Saettone *et al. BMC Bioinformatics* (2019) 20:533
https://doi.org/10.1186/s12859-019-3100-2

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

RACS: rapid analysis of ChIP-Seq data for contig based genomes

Alejandro Saettone[1†], Marcelo Ponce[2†], Syed Nabeel-Shah[3] and Jeffrey Fillingham[1*]

Abstract

Background: Chromatin immunoprecipitation coupled to next generation sequencing (ChIP-Seq) is a widely-used molecular method to investigate the function of chromatin-related proteins by identifying their associated DNA sequences on a genomic scale. ChIP-Seq generates large quantities of data that is difficult to process and analyze, particularly for organisms with a contig-based sequenced genomes that typically have minimal annotation on their associated set of genes other than their associated coordinates primarily predicted by gene finding programs. Poorly annotated genome sequence makes comprehensive analysis of ChIP-Seq data difficult and as such standardized analysis pipelines are lacking.

Results: We present a one-stop computational pipeline, "Rapid Analysis of ChIP-Seq data" (RACS), that utilizes traditional High-Performance Computing (HPC) techniques in association with open source tools for processing and analyzing raw ChIP-Seq data. RACS is an open source computational pipeline available from any of the following repositories https://bitbucket.org/mjponce/RACS or https://gitrepos.scinet.utoronto.ca/public/?a=summary&p=RACS. RACS is particularly useful for ChIP-Seq in organisms with contig-based genomes that have poor gene annotation to aid protein function discovery.
To test the performance and efficiency of RACS, we analyzed ChIP-Seq data previously published in a model organism *Tetrahymena thermophila* which has a contig-based genome. We assessed the generality of RACS by analyzing a previously published data set generated using the model organism *Oxytricha trifallax*, whose genome sequence is also contig-based with poor annotation.

Conclusions: The RACS computational pipeline presented in this report is an efficient and reliable tool to analyze genome-wide raw ChIP-Seq data generated in model organisms with poorly annotated contig-based genome sequence. Because RACS segregates the found read accumulations between genic and intergenic regions, it is particularly efficient for rapid downstream analyses of proteins involved in gene expression.

Keywords: Chromatin immunoprecipitation, Next generation sequencing, Bioinformatics pipeline, High-performance computing, *Tetrahymena thermophila*

Background

In the last few years, traditional HPC centers, such as SciNet at the University of Toronto [1], have been witnessing the emergence of increasing amounts of work-flows from non-typical disciplines in the field of computational science [2]. Among those, disciplines related to bioinformatics appear to be the most prominent in terms of demanding resources and tackling complex biological questions an example of which related to the understanding of the mechanisms underlying transcription. Some of these biological questions are being answered by Next Generation Sequencing (NGS). For example, NGS-based methodologies are helping to address biological questions including the human genome project [3], the human microbiome project [4], RNA-Seq to analyze gene expression [5, 6] and Chromatin immunoprecipitation coupled to NGS (ChIP-Seq) to assess global DNA-binding sites [5, 6].

*Correspondence: jeffrey.fillingham@ryerson.ca
†Alejandro Saettone and Marcelo Ponce contributed equally to this work.
[1]Department of Chemistry and Biology, Ryerson University, 350 Victoria St, M5B 2K3 Toronto, Canada
Full list of author information is available at the end of the article

The advantage of these NGS methodologies for researchers is that high-throughput sequencing allows millions of DNA molecules to be read at the same time [7–9]. The output of NGS is therefore substantial and can be overwhelming for analyses [10, 11]. These analyses are facilitated in model organisms that feature well-annotated genomes such as humans and yeast where genomic sequence is presented in full chromosomal form, the DNA sequence of which can be found as individual files. These genomes have available annotation files that depict the chromosome-specific DNA base pair coordinates of cis-acting DNA sequences including, open reading frames (ORFs), untranslated regions, transcription start sites, and promoter sequences as well as information about the genes themselves taken from the scientific literature making the interpretation of ChIP-Seq data of transcription proteins more accessible. The difficulties during NGS analyses can be compounded if the genome under study is presented as contig-based (contiguous) sequence assemblies, as is the case in the model ciliates *T.thermophila* and *O.trifallax*. A contig-based genome sequence is structured and presented as a basic assembly of consensus regions based on overlapping DNA sequences obtained from DNA sequencing. Contig-based genome sequences are usually available as a conglomerate of individual contigs in a large file. These genome sequences frequently provide files with minimal annotations of predicted genes usually reflecting the lack of available information in the literature.

