BioConductor: Microarray versus Next-Generation Sequencing toolsets

Sander Bollen, *Utrecht University*

February 11, 2014

Both microarray and next generation sequencing platforms generate copious amounts of data for each experiment. Thus, computational tools are required to handle and process such experiments.

For microarray analysis, many open-source R packages have been developed, most of which are available through the BioConductor project. With certain microarray platforms, it is now possible to perform the entire analysis chain using only BioConductor tools. A comprehensive next-generation sequencing toolchain in pure R/BioConductor has until so far not been possible. Next-generation sequencing analysis still requires the use of disparate tools on disparate platforms: only some of them in R.

This literature review reviewed the existing BioConductor toolsets for both approaches, and highlights key differences and similarities.

Introduction

Both microarray and next-generation sequencing techniques require adequate bioinformatics tools to process and analyse data. Bioinformatics repositories exist for many programming languages, but the BioConductor repository for the R statistical language is one of the biggest collection of bioinformatics tools [1]. As of January 2014, the BioConductor R repository contains over 1,500 individual packages. The majority of packages are annotation packages, leaving about 600 non-annotation packages (see figure 1 and table 1).

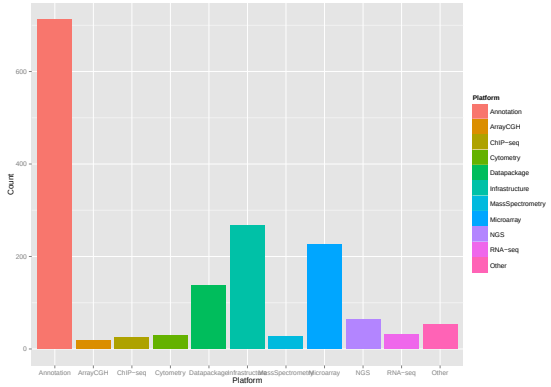

Figure 1: *BioConductor package classifications*

Of these remaining 600 packages, 225 packages are specifically targeted at microarray analysis. Just half of this amount is specifically tailored to next-generation sequencing (including RNA-seq and ChIP-seq). Many packages are generic packages that model statistical models, offer visualization tools, provide BioConductor data structures (e.g. expressionSets), or offer ways to connect to other non-BioConductor tools (e.g. Ensembl) (see table 1).

Microarray platforms

All microarray platforms are based on the hybridization of short probes to some target sequence on to a so-called "array". The target sequences are directly or indirectly labeled with some fluorescent dye. Using this fluorescence, the abundance of target sequence can be measured. This is the so-called one-color approach, used by Affymetrix and Illumina. Two-color

Type	Count
Annotation	713
Infrastructure	267
Microarray	225
Datapackage	138
NGS	63
RNA-seq	31
Cytometry	29
MassSpectrometry	26
ChIP-seq	25
ArrayCGH	18
Other	52
-	-
Total	1587

Table 1: *BioConductor package classifications*

approaches also exist, as used by Agilent [1]. In this approach, two samples are labeled with fluorescent dyes, each sample with a different color. Both samples are then simultaneously hybridized to the same array. The measured values in this approach are thus a *relative* measure.

Arrays have been developed for many purposes. The traditional purpose is to measure gene expression. In this setup, total RNA is extracted from a sample, then reverse-transcribed into cDNA and optionally retranscribed into cRNA (Affymetrix [2]). The resulting cDNA or cRNA is then directly or indirectly fluorescently labeled, and hybridized onto an array containing many probes. The probes used here generally target known transcripts. A special case of gene expression arrays is the exon array. In this case, probes are targeted to different (potential) splice variants of a gene. As such, different splice variants can be detected.

Four other types of experiments for which microarray platforms have proved useful are array-CGH (aCGH), array-based SNP profiling, ChIP-on-chip and MeDIP-on-chip platforms. With aCGH, genomic DNA is extracted from a sample, fluorescently labeled and subsequently hybridized onto an array. Instead of measuring the expression state, this approach measures the abundance of specific genomic regions. As such, chromosomal abnormalities can be detected. With array-based SNP profiling, the probes on the array target known single nucleotide polymorphisms (SNPs). As with aCGH, genomic DNA is extracted from the sample. Thus, this allows the detection of the abundance of SNPs in a sample. However, new SNPs are not discovered.

Arrays technologies can be combined with precipitation technologies, such as ChIP-on-chip and MeDIP-on-chip technologies. During ChIP-on-chip, sequences of DNA are first immunoprecipitated by targeting a DNA-bound protein with an antibody. The precipitated sequences are then analyzed by DNA array. MeDIP-on-chip precipitates DNA by targeting methylated DNA with an antibody, after which the sequences are analyzed by DNA array. These two technologies thus enable one to image the epigenetic landscape.

Next-generation sequencing platforms

For decades, sequencing was a tedious and slow process. For example, the human genome project, completed in 2001, took 11 years to complete and cost over a billion dollars. The predominant technique back then was so-called Sanger sequencing [2]. Sanger sequencing is based on a PCR reaction with modified nucleotides. Apart from normal deoxynucleotides (dNTPs), the reaction mix also contains fluorescently labeled dideoxynucleotides (ddNTPs). The incorporation of these ddNPTs into the DNA chain prevents the polymerase from elongating the chain any further. The incorporation of ddNPTs is a stochastic process, thus the reaction mix eventually contains many different DNA molecules, each progressively longer. This resulting mixture is then separated by capillary chromatography, and the fluorescence is measured.

In recent years, several vendors have introduced new ways of sequencing. These *next-generation sequencing* platforms are orders of magnitude faster than traditional Sanger sequencing. This allows the sequencing of entire genomes in as little as a few days. The costs have also plummeted [3], with the most current platforms skimming the $1000 per genome goal [4]. This "revolution" in sequencing has allowed the creation of entirely new fields, such as RNA-seq (gene expression analysis using sequencing tools), exome sequencing, ChIP-seq (protein affinity analysis) and MeDIP-seq (DNA methylation analysis).

The chemistry involved in these platforms is very diverse, but generally involves some sort of amplification on a surface. Current next-generation sequencing platforms available are Illumina's HiSeq, Illumina's MiSeq, Life Technologies' Ion Torrent PGM, Life Technologies' SOLiD sequencing, Roche's 454

[1] It is also possible to use Agilent chips in a one color approach

[2] Even today, Sanger sequencing is still considered to be the "gold standard" in terms of quality

sequencing, and Pacific BioSystems' RS system. Expected technologies in the short to medium term future include nanopore sequencing.

454 sequencing was the first to enter the market in 2005, and relies on pyrosequencing [5]. Fragments are attached to emulsified beads. Amplification of fragments on the beads is then performed while still in emulsion [6]. The beads are subsequently transferred to "pico-titer plate", where the final pyrosequencing step takes place.

