of amino acids prove effective for succinylation site prediction. BMC Genomics 19, 923.
Love, M.I., Huber, W., Anders, S., 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biology 15, 550.
Lyons, J., Paliwal, K.K., Dehzangi, A., *et al.*, 2016. Protein fold recognition using HMM–HMM alignment and dynamic programming. Journal of Theoretical Biology 393, 67–74.
Machanick, P., Bailey, T.L., 2011. MEME-ChIP: Motif analysis of large DNA datasets. Bioinformatics 27, 1696–1697.
Madsen, B.E., Browning, S.R., 2009. A groupwise association test for rare mutations using a weighted sum statistic. PLoS Genetics 5, e1000384.
Malhis, N., Gsponer, J., 2015. Computational identification of MoRFs in protein sequences. Bioinformatics 31, 1738–1744.
Malhis, N., Jacobson, M., Gsponer, J., 2016. MoRFchibi SYSTEM: Software tools for the identification of MoRFs in protein sequences. Nucleic Acids Research 44, W488–W493.
Marchini, J., Howie, B., Myers, S., Mcvean, G., Donnelly, P., 2007. A new multipoint method for genome-wide association studies by imputation of genotypes. Nature Genetics 39, 906–913.
Marschall, T., Costa, I.G., Canzar, S., *et al.*, 2012. CLEVER: Clique-enumerating variant finder. Bioinformatics 28, 2875–2882.
Martin, M., 2012. Cutadapt removes adapter sequences from high-throughput sequencing reads. Bioinformatics in Action 17, 10–12.
Ma, S., Kosorok, M.R., 2009. Identification of differential gene pathways with principal component analysis. Bioinformatics 25, 882–889.
Mckenna, A., Hanna, M., Banks, E., *et al.*, 2010. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. Genome Research 20, 1297–1303.
Meijering, E., 2012. Cell segmentation: 50 years down the road. IEEE Signal Processing Magazine 29, 140–145.
Meijering, E., Jacob, M., Sarria, J.-C.F., *et al.*, 2004. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. Cytometry Part A 58A, 167–176.
Metzker, M.L., 2010. Sequencing technologies – the next generation. Nature Reviews Genetics 11, 31–46.
Moffat, J., Grueneberg, D.A., Yang, X., *et al.*, 2006. A lentiviral RNAi library for human and mouse genes applied to an arrayed viral high-content screen. Cell 124, 1283–1298.
Morgenthaler, S., Thilly, W.G., 2007. A strategy to discover genes that carry multi-allelic or mono-allelic risk for common diseases: A cohort allelic sums test (CAST). Mutation Research 615, 28–56.
Neumann, B., Walter, T., Hériché, J.-K., *et al.*, 2010. Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. Nature 464, 721–727.
Newton-Cheh, C., Johnson, T., Gateva, V., *et al.*, 2009. Genome-wide association study identifies eight loci associated with blood pressure. Nature Genetics 41, 666–676.
Ng, L., Pathak, S., Kuan, C., *et al.*, 2007. Neuroinformatics for genome-wide 3-D gene expression mapping in the mouse brain. IEEE/ACM Transactions on Computational Biology and Bioinformatics 4, 382–393.
Ozaki, K., Ohnishi, Y., Iida, A., *et al.*, 2002. Functional SNPs in the lymphotoxin-alpha gene that are associated with susceptibility to myocardial infarction. Nature Genetics 32, 650–654.
Pabinger, S., Dander, A., Fischer, M., *et al.*, 2014. A survey of tools for variant analysis of next-generation genome sequencing data. Briefings in Bioinformatics 15, 256–278.
Pedersen, B.S., Layer, R.M., Quinlan, A.R., 2016. Vcfanno: fast, flexible annotation of genetic variants. Genome Biology 17, 118.
Peng, H., Long, F., Ding, C., 2005b. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1226–1238.
Peng, H., Long, F., Eisen, M.B., Myers, E.W., 2006. Clustering gene expression patterns of fly embryos. In: Proceedings of the 3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro, pp. 1144–1147.
Pimentel, H., Bray, N.L., Puente, S., Melsted, P., Pachter, L., 2017. Differential analysis of RNA-seq incorporating quantification uncertainty. Nature Methods 14, 687–690.
Pincus, Z., Theriot, J.A., 2007. Comparison of quantitative methods for cell-shape analysis. Journal of Microscopy 227, 140–156.