ChIP-Seq is used in gene expression studies to make predictions about the function(s) a protein in transcription based on its position within a gene [9, 12]. For example, if the Protein of Interest (POI) accumulates within genes rather than intergenic regions, we could infer that it might have a direct role in transcription regulation. An enrichment of the ChIP peaks near the 5'UTRs would suggest that the POI likely functions in transcription initiation. On the other hand, accumulation of ChIP peaks at the 3' ends would suggest a role in transcription termination while proteins involved in elongation are typically found throughout the coding region. Note this is only a first approximation since gene expression can also be coordinated by elements that are not in close proximity to the specific gene [13].

To determine POI position(s) within a genome from raw ChIP-Seq data, the files containing gene coordinates are needed. It is important to note that less developed genomes such as that of *T.thermophila* and *O.trifallax* provide files containing the predicted coordinates for gene positions as minimum annotation. Current ChIP-Seq applications such as MACS2 [14] do not directly address whether the accumulation of the POI is in a specific area such as genic or intergenic region. To obtain a genome file that can be used by a software like MACS2 many other computational steps are required. After the initial alignment, the data is typically analyzed by a peak calling software, such as MACS2, which provide with peaks coordinates. The user then needs to further process the peaks obtained with third-party softwares such as BEDTools [15] to assess the local enrichment within genic and/or intergenic regions.

Our computational pipeline **R**apidly **A**nalyze **C**hIP-**S**eq data (RACS) can be used for any genome that has files containing coordinate sequences of interest. Our pipeline provides a unified tool to perform comprehensive ChIP-Seq data analysis. For instance, with RACS users obtain the co-ordinates of ChIP peaks as well as information regarding their relative enrichment across the genome, i.e. number of significant peaks found with genic versus non-genic regions. We suggest that RACS is a versatile computation pipeline suitable to analyze ChIP-Seq data generated using any model organism.

RACS pipeline implementation

In this work, we describe and demonstrate the utility of the RACS pipeline using two ChIP-Seq data sets generated in two different model organisms including *T.thermophila* and *O.trifallax*. The *T. thermophila* ChIP-Seq data set originates from our recent study [16] on the Ibd1 protein that we found to be a component of multiple chromatin remodeling complexes and localized mainly to highly transcribed genes. Here, we used RACS to refine the Ibd1 ChIP-Seq analysis by subtracting data from an untagged control sample. The *O.trifallax* data set is derived from a study that suggests that RNA Polymerase II (RNAPII) is involved on genome-wide nanochromosome transcription during development [17]. RACS analysis gives results comparable to the reported ChIP-Seq data for *O.trifallax* RNAPII supporting the use of RACS as a generic pipeline.

The RACS pipeline is an open source set of shell and R scripts, which are organized in three main categories:

- the *core pipeline* tools, which allow the user to compute reads differentiating between genic and intergenic regions automatically
- auxiliary *post-processing* scripts[1] for normalization using the "Cluster Passing Filtering" (PF) values
- and *utilities* to validate results by visualizing the reads accumulation and run comparisons with other software tools, such as IGV and MACS2 respectively.

The RACS repository includes the core or main scripts placed in the "core" directory. The comparison and auxiliary tools are placed in a "tools" directory. We have also included examples of submission scripts in the "hpc"

[1] Alternatively, we have also included an *auxiliary spreadsheets* that could be used instead of the script to perform the post-processing and normalization manually, as well as, to check against negative controls.

directory, with PBS [18, 19] and SLURM [20, 21] examples of submission scripts, so that users with access to HPC resources can take advantage of them. Additionally, we have included a "datasets" directory containing scripts that allow the user to download the data used in these analyses. Details about the pipeline implementation and how to use it are included in the 'README' file available within the RACS repositories. A generic top-down overview of the pipeline implementation for the data analysis, is shown in Fig. 1.