Illumina HiSeq sequencing utilizes the "sequencing-by-synthesis" technology. This relies on clonal clusters of fragments on an eight-lane flow cell [7]. Sequencing then starts by a new round of amplification with reversible terminator dNTPs. These reversible terminator dNTPs are fluorescently labeled. After each addition of a base, the reaction stops, and the fluorescent signal is measured. The fluorescent tag (the terminator) is then cleaved off enzymatically, and a new cycle of synthesis can start [7, 8].

With SOLiD sequencing, the sequence method is not direct. Instead, it uses 2-base encoding, which means the sequence of a fragment can be deduced after a coverage of 5 [8]. As with 454 sequencing, clonal amplification is performed on beads, after which sequencing occurs by ligation [9].

Ion Torrent Personal Genome Sequencer (PGM) was the first next-generation desktop sequencer. It aims to provide quick and easy sequencing. DNA fragments are ligated with specific adapter sequences, and then clonally amplified in emulsion on microbeads [10]. The Ion Torrent technology then builds sequences sequentially. When bases are incorporated into the growing molecule, protons are released proportionally to the amount of bases incorporated [10], and this is then measured by semiconductor-based technology.

Another emergent technology Pacific Biosystems' RS system. All other systems feature an early amplification step before sequencing commences. The RS system differs from here in that it is a single-molecule real time (SMRT) sequencing method [10]. A single DNA molecule is amplified once by a polymerase. The polymerase is attached to the bottom of a zero-mode waveguide nanostructure. Fluorescently labeled dNTPs are only excited when they are being incorporated in the growing DNA chain [11]. The fluorescent tag is removed when incorporation is complete. In this manner, single molecules can be sequenced in real time [11].

This review

To classify BioConductor packages, this review has classified packages with their most specific biocViews. Popularity of packages was determined by package download statistics and citation count in Google Scholar (if available).

Raw data processing

The very first step in any analysis is acquiring raw data. This raw data comes in a wide variety of formats, and as such many vendors have integrated raw data processing into their hardware units.

Microarray

In most microarray platforms, the entire microarray plate is measured at the same time. This results in whole-plate image files as being the raw data format for many platforms. These are subsequently processed into a table format by most vendors.

Of the two biggest vendors - Affymetrix GeneChip and Illumina BeadArray - affymetrix provides the most transparent file format. This is partly due to each Illumina BeadArray being unique; the probes are randomly located on the array [12]. This requires a location file for each and every array, whereas Affymetrix GeneChip arrays have a fixed and well-known location for their probes. The Affymetrix CEL file format is furthermore well-documented. This makes it relatively easier for developers to work with Affymetrix files. The BioConductor `affy` package can read and process affymetrix CEL files natively without any previous processing [13].

Many chips can be read by some third-party scanners, such as GenePix. Output by GenePix scanners is supported by the BioConductor `limma` package, as per the `read.maimages` function [14, 15], which thus makes it possible to process some highly specific microarray platforms such as Agilent microarrays.

Next-generation sequencing

Data output formats for microarray platforms are extremely diverse. Contrary to this trend, an informal standard has been developed for next generation sequencing. Almost all NGS platforms eventually output a FASTQ file, which resembles a normal FASTA format sequence file, with phred quality scores attached to every base. One entry is usually present for each and every read. A phred quality score refers to the chance of a wrongly called

base. It is an inverse logarithmic scale, where a higher score means an ever smaller chance of a wrongfully called base (see formula (1)). Higher is therefore better. See listing 1 for an example of a FASTQ entry.

$$Q = -10log_{10}P \quad (1)$$

where:
Q = quality
P = chance of wrongfully calling a base

The phred scores are represented by an ASCII character in FASTQ files. Several encodings of phred scores exist. The Sanger institute uses ASCII values 33 to 126 to represent phred scores from 0 to 93 [16]. Newer versions of Illumina (>1.8) will use sanger encodings for their FASTQ files. FASTQ files are human-readable files completely in ASCII format.

```
@some_sequence_identifier
AAAAAAAAAA
+
!!!!!!!!!!
```

Listing 1: *"An example FASTQ entry of a sequence of adenines with the lowest possible quality score (!)"*

Quality control

Microarray

An essential step for microarray analysis is quality control, as array-based technologies produce inherent biases. Many BioConductor packages have been developed to asses the quality of experiments. For spotted arrays the `arrayQuality` can be used. The `arrayQualityMetrics` [17] package utilizes the BioConductor container formats for array experiments, such as `ExpressionSets`, and is therefore very useful for analysis of Affymetrix chips. However, it does support both one-color and two-color experiments, and is thus not essentially limited to one platform. More specific quality control packages targeting just one platform include `affyExpress`. Several Affymetrix-specific quality control packages exist, such as `ArrayTools` [18], `affyQCReport` [19], `simpleaffy` [20], `yaqcaffy` (which is a wrapper for `simpleaffy`), and `affyPLM` [21].

Next-generation sequencing

Quality control for NGS applications is somewhat less of an issue than with microarray applications. As most sequencers output FASTQ files, which already *include* quality measures, quality control has partially already been accounted for. Nevertheless, quality control can be helpful, and in many cases required. As such, packages have been developed for quality control for NGS applications. The most prominent of these is `htSeqTools` [22].

Preprocessing

An essential step in any microarray or NGS analysis is preprocessing of the raw data. Microarray technologies require normalization, whereas most next-generation sequencing workflows demand an alignment step.

Microarray

The normalization of microarray data is dependent on the platform used. The most common platforms, Affymetrix GeneChip and Illumina BeadArray, require very different methods of normalization.

Affymetrix GeneChip

Most microarray normalization tools targeting the Affymetrix GeneChip platform use either one of the following two normalization algorithms: Robust Multichip Average (RMA) [23–25] or the outdated MAS5. MAS5 is the traditional normalization tool but has become outdated, whereas RMA is somewhat newer. RMA has the advantage of being significantly faster than MAS5. Whereas RMA normalizes expression over multiple arrays (hence the *Multichip* denominator), MAS5 normalizes each array independently. This results in MAS5 being more suited when just very few samples are available. RMA is thus more suited to analyses which feature many samples. Both methods are provided by the bioconductor `affy` package [13]. The `affy` package only processes older Affymetrix expression arrays. Newer arrays will have to be processed with the `oligo` package, which is also able to process SNP and exon arrays. An improved version of the RMA algorithm exists which uses the probe sequence to give a better normalization. This method is called gcRMA, and is implemented in the `gcrma` package [26]. Affymetrix has published a newer normalization method in BioConductor, which they call "Probe Logarithmic Error Intensity Estimate", and is implemented in the `plier` [27] package.