Podolskiy, D.I., Lobanov, A.V., Kryukov, G.V., Gladyshev, V.N., 2016. Analysis of cancer genomes reveals basic features of human aging and its role in cancer development. Nature Communications 7, 12157.
Polikar, R., 2006. Ensemble based systems in decision making. IEEE Circuits and Systems Magazine 6, 21–45.
Pratt, W.K., 2014. Introduction to Digital Image Processing. CRC Press Taylor & Francis Group.
Price, A.L., Kryukov, G.V., De Bakker, P.I., *et al.*, 2010. Pooled association tests for rare variants in exon-resequencing studies. American Journal of Human Genetics 86, 832–838.
Purcell, S., Neale, B., Todd-Brown, K., *et al.*, 2007. PLINK: A tool set for whole-genome association and population-based linkage analyses. American Journal of Human Genetics 81, 559–575.
Qin, Z.S., Jianjun, Y., Shen, J., *et al.*, 2010. HPeak: An HMM-based algorithm for defining read-enriched regions in ChIP-Seq data. BMC Bioinformatics 11, 369.
Quinlan, A.R., Hall, I.M., 2010. BEDTools: A flexible suite of utilities for comparing genomic features. Bioinformatics 26, 841–842.
Qu, L., Long, F., Peng, H., 2015. 3-D Registration of biological images and models: Registration of microscopic images and its uses in segmentation and annotation. IEEE Signal Processing Magazine 32, 70–77.
Rashid, N.U., Giresi, P.G., Ibrahim, J.G., Sun, W., Lieb, J.D., 2011. ZINBA integrates local covariates with DNA-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions. Genome Biology 12, R67.
Remmert, M., Biegert, A., Hauser, A., Söding, J., 2011. HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nature Methods 9, 173–175.
Ritchie, M.E., Diyagama, D., Neilson, J., *et al.*, 2006. Empirical array quality weights in the analysis of microarray data. BMC Bioinformatics 7, 261.
Ritchie, M.E., Phipson, B., Wu, D., *et al.*, 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Research 43.
Ritchie, M.E., Silver, J., Oshlack, A., *et al.*, 2007. A comparison of background correction methods for two-colour microarrays. Bioinformatics 23, 2700–2707.
Rohr, K., Fornefett, M., Stiehl, H.S., 2003. Spline-based elastic image registration: Integration of landmark errors and orientation attributes. Computer Vision and Image Understanding 90, 153–168.
Van Rooden, S.M., Heiser, W.J., Kok, J.N., *et al.*, 2010. The identification of Parkinson's disease subtypes using cluster analysis: A systematic review. Movement Disorders 25, 969–978.

Roy, S., Coldren, C., Karunamurthy, A., *et al.*, 2018. Standards and guidelines for validating next-generation sequencing bioinformatics pipelines. The Journal of Molecular Diagnostics 20, 4–27.
Roy, S., Laframboise, W.A., Nikiforov, Y.E., *et al.*, 2016. Next-generation sequencing informatics challenges and strategies for implementation in a clinical environment. Archives of Pathology & Laboratory Medicine 140, 958–975.
Ruffier, M., Kähäri, A., Komorowska, M., *et al.*, 2017. Ensembl core software resources: Storage and programmatic access for DNA sequence and genome annotation. Database 2017.bax020.
Rusk, N., Kiermer, V., 2008. Primer: Sequencing – the next generation. Nature Methods 5, 15.
Rust, M.J., Bates, M., Zhuang, X., 2006. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). Nature Methods 3, 793–796.
Salvado, J., Roque, B., 2005. Detection of calcifications in digital mammograms using wavelet analysis and contrast enhancement. In: Proceedings of the IEEE International Workshop on Intelligent Signal Processing.
Sathirapongsasuti, J.F., Lee, H., Horst, B.A.J., *et al.*, 2011. Exome sequencing-based copy-number variation and loss of heterozygosity detection: ExomeCNV. Bioinformatics 27, 2648–2654.
Schölkopf, B., Zien, A., 2006. Introduction to semi-supervised learning. In: Chapelle, O., Schölkopf, B., Zien, A. (Eds.), Semi-Supervised Learning. MIT Press.
Sepp, K.J., Hong, P., Lizarraga, S.B., *et al.*, 2008. Identification of neural outgrowth genes using genome-wide RNAi. PLoS Genetics 4, e1000111.
Shalon, D., Smith, S.J., Brown, P.O., 1996. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. Genome Research 6, 639–645.