Illumina BeadArray

The `lumi` BioConductor package provides many methods relating to Illumina BeadArray analysis, including normalization [28]. The package contains several methods of normalizing BeadArray data (the `adjColorBias.quantile)`, `smoothQuantileNormalization`, `adjColorBias.ssn`, `lumiMethyN`, `lumiN`, `normalizeMethylation.quantile`, `normalizeMethylation.ssn`, `rankinvariant`, `rns`, `ssn`, and `normalize.loess` functions. The `lumiExpresso` function does all processing steps automatically.

The `adjColorBias.quantile` and `normalizeMethylation.quantile` functions are wrappers for the `smoothQuantileNormalization` function, which does the actual work. Likewise, functions `adjColorBias.ssn` and `normalizeMethylation.ssn` are wrappers for the `ssn` function.

With quantile normalization, the test sample is "aligned" to a reference sample, in such manner that the test sample will eventually have to same statistical properties as the reference sample [23]. Loess normalization uses local regressions of the log ratios of expression.

Furthermore, the large BioConductor microarray analysis package `limma` can analyse Illumina BeadArray data since version 3.0.0 [14].

Agilent

The package `limma` can also analyze two-color microarray packages such as those used by Agilent [14, 29]. When Agilent arrays have been analyzed with GenePix scanners, they can be loaded with the `limma` function `read.maimages`, upon which they can be normalized with the `normalizeWithinArrays` function [14], using six different normalization methods [15, 29]. The currently supported normalization methods are `loess`, `print-tip loess`, `composite`, `control` and `robust spline` [29].

The `loess` method normalizes the array with local regressions of the general log ratios of expression of the array [29]. The `print-tip loess` method refines this method by using loess regressions based on the print-tip location [29]. The `composite` method uses additional information regarding control spots, if available [29]

Next-generation sequencing

Once the FASTQ files have been acquired, the next step a Next-Generation Sequencing workflow generally involves mapping the acquired reads to a reference genome. The prime tool in existence for this is Bowtie [30]. Two versions of Bowtie exists, Bowtie 1 and Bowtie 2 [31]. Bowtie 1 is mainly useful for NGS machines utilizing small reads (<50 bp/read). Bowtie 2, on the other hand, is the recommended version for NGS machines utilizing reads of more than 50 bp, and therefore most suited to newer machines. Bowtie 2 also handles gapped alignments better than Bowtie 1 [31]. Bowtie 2 is *not* an update of Bowtie 1; rather, it is designed for a different use case. As such, both versions are in active development. An R/bioconductor wrapper for Bowtie exists, and is called `RBowtie` [32]. A BioConductor wrapper for Bowtie2 does not exist as of January 2014.

BowTie and related tools output Sequence Alignment Maps - SAM files. A SAM file is a human-readable format containing alignments. These SAM files can be very large, and are therefore often converted into BAM files, which is a compressed binary format. The main tool to analyse SAM/BAM files is SAMtools [33], which has been partially ported to R/Bioconductor as `RSamtools` [34]. SAMtools is used to sort and index SAM/BAM files. Furthermore, it can convert between the two formats, and it is possible to merge samples. The tool can be used to create sub-alignments; for instance, only those alignments on chromosome 1. As of January 2014, `Rsamtools` can only process BAM files; SAM files are not supported, and conversion from/to BAM files is not possible either. Therefore, users of `Rsamtools` still need the standalone version of `SAMtools`.

de novo sequencing

When no reference genome is available, it is necessary to assemble the genome from scratch. Many assemblers for FASTQ files have been developed, including ABySS [35], ALLPATHS [36], Geneious [37], IDBA [38], MIRA, Newbler, SeqMan NGen, SeqPrep, Sequencher, SOAPdenovo [39], and Velvet [40]

Velvet [40] is one of the most popular assemblers for *de novo* sequencing. The mathematics behind Velvet relies on De Bruijn graphs. Its memory usage is quite high; it requires at least 12GB of RAM memory. It supports most common sequencing formats (FASTA (and compressed version), FASTQ (and compressed version), SAM, BAM, eland and gerald files). Velvet can read from standard input, which means

that it can work directly with the output of another program. This means that it is not necessary to save (very large) input files, but instead go directly to assembly. Velvet - and other De Bruijn assemblers - split sequence reads into hashes of length k, so-called k-mers. The size of k is an important factor in assembler performance. Therefore, it is necessary to run Velvet with a number of k values. A BioConductor package wrapping or implementing Velvet unfortunately does not exist yet. Other assemblers are also not yet supported by BioConductor.

RNAseq

RNA-seq is a special case as it uses the technologies of next-generation sequencing for expression profiling. As such, it has unique workflow that requires a taste of both worlds.

With RNA-seq, the read *counts* are important, as they not only provide information about the quality of the sequence, but also about the *amount* of RNA present for this particular sequence, and thus provides information about the expression state of a particular transcript.

Generally speaking, RNA-seq data must be aligned first, using methods described above. But whereas preprocessing stops with general sequencing, RNA-seq requires an additional microarray-like normalization step. This, firstly introduces some format issues: the main BioConductor package for RNA-seq, `edgeR` [41], requires its input data to be in a tab-delimited file of read counts with gene symbols in one column [42]. This can be supplied by an other BioConductor package, namely the `htSeqTools` quality control package [22].

The `edgeR` package does not try to estimate absolute expression values [41, 42]. Instead, it only tries to estimate relative expression values between samples. This makes normalization somewhat less complicated. For normalization, `edgeR` uses an experimental scaling normalization method called TMM normalization [41–43]. TMM normalization is meant to improve upon earlier normalization methods that tend to use scaling to library size [43–46] or quantile normalization [47].

Apart from `edgeR`, several other RNA-seq analysis tools exist, such as ERANGE [46].

Post-processing

Microarrays

Gene Expression

Both the `affy` and `lumi` packages - respectively targeting Affymetrix and Illumina arrays - and associated packages (such as `affycoretools`) are useful for normalization and data processing, but in themselves do generally not provide functions to calculate differentially expressed genes. The `limma` package is, for many applications, the workhorse calculating differentially expressed genes, making venn diagrams and heatmaps.

Many specialized packages exist that target a specific analysis. These are generally independent from one another. For instance, for Bayesian analysis of differential gene expression - which is not covered in `limma` - 13 different BioConductor packages exist; `BAC` [48, 49], `BGmix` [50], `bgx` [51], `BHC` [52], `birta` [53], `bridge` [54, 55], `iBMQ` [56], `iChip` [57, 58], `iterativeBMA` [59], `iterativeBMAsurv` [60], `mBPCR` [61], `siggenes` [62] and `XDE`. It is beyond the scope of this review to exhaustively list all packages for microarray analysis.

SNP arrays

There are several packages specifically targeted for SNP arrays. For Illumina BeadArray SNP arrays, there is `beadarraySNP`. For Affymetrix GeneChip SNP arrays, there is the `RLMM` package, as well as the `crlmm` package which is able to process both platforms. It uses the Corrected Robust Linear Model with Maximum likelihood distance (CRLMM) method [63–68]. The `crlmm` package is an improvement of the earlier `oligo` package, which also implements the CRLMM method [65]. The `crlmm` package can natively process Affymetrix CEL files.

The `SNPchip` package can be used to visualize copy number alterations and SNPs [69].

SNP array techniques have enabled genome-wide association (GWAS) studies to be conducted on large scales. As such, several BioConductor packages have been developed for GWAS studies. The main BioConductor GWAS package is `GWASTools` [70]. It contains several classes for GWAS data representation, and is able to perform analyses. The installation of `GWASTools` is non-trivial, and depends on the installation of the non-R `NetCDF` program. Outside of BioConductor, the *ABEL family of R packages provides rich tools for analyzing genomics experiments. The GenABEL package, available through CRAN,

provides a comprehensive toolset for GWAS studies [71].

Next-generation sequencing

The BioConductor package `QuasR` [72] is an amalgamation of several R/BioConductor packages for next-generation sequencing. It is meant to provide a single package through which all common steps in NGS analysis can be carried out, from start to end. Core packages utilized by `QuasR` include `IRanges` and `RSamtools`. As of January 2014, it contains functions for the analysis of ChIP-seq, RNA-seq and Bi-seq experiments. As a backend for alignments, it uses either Bowtie [30] or SpliceMap [73] through the `Rbowtie` package. There is clear documentation for `QuasR`, for both inexperienced and experienced R users.

The `GenomicRanges` [74, 75] package, in concurrence with the `IRanges` [76] package, is an important infrastructural package for sequencing data in BioConductor. It supplies a data structure - a `GRanges` object - that allows users to work with sequences.

SNPs

Ultra-high throughput sequencing technologies allow the identification of variants from individual genomes. This essentially allows the detection of rare variants. Unfortunately, not many BioConductor packages targeting variant detection from sequence data have as of yet been developed. Two highly interrelated data infrastructure packages, `SeqArray` [77] and `SeqVarTools` [78] have so far been developed. These packages allow handling variant data without requiring large amounts of RAM memory. In stead, they store the (compressed) data on disk.

Other packages aiming to *detect* variants are `hapFabia` [79] and `deepSNV` [80]. The `deepSNV` package aims to detect sub-clonal variants from ultra-deep sequencing data.

RNA-seq

RNA-seq is a potentially far more powerful technique to measure gene expression than microarray. Whereas background noise is persistent in microarray platforms, this is almost non-existing with RNA-seq, resulting in a very high dynamic range [81]. Furthermore, it opens up the possibility of detecting previously unknown splice variants, which is impossible with probe-based platforms such as microarrays.

Tools to discover *de novo* splice junctions from RNA-seq data are available, for BioConductor users most prominently in the form of `SpliceMap`. This works by alignment two split reads of 25 bases each with Bowtie. The unaligned half-reads are then used to determine the locations of previously unmapped splice junctions [73]. Interestingly, SpliceMap has been included in the `RBowtie` package [32]. It outputs data in SAM/BAM format. Several other non-R tools exist to detect splice junctions, including ABMapper [82]

A comprehensive RNA-seq analysis tool is Cufflinks [83–85], which - apart from calculating differential expression and regulation - can assemble transcripts. It uses the TopHat gapped read aligner [85, 86]. BioConductor packages `cummeRbund` [83] and `spliceR` [87] can be used to visualize the results of Cufflinks.

Two high-level BioConductor packages for the detection of differential expression of RNA-seq data are `DESeq` [88, 89] and `DEXSeq` [90, 91]. `DESeq` is useful for detecting differential expression at the gene level, whereas `DEXSeq` is useful for detecting differential expression at the exon level. Both packages were developed by the same lab. The input required by `DESeq` is a simple table with sequence read count per gene (rows) per sample (columns). Such a data structure can be constructed by using the `GenomicRange` infrastructure package. `DEXSeq`, however, requires the use of a Python package and a species-specific "General Transfer Format" (GTF) annotation file downloaded from Ensembl to build the data structure required. Both packages then normalize the data and call differentially expressed genes. These two packages are richly documented.

ChIP-seq

ChIP-seq assays - and other affinity assays, such as MeDIP-seq - feature peak calling. In this setup, "peaks" of bound protein to some genomic region are called. In a typical ChIP-seq analysis, this peak calling occurs after alignment of sequences. See figure 2 for a schematic of a typical ChIP-seq analysis.

Several BioConductor packages exist for the analysis of ChIP-seq data. The most obviously named of these are `chipseq` and `ChIPseqR` [92], but many others exist: `BayesPeak` [93, 94], `CSAR` [95], `DBChIP` [96], `DiffBind` [97, 98], `iSeq` [57, 58, 99], `jmosaics` [100], `mosaics` [101, 102], `nucleR` [103] and `triform` [104].

The three most popular packages (as determined by BioConductor download statistics) are `chipseq`, `DiffBind` and `BayesPeak`.

`DiffBind` is a useful package for identifying dif-

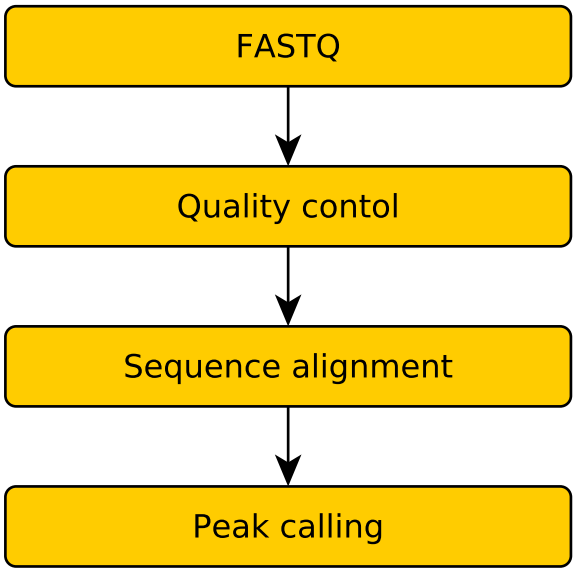

Figure 2: *Workflow diagram for a typical ChIP-seq analysis*

ferentially bound sites on the genome. It is thus primarily useful for ChIP-seq experiments which two or multiple states (e.g. to screen for drug-induced alterations in protein affinity to certain genomic sites). It borrows some statistical models from RNA-seq tools such as `edgeR`, since ChIP-seq also relies on count data as RNA-seq does. It also contains several plot functions, such as heatmaps, to visualize ChIP-seq results. It requires the use of both BAM files and a csv file containing so-called "peaksets" - i.e. a csv file containing sets of candidate binding sites in the form of genomic intervals. These are then internally converted into a GRanges object.