Shamir, L., Delaney, J.D., Orlov, N., Eckley, D.M., Goldberg, I.G., 2010. Pattern Recognition software and techniques for biological image analysis. PLoS Computational Biology 6, e1000974.
Shannon, P., Markiel, A., Ozier, O., *et al.*, 2003. Cytoscape: A software environment for integrated models of biomolecular interaction networks. Genome Research 13, 2498–2504.
Shao, Z., Zhang, Y., Yuan, G.-C., Orkin, S.H., Waxman, D.J., 2012. MAnorm: A robust model for quantitative comparison of ChIP-Seq data sets. Genome Biology 13, R16.
Sharan, R., Ulitsky, I., Shamir, R., 2007. Network-based prediction of protein function. Molecular Systems Biology 3, 88.
Sharma, R., Bayarjargal, M., Tsunoda, T., Patil, A., Sharma, A., 2018a. MoRFPred-plus: Computational identification of MoRFs in protein sequences using physicochemical properties and HMM profiles. Journal of Theoretical Biology 437, 9–16.
Sharma, A., Boroevich, K., Shigemizu, D., *et al.*, 2017a. Hierarchical maximum likelihood clustering approach. IEEE Transactions on Biomedical Engineering 64, 112–122.
Sharma, R., Dehzangi, A., Lyons, J., *et al.*, 2015. Predict Gram-positive and Gram-negative subcellular localization via incorporating evolutionary information and physicochemical features into Chou's general PseAAC. IEEE Transactions on Nanobioscience 14, 915–926.
Sharma, A., Imoto, S., Miyano, S., 2012a. A between-class overlapping filter-based method for transcriptome data analysis. Journal of Bioinformatics and Computational Biology 10, 1250010.
Sharma, A., Imoto, S., Miyano, S., 2012b. A filter based feature selection algorithm using null space of covariance matrix for DNA microarray gene expression data. Current Bioinformatics 7, 289–294.
Sharma, A., Imoto, S., Miyano, S., 2012c. A top-r feature selection algorithm for microarray gene expression data. IEEE/ACM Transactions on Computational Biology and Bioinformatics 9, 754–764.
Sharma, A., Imoto, S., Miyano, S., Sharma, V., 2012d. Null space based feature selection method for gene expression data. International Journal of Machine Learning and Cybernetics 3, 269–276.
Sharma, A., Kamola, P.J., Tsunoda, T., 2017b. 2D-EM clustering approach for high-dimensional data through folding feature vectors. BMC Bioinformatics. 18, 547.
Sharma, A., Koh, C.H., Imoto, S., Miyano, S., 2011. Strategy of finding optimal number of features on gene expression data. Electronics Letters 47, 480–482.
Sharma, R., Kumar, S., Tsunoda, T., Patil, A., Sharma, A., 2016b. Predicting MoRFs in protein sequences using HMM profiles. BMC Bioinformatics 17, 504.
Sharma, A., López, Y., Tsunoda, T., 2017c. Divisive hierarchical maximum likelihood clustering. BMC Bioinformatics 18, 546.
Sharma, A., Lyons, J., Dehzangi, A., Paliwal, K.K., 2013. A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition. Journal of Theoretical Biology 320, 41–46.
Sharma, A., Paliwal, K.K., 2007. Fast principal component analysis using fixed-point algorithm. Pattern Recognition Letters 28, 1151–1155.
Sharma, A., Paliwal, K.K., 2011. A gene selection algorithm using Bayesian classification approach. American Journal of Applied Sciences 9, 127–131.
Sharma, A., Paliwal, K.K., Imoto, S., Miyano, S., 2014. A feature selection method using improved regularized linear discriminant analysis. Machine Vision and Applications 25, 775–786.
Sharma, R., Raicar, G., Tsunoda, T., Patil, A., Sharma, A., 2018b. OPAL: prediction of MoRF regions in intrinsically disordered protein sequences. Bioinformatics 34, 1850–1858.
Sharma, A., Shigemizu, D., Boroevich, K.A., *et al.*, 2016a. Stepwise iterative maximum likelihood clustering approach. BMC Bioinformatics 17, 319.
Sherry, S.T., Ward, M.H., Kholodov, M., *et al.*, 2001. dbSNP: The NCBI database of genetic variation. Nucleic Acids Research 29, 308–311.
Shigemizu, D., Abe, T., Morizono, T., *et al.*, 2014. The construction of risk prediction models using GWAS data and its application to a type 2 diabetes prospective cohort. PLoS ONE 9, e92549.