Discussion

BioConductor is a rich source of R packages for both microarray and next-generation sequencing platforms. As of January 2014, BioConductor contains packages for both microarray and NGS analysis. Overall, microarray analysis is still BioConductor's main focus point. However, in recent years, next-generation sequencing tools have been added to BioConductor, together with NGS infrastructure packages such as `GenomicRanges` and `IRanges`, making BioConductor a useful repository for NGS analysis as well.

The analysis of Microarray experiments tends to be separated over many packages, each tailored to a specific set of problems. This reflects UNIX philosophy where the common paradigm is that programs

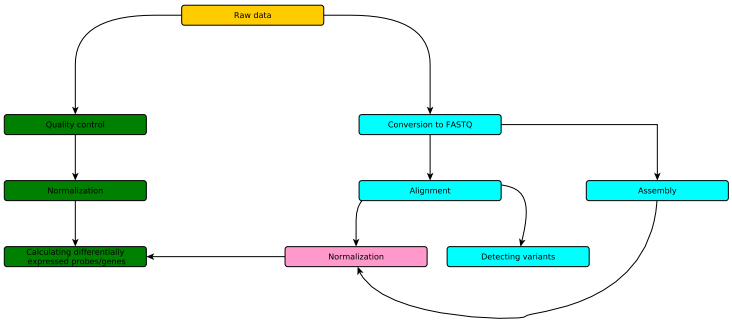

Figure 3: *Workflows for microarray (green) and NGS (blue) analysis. RNA-seq highlighted in pink*

should do only one thing and do it well. The analysis of next-generation sequencing data - while requiring more non-R tools - is generally more focused on a small number of central packages which provide a total work flow (such as `QuasR` and `edgeR`). Surely this makes for better user-friendliness.

There are several areas of note which require more adequate thought. Issues that might well harm BioConductor in the short or long term are areas of compatibility, technical problems in R itself, and community (or lack thereof) issues.

Compatibility

One major area of concern in comparing microarray tools and next-generation sequencing tools is the incompatibility of data formats. Especially in the microarray community, standardization is lacking. Many different file and data formats exist, which makes interoperability complex and confusing. This is partly due to inherent technological differences between different vendors. In a microarray setup, probes have to be mapped to fluorescence values. Vendors differ tremendously in probes or probesets, which makes comparing microarray data from different vendors inherently complex.

Sequencing data, on the other hand, has a clear final data output: a DNA sequence with some quality score. This is eventually vendor-independent, making standardization relatively easy. Nevertheless, even here, file formats have not been entirely standardized. FASTQ files still can come in different encodings for quality scores.

The lack of standardization is undoubtedly harmful for software development and analysis of high-throughput data.

BioConductor's `Biobase` meta-package supplies an `ExpressionSet` object (commonly referred to as an *eSet*), which aims to provide a single data structure for microarray analysis packages [1, 105].

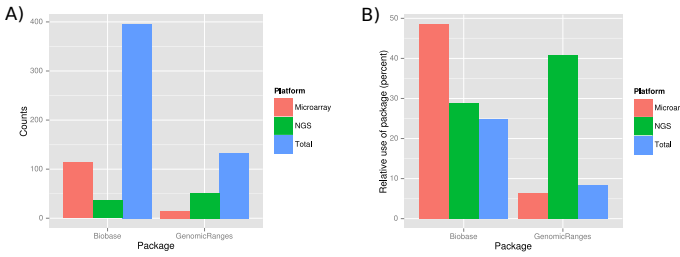

Figure 4: *Use of Biobase and GenomicRange packages in BioConductor. A: Absolute number of analysis packages using Biobase or GenomicRange. B: Relative amount of packages using Biobase or GenomicRange, as a percentage of total amount of analysis package for Microarray or NGS*

The `ExpressionSet` class natively works well for Affymetrix data, but additions are required for the `lumi` package [28, 106]. Furthermore, the popular `limma` package does not use `ExpressionSet` objects internally, although it can process them. Interally, `limma` uses a class called `RGList`.

For NGS analysis, the `GenomicRange` package supplies the `GRanges` class. This class can contain sequences and ranges of sequences [74, 75], making it ideal for any sequencing tools.

BioConductor does not enforce the use `Biobase` or `GenomicRanges`. The adoption of these data standards is therefore up to the developers of each individual package. The result is that most packages do not support those two core packages. Using the `Depends On Me` and `Imports Me` fields in the BioConductor pages for `Biobase` and `GenomicRanges`, it is possible to find those packages that use these packages. As can be seen in figure 4, overall adoption of `Biobase` is almost 50 percent for microarray analysis packages, whereas 25 percent of NGS packages use `Biobase`. Conversely, slightly more than 40 percent of NGS packages use `GenomicRanges`, whereas the use of this package by microarray analysis packages is, at 6 percent, very small. As `GenomicRanges` is a younger package than `Biobase`, it is to be expected that use of `GenomicRanges` will increase over time.

The diversity of microarray chips requires adequate annotation for each individual chips. This is currently the easiest for Affymetrix arrays, as Affymetrix supplies Chip Description Files (CDF) for each platform. Earlier Affymetrix chips have a somewhat complicated annotation infrastructure in BioConductor, requiring three different packages per platform (a cdf package, a probe package and an annotation package). In more recent years, in concurrence with the more general `oligo` package, this has been streamlined, with the advent of Platform Design Information (pdInfo) packages. The annotation information contained in Affymetrix CDF files is based on UniGene information available to Affymetrix at the time of chip creation. Over the years, UniGene information has been refined tremendously, requiring CDF files to be updated [107]. Affymetrix does not provide updated CDF files. As such, some independent projects exist to update CDF files, the most prominent of which is the "BrainArray Custom CDF" project. These custom CDF files can then be loaded in a so-called "CDF environment" by the `makecdfenv` package, which can be used in concurrence with `affy`, or one can use the `pdInfoBuilder` package to create a pdInfo package for use with `oligo`. Illumina annotation is provided by the `lumiHumanAll.db`, `lumiHumanIDMapping` and `illuminaHumanv1BeadID.db` packages, which provide mappings of Illumina IDs to reference genome IDs. These can then be read with the generic `annotate` package. For species other than human, it is necessary to use the nuID system [12, 108].

R

Some aspects of R make it rather difficult to work with from a next-generation sequencing point of view. The R language traditionally puts all data in working memory. When large files are read with R, the entire contents of said file enter the working memory. This makes R very memory intensive. For microarrays, this is generally a smaller issue, as array files are not too large. FASTQ and SAM/BAM files used for NGS, on the other hand, can be on the order of hundreds of gigabytes. Loading all this data into RAM memory is impossible on anything smaller than a cluster. Two proprietary versions of R - Oracle R Enterprise and Revolution R Enterprise - exist that aim to solve this problem.

To receive updates for packages, BioConductor requires its users to have the most recent version of R, which as of January 2014 is R 3.0.2. R version 3.0 was released in April 2013. R version 3.0, while requiring the re-installation of all packages, contains some important updates of interest to large datasets. In R versions 2.x and lower, the maximum vector length was $2^{31} - 1$, (roughly 2.1 billion elements). As a matrix in R is a single vector (unlike a data frame, which is a list of vectors), this had some serious implications. Namely, the maximum dimensions of an n-by-n matrix are just 46,430 by 46,430 elements. These dimensions are easily achieved by analysis of microarray or next-generation sequencing data

sets. In R 3.0, the maximum size of a vector has been raised to 2^{52} elements (roughly 4.5 quadrillion elements). The maximum size of an n-by-n matrix is now with $n = 94{,}906{,}266$. The memory requirements for such a vector are, at over 33TB, however, gigantic.

As the amount of generated data by biologic experiments grows phenomenally, systems that attempt to put all this data in RAM become unsustainable. If R and BioConductor want to stay in the game, some solution will have to be found. One solution that does not require a complete overhaul of R is the use of some form of compression. Some attempts at this have already been done. The `IRanges` package, which itself forms the basis of the `GenomicRanges` meta-package, uses run-length encoding to compress sequences [76]. Since sequencing data is at core simple text, a large part of which is highly repetitive, it is fortunately highly compressible: a 30 gigabyte FASTQ file compressed with gzip will generally result in a file on the order of 10GB, thus having a compression ratio of approximately 60-70%. Using modern compressors such as xz, compression ratios around 80% should be attainable.

Community

One requirement of BioConductor packages is that they are Open Source. This means that, in principle, anyone can modify the code. Many Open Source projects feature an active community, where bugs are reported and fixed by members other than the original authors. Unfortunately, Bioconductor has no community-driven approach. Access to repositories is only given to the original author(s) of the package. While write-access to scientific software is in all likelihood not a smart idea, having the option to at least suggest features, upgrades, bug-fixes etcetera in a user-friendly way would most likely enhance usability of BioConductor products. This makes modern bug reporting practically non-existent, apart from a cumbersome mailing list system. BioConductor could potentially learn from the R-Forge project, where many CRAN R packages are maintained in a more community-based system.

References

1. Gentleman, R. C. *et al.* Bioconductor: open software development for computational biology and bioinformatics. *Genome biology* **5,** R80 (2004).
2. Dalma-Weiszhausz, D. D., Warrington, J., Tanimoto, E. Y. & Miyada, C. G. [1] The Affymetrix GeneChip® Platform: An Overview. *Methods in enzymology* **410,** 3–28 (2006).
3. Sboner, A., Mu, X. J., Greenbaum, D., Auerbach, R. K., Gerstein, M. B., *et al.* The real cost of sequencing: higher than you think. *Genome Biol* **12,** 125 (2011).
4. Hayden, E. C. *Is the $1000 genome for real?* Jan. 2014. <http://www.nature.com/news/is-the-1-000-genome-for-real-1.14530>.
5. <http://my454.com/products/technology.asp>.
6. Berka, J. *et al. Bead Emulsion Nucleic Acid Amplification* US Patent 20,130,078,638. Mar. 2013.
7. <http://res.illumina.com/documents/products/techspotlights/techspotlight_sequencing.pdf>.
8. Liu, L. *et al.* Comparison of next-generation sequencing systems. *Journal of Biomedicine and Biotechnology* **2012** (2012).
9. <http://www.lifetechnologies.com/nl/en/home/life-science/sequencing/next-generation-sequencing/solid-next-generation-sequencing/solid-next-generation-sequencing-systems-reagents-accessories/solid-next-generation-sequencing-chemistry.html#>.
10. Quail, M. A. *et al.* A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC genomics* **13,** 341 (2012).
11. Eid, J. *et al.* Real-time DNA sequencing from single polymerase molecules. *Science* **323,** 133–138 (2009).
12. Ritchie, M. E., Dunning, M. J., Smith, M. L., Shi, W. & Lynch, A. G. BeadArray expression analysis using bioconductor. *PLoS computational biology* **7,** e1002276 (2011).
13. Gautier, L., Cope, L., Bolstad, B. M. & Irizarry, R. A. affy-analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics* **20,** 307–315 (2004).

14. Smyth, G. K. in *Bioinformatics and computational biology solutions using R and Bioconductor* 397–420 (Springer, 2005).

15. Ritchie, M. E. *et al.* A comparison of background correction methods for two-colour microarrays. *Bioinformatics* **23,** 2700–2707 (2007).

16. Cock, P. J., Fields, C. J., Goto, N., Heuer, M. L. & Rice, P. M. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic acids research* **38,** 1767–1771 (2010).

17. Kauffmann, A., Gentleman, R. & Huber, W. arrayQualityMetrics - a bioconductor package for quality assessment of microarray data. *Bioinformatics* **25,** 415–416 (2009).

18. Wu, X. & Li, X. A. *ArrayTools: Array Quality Assessment and Analysis Tool* Oct. 2013. <http://www.bioconductor.org/packages/release/bioc/vignettes/ArrayTools/inst/doc/ArrayTools.pdf>.

19. Parman, C. & Halling, C. affyQCReport: A Package to Generate QC Reports for Affymetrix Array Data. *Bioconductor Package Help Documentation* (2008).

20. Wilson, C. L. & Miller, C. J. Simpleaffy: a BioConductor package for Affymetrix Quality Control and data analysis. *Bioinformatics* **21,** 3683–3685 (2005).

21. Bolstad, B. *affyPLM: Model Based QC Assessment of Affymetrix GeneChips* 2006. <http://www.bioconductor.org/packages/release/bioc/vignettes/affyPLM/inst/doc/QualityAssess.pdf>.

22. Planet, E., Attolini, C. S.-O., Reina, O., Flores, O. & Rossell, D. htSeqTools: high-throughput sequencing quality control, processing and visualization in R. *Bioinformatics* **28,** 589–590 (2012).

23. Bolstad, B. M., Irizarry, R. A., Åstrand, M. & Speed, T. P. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19,** 185–193 (2003).

24. Irizarry, R. A. *et al.* Summaries of Affymetrix GeneChip probe level data. *Nucleic acids research* **31,** e15–e15 (2003).

25. Irizarry, R. A. *et al.* Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **4,** 249–264 (2003).

26. Wu, Z. & Irizarry, R. *Description of gcrma* 2005. <http://www.bioconductor.org/packages/2.14/bioc/vignettes/gcrma/inst/doc/gcrma2.0.pdf>.

27. Inc., A., Miller, C. J. & PICR. *plier reference manual* 2014. <http://www.bioconductor.org/packages/release/bioc/manuals/plier/man/plier.pdf>.

28. Du, P., Kibbe, W. A. & Lin, S. M. lumi: a pipeline for processing Illumina microarray. *Bioinformatics* **24,** 1547–1548 (2008).