Silver, J.D., Ritchie, M.E., Smyth, G.K., 2009. Microarray background correction: Maximum likelihood estimation for the normal-exponential convolution. Biostatistics 10, 352–363.
Slenter, D.N., Kutmon, M., Hanspers, K., *et al.*, 2018. WikiPathways: A multifaceted pathway database bridging metabolomics to other omics research. Nucleic Acids Research 46, D661–D667.
Soneson, C., Love, M.I., Robinson, M.D., 2015. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. F1000Research 4, 1521.
Spyrou, C., Stark, R., Lynch, A., Tavaré, S., 2009. BayesPeak: Bayesian analysis of ChIP-seq data. BMC Bioinformatics 10, 299.
Subramanian, A., Tamayo, P., Mootha, V.K., *et al.*, 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America 102, 15545–15550.
Sun, C., Medvedev, P., 2017. VarMatch: Robust matching of small variant datasets using flexible scoring schemes. Bioinformatics 33, 1301–1308.
Tarca, A.L., Bhatti, G., Romero, R., 2013. A comparison of gene set analysis methods in terms of sensitivity, prioritization and specificity. PLoS ONE 8, e79217.
Thomas-Chollier, M., Darbo, E., Herrmann, C., *et al.*, 2012. A complete workflow for the analysis of full-size ChIP-seq (and similar) data sets using peak-motifs. Nature Protocols 7, 1551–1568.
Uversky, V., 2014. Introduction to Intrinsically Disordered Proteins (IDPs). Chemical Reviews 114, 6557–6560.
Vidal, M., Cusick, M.E., Barabasi, A.L., 2011. Interactome networks and human disease. Cell 144, 986–998.
Viola, P., Wells, W., 1997. Alignment by maximization of mutual information. International Journal of Computer Vision 24, 137–154.
Walter, T., Shattuck, D.W., Baldock, R., *et al.*, 2010. Visualization of image data from cells to organisms. Nature Methods 7, S26–S41.
Wang, K., Li, M., Hakonarson, H., 2010. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Research 38, e164.
Wang, B., Mezlini, A.M., Demir, F., *et al.*, 2014. Similarity network fusion for aggregating data types on a genomic scale. Nature Methods 11, 333–337.
Wright, P.E., Dyson, H.J., 2015. Intrinsically disordered proteins in cellular signalling and regulation. Nature Reviews Molecular Cell Biology 16, 18–29.
Wu, M.C., Lee, S., Cai, T., *et al.*, 2011. Rare-variant association testing for sequencing data with the sequence kernel association test. American Journal of Human Genetics 89, 82–93.

Xu, S., Grullon, S., Ge, K., Peng, W., 2014. Spatial clustering for identification of ChIP-Enriched Regions (SICER) to map regions of histone methylation patterns in embryonic stem cells. In: Kidder B. (Eds.), Stem Cell Transcriptional Networks, Humana Press, New York.
Yang, Y., Chongxun, Z., Lin, P., 2004. A novel fuzzy C-means clustering algorithm for image thresholding. Measurement Science Review 4, 11–19.
Yang, Y., Heffernan, R., Paliwal, K., *et al.*, 2017. SPIDER2: A package to predict secondary structure, accessible surface area and main-chain torsional angles by deep neural networks. Methods in Molecular Biology 1484, 55–63.
Yu, D., Kim, M., Xiao, G., Hwang, T.H., 2013. Review of biological network data and its applications. Genomics & Informatics 11, 200–210.
Zambelli, F., Pesole, G., Pavesi, G., 2013. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. Briefings in Bioinformatics 14, 225–237.
Zhang, Y., Liu, T., Meyer, C.A., *et al.*, 2008. Model-based analysis of ChIP-Seq (MACS). Genome Biology 9, R137.
Zhou, J., Peng, H., 2007. Automatic recognition and annotation of gene expression patterns of fly embryos. Bioinformatics 23, 589–596.
Zhu, L.J., Gazin, C., Lawson, N.D., *et al.*, 2010. ChIPpeakAnno: a Bioconductor package to annotate ChIP-seq and ChIP-chip data. BMC Bioinformatics 11, 237.
Zhu, Z., Zhang, F., Hu, H., *et al.*, 2016. Integration of summary data from GWAS and eQTL studies predicts complex trait gene targets. Nature Genetics 48, 481–487.

Relevant Websites

www.r-project.org
R Project.
www.bioconductor.org
Bioconductor.