29. Smyth, G. K. & Speed, T. Normalization of cDNA microarray data. *Methods* **31,** 265–273 (2003).

30. Langmead, B., Trapnell, C., Pop, M., Salzberg, S. L., *et al.* Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* **10,** R25 (2009).

31. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nature methods* **9,** 357–359 (2012).

32. Hahne, F., Lerch, A. & Stadler, M. Rbowtie: A r wrapper for bowtie and splicemap short read aligners. *URL http://bioconductor.org/packages/release/bioc/html/Rbowtie.html* (2012).

33. Li, H. *et al.* The sequence alignment/map format and SAMtools. *Bioinformatics* **25,** 2078–2079 (2009).

34. Morgan, M. & Pages, H. Rsamtools: Import aligned BAM file format sequences into R/Bioconductor. *URL http://bioconductor. org/-packages/release/bioc/html/Rsamtools. html. R package version* **1** (2010).

35. Simpson, J. T. *et al.* ABySS: a parallel assembler for short read sequence data. *Genome research* **19,** 1117–1123 (2009).

36. Butler, J. *et al.* ALLPATHS: de novo assembly of whole-genome shotgun microreads. *Genome research* **18,** 810–820 (2008).

37. Kearse, M. *et al.* Geneious Basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics* **28,** 1647–1649 (2012).

38. Peng, Y., Leung, H. C., Yiu, S.-M. & Chin, F. Y. *IDBA–a practical iterative de Bruijn graph de novo assembler* in *Research in Computational Molecular Biology* (2010), 426–440.

39. Li, R. *et al.* De novo assembly of human genomes with massively parallel short read sequencing. *Genome research* **20,** 265–272 (2010).

40. Zerbino, D. R. & Birney, E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome research* **18,** 821–829 (2008).

41. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26,** 139–140 (2010).

42. Robinson, M., McCarthy, D., Chen, Y. & Smyth, G. K. edgeR: differential expression analysis of digital gene expression data User's Guide (2011).

43. Robinson, M. D. & Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol* **11,** R25 (2010).

44. Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M. & Gilad, Y. RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research* **18,** 1509–1517 (2008).

45. Sultan, M. *et al.* A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science* **321,** 956–960 (2008).

46. Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L. & Wold, B. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature methods* **5,** 621–628 (2008).

47. Cloonan, N. *et al.* Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nature methods* **5,** 613–619 (2008).

48. Johnson, W. E. *et al.* Model-based analysis of tiling-arrays for ChIP-chip. *Proceedings of the National Academy of Sciences* **103,** 12457–12462 (2006).

49. Gottardo, R., Li, W., Johnson, W. E. & Liu, X. S. A flexible and powerful Bayesian hierarchical model for ChIP–chip experiments. *Biometrics* **64,** 468–478 (2008).

50. Lewin, A., Bochkina, N. & Richardson, S. Fully Bayesian mixture model for differential gene expression: simulations and model checks. *Statistical applications in genetics and molecular biology* **6** (2007).

51. Hein, A.-M. K., Richardson, S., Causton, H. C., Ambler, G. K. & Green, P. J. BGX: a fully Bayesian integrated approach to the analysis of Affymetrix GeneChip data. *Biostatistics* **6,** 349–373 (2005).

52. Savage, R. *et al.* R/BHC: fast Bayesian hierarchical clustering for microarray data. *BMC bioinformatics* **10,** 242 (2009).

53. Zacher, B. *et al.* Joint Bayesian inference of condition-specific miRNA and transcription factor activities from combined gene and microRNA expression data. *Bioinformatics* **28,** 1714–1720 (2012).

54. Gottardo, R., Raftery, A. E., Yeung, K. Y. & Bumgarner, R. E. Quality control and robust estimation for cdna microarrays with replicates. *Journal of the American Statistical Association* **101** (2006).

55. Gottardo, R., Raftery, A. E., Yee Yeung, K. & Bumgarner, R. E. Bayesian robust inference for differential gene expression in microarrays with multiple samples. *Biometrics* **62,** 10–18 (2006).

56. Scott-Boyer, M. P. *et al.* An Integrated Hierarchical Bayesian Model for Multivariate eQTL Mapping. *Statistical Applications in Genetics and Molecular Biology* **11** (2012).

57. Mo, Q. & Liang, F. Bayesian Modeling of ChIP-chip Data Through a High-Order Ising Model. *Biometrics* **66,** 1284–1294 (2010).

58. Mo, Q. & Liang, F. A hidden Ising model for ChIP-chip data analysis. *Bioinformatics* **26,** 777–783 (2010).

59. Yeung, K. Y., Bumgarner, R. E. & Raftery, A. E. Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics* **21,** 2394–2402 (2005).

60. Annest, A., Bumgarner, R. E., Raftery, A. E. & Yeung, K. Y. Iterative bayesian model averaging: A method for the application of survival analysis to high-dimensional microarray data. *BMC bioinformatics* **10,** 72 (2009).

61. Rancoita, P., Hutter, M., Bertoni, F. & Kwee, I. Bayesian DNA copy number analysis. *BMC bioinformatics* **10,** 10 (2009).

62. Schwender, H., Krause, A. & Ickstadt, K. Identifying interesting genes with siggenes. *The Newsletter of the R Project Volume 6/5, December 2006* **34,** 45 (2006).

63. Carvalho, B. S. & Irizarry, R. A. A framework for oligonucleotide microarray preprocessing. *Bioinformatics* **26,** 2363–2367 (2010).

64. Carvalho, B., Bengtsson, H., Speed, T. P. & Irizarry, R. A. Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. *Biostatistics* **8,** 485–499 (2007).

65. Scharpf, R. B., Irizarry, R., Ritchie, W., Carvalho, B. & Ruczinski, I. Using the R package crlmm for genotyping and copy number estimation (2010).

66. Carvalho, B., Louis, T. A. & Irizarry, R. A. Quantifying uncertainty in genotype calls (2009).

67. Ritchie, M. E., Carvalho, B. S., Hetrick, K. N., Tavaré, S. & Irizarry, R. A. R/Bioconductor software for Illumina's Infinium whole-genome genotyping BeadChips. *Bioinformatics* **25,** 2621–2623 (2009).

68. Scharpf, R. *et al.* A multilevel model to address batch effects in copy number estimation using SNP arrays. *Biostatistics (Oxford, England)* **12,** 33–50 (2011).

69. Scharpf, R. B., Ting, J. C., Pevsner, J. & Ruczinski, I. SNPchip: R classes and methods for SNP array data. *Bioinformatics* **23,** 627–628 (2007).

70. Gogarten, S. M. *et al.* GWASTools: an R/Bioconductor package for quality control and analysis of Genome-Wide Association Studies. *Bioinformatics* **28,** 3329–3331 (2012).

71. Aulchenko, Y. S., Ripke, S., Isaacs, A. & Van Duijn, C. M. GenABEL: an R library for genome-wide association analysis. *Bioinformatics* **23,** 1294–1296 (2007).

72. Lerch, A., Gaidatzis, D. & Stadler, M. An Introduction to QuasR.

73. Au, K. F., Jiang, H., Lin, L., Xing, Y. & Wong, W. H. Detection of splice junctions from paired-end RNA-seq data by SpliceMap. *Nucleic Acids Research* **38,** 4570–4578 (2010).

74. Aboyoun, P., Pages, H. & Lawrence, M. *GenomicRanges: Representation and manipulation of genomic intervals* R package version 1.14.4 ().

75. Lawrence, M. *et al.* Software for computing and annotating genomic ranges. *PLoS computational biology* **9,** e1003118 (2013).

76. Pages, H., Aboyoun, P. & Lawrence, M. *IRanges: Infrastructure for manipulating intervals on sequences* R package version 1.20.6 ().

77. Zheng, X., Gogarten, S. M. & Laurie, C. *SeqArray: Big Data Management of Genome-wide Sequencing Variants* R package version 1.2.0 (2013). <http://corearray.sourceforge.net/tutorials/SeqArray/>.

78. Gogarten, S. M. & Zheng, X. *SeqVarTools: Tools for variant data* R package version 1.0.0 ().

79. Hochreiter, S. HapFABIA: Identification of very short segments of identity by descent characterized by rare variants in large sequencing data. *Nucleic acids research* **41,** e202–e202 (2013).

80. Gerstung, M. *et al.* Reliable detection of subclonal single-nucleotide variants in tumour cell populations. *Nature communications* **3,** 811 (2012).

81. Lou, S.-K. *et al.* Detection of splicing events and multiread locations from RNA-seq data based on a geometric-tail (GT) distribution of intron length. *BMC bioinformatics* **12,** S2 (2011).

82. Lou, S.-K. *et al.* ABMapper: a suffix array-based tool for multi-location searching and splice-junction mapping. *Bioinformatics* **27,** 421–422 (2011).

83. Trapnell, C. *et al.* Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature protocols* **7,** 562–578 (2012).

84. Roberts, A., Pimentel, H., Trapnell, C. & Pachter, L. Identification of novel transcripts in annotated genomes using RNA-Seq. *Bioinformatics* **27,** 2325–2329 (2011).

85. Trapnell, C. *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology* **28,** 511–515 (2010).

86. Robertson, G. *et al.* De novo assembly and analysis of RNA-seq data. *Nature methods* **7,** 909–912 (2010).

87. Vitting-Seerup, K., Porse, B. T., Sandelin, A. & Waage, J. E. *spliceR: An R package for classification of alternative splicing and prediction of coding potential from RNA-seq data.* tech. rep. (PeerJ PrePrints, 2013).

88. Anders, S. & Huber, W. Differential expression analysis for sequence count data. *Genome Biol* **11,** R106 (2010).

89. Anders, S. & Huber, W. Differential expression of RNA-Seq data at the gene level–the DESeq package. <http://www.bioconductor.org/packages/release/bioc/vignettes/DESeq/inst/doc/DESeq.pdf> (2012).

90. Anders, S., Reyes, A. & Huber, W. Detecting differential usage of exons from RNA-seq data. *Genome research* **22,** 2008–2017 (2012).

91. Reyes, A., Anders, S. & Huber, W. Analyzing RNA-seq data for differential exon usage with the DEXSeq package. <http://www.bioconductor.org/packages/release/bioc/vignettes/DEXSeq/inst/doc/DEXSeq.pdf> (2012).

92. Humburg, P., Helliwell, C. A., Bulger, D. & Stone, G. ChIPseqR: analysis of ChIP-seq experiments. *BMC bioinformatics* **12,** 39 (2011).

93. Spyrou, C., Stark, R., Lynch, A. G. & Tavaré, S. BayesPeak: Bayesian analysis of ChIP-seq data. *BMC bioinformatics* **10,** 299 (2009).

94. Cairns, J. *et al.* BayesPeak-an R package for analysing ChIP-seq data. *Bioinformatics* **27,** 713–714 (2011).

95. Muiño, J. M., Kaufmann, K., van Ham, R., Angenent, G. C., Krajewski, P., *et al.* ChIP-seq Analysis in R (CSAR): An R package for the statistical detection of protein-bound genomic regions. *Plant Methods* **7** (2011).

96. Liang, K. & Keleş, S. Detecting differential binding of transcription factors with ChIP-seq. *Bioinformatics* **28,** 121–122 (2012).

97. Stark, R. & Brown, G. *DiffBind: differential binding analysis of ChIP-Seq peak data* 2013.

98. Ross-Innes, C. S. *et al.* Differential oestrogen receptor binding is associated with clinical outcome in breast cancer. *Nature* **481,** 389–393 (2012).

99. Mo, Q. A fully Bayesian hidden Ising model for ChIP-seq data analysis. *Biostatistics* **13,** 113–128 (2012).

100. Zeng, X. *et al.* jMOSAiCS: joint analysis of multiple ChIP-seq datasets. *Genome Biology* **14,** R38 (2013).

101. Kuan, P. F. *et al.* A statistical framework for the analysis of ChIP-Seq data. *Journal of the American Statistical Association* **106,** 891–903 (2011).

102. Sun, G., Chung, D., Liang, K. & Keleş, S. in *Deep Sequencing Data Analysis* 193–212 (Springer, 2013).

103. Flores, O. & Orozco, M. nucleR: a package for non-parametric nucleosome positioning. *Bioinformatics* **27,** 2149–2150 (2011).

104. Kornacker, K., Rye, M., Håndstad, T. & Drabløs, F. The Triform algorithm: improved sensitivity and specificity in chip-seq peak finding. *BMC bioinformatics* **13,** 176 (2012).

105. Falcon, S., Morgan, M. & Gentleman, R. *An introduction to Biocinductor's ExpressionSet class* 2007. <http://www.bioconductor.org/packages/2.14/bioc/vignettes/Biobase/inst/doc/ExpressionSetIntroduction.pdf>.

106. Du, P., Kibbe, W. A. & Lin, S. *Using lumi, a package processing Illumina Microarray* 2007. <http://www.bioconductor.org/packages/release/bioc/vignettes/lumi/inst/doc/lumi.pdf>.

107. Dai, M. *et al.* Evolving gene/transcript definitions significantly alter the interpretation of GeneChip data. *Nucleic acids research* **33,** e175–e175 (2005).

108. Du, P., Kibbe, W. A. & Lin, S. M. nuID: a universal naming scheme of oligonucleotides for illumina, affymetrix, and other microarrays. *Biol Direct* **2,** 16 (2007).