The RACS pipeline will run in any standard workstation with a Linux-type operating system. In addition, the following open source tools are needed by the RACS core scripts:

- Burrows-Wheeler Alignment (BWA) version 0.7.13 [22]
- Sequence Alignment/Map (SAMtools) version 1.3.1 [23]
- the R statistical language [24]

Our pipeline is open source, and the scripts are available to download and accessible from public repositories[2].

The pipeline requires as input the *.fastq* files (obtained from NGS) from the ChIP-Seq experiments and the specific genome assembly files and a file containing the gene annotations (e.g. *.gff3* files containing genic regions) corresponding to the organism.

For *T.themophila* these files are: `T_thermophila_June2014.assembly.fasta` and `T_thermophila_June2014.gff3`. Both files can be found at http://ciliate.org/index.php/home/downloads [25].

For *O.trifallax* these files are: `Oxytricha_trifallax_022112_assembly.fasta` and `Oxytricha_trifallax_022112.gff3`. Both files can be found at http://oxy.ciliate.org/index.php/home/downloads [25].

Core pipeline tools

Our core scripts do not require any additional packages other than the ones mentioned above; however, the comparison tools, depending on what format the data to compare with is given, might use some additonal R packages, such as a spreadsheet reader package. For instance, we have included one named *.xlsx* which allows to read proprietary formats. The results of the genic and intergenic regions are generated in two *.csv* files. These are standard text ASCII files, which can be read with any typical spreadsheet software or R.

[2]Both repositories are synchronized, so that the latest version of RACS is available and can be obtained from both: https://gitrepos.scinet.utoronto.ca/public/?a=summary&p=RACS or https://bitbucket.org/mjponce/RACS

Determination of the genic regions

To count the amount of reads in each genic region the core pipeline script was implemented using Linux shell commands combined with the usage of `BWA` and `SAMtools`. The input files are the genome of reference (`T_thermophila_June2014.assembly.fasta`), the gene annotation file (`T_thermophila_June2014.gff3`) and the INPUT and IP files obtained from NGS. The INPUT files contain the information obtained from NGS prior to the immunoprecipitation; thus, this file contains the initial reference amount of DNA reads. The IP file contains the data after the immunoprecipitation; thus, this file contains the DNA that were enriched by the POI. After the INPUT and IP sequences are aligned with the genome and sorted, the script uses a loop to count the reads in each genic region and deposits the obtained data in a file named "`FINAL.table.`*INPUTfile-IPfile*"; where *INPUTfile* and *IPfile* are the INPUT and IP files respectively. Figure 1 depicts a flowchart representing the required steps to obtain the final table containing the number of reads found in each of the genic regions. Details of the processing stages are shown in Fig. 2, in relation to *T.thermophila* scaffold database and the breakdown of each these steps.

The RACS pipeline was implemented to specifically target data from the *T.thermophila* organism in particular utilizing an specific *gff3* file. However due to the modular fashion in which RACS was implemented, it is possible for users targeting different organisms and even different markers, to "instruct" RACS to do so. At the level of the IGR, if the reference file follows the usual `gff3` structure, nothing has to be modified in the pipeline. As a matter of fact, we implemented several checks in order to verify and guarantee the consistency of the data provided by this file. At the level of the ORF, the user will need to specify a few parameters that will be used when the targets depart from the ones used by default in the pipeline. The terms and filters allow the user to target either genes or mRNA or any other specifier within the reference file, making essentially agnostic of the organism type. In order to achieve this, the user should provide a 'definition' file, indicating the targets for the pipeline for which to filter for the reference file. We have included a subdirectory in the repository "core/defns", where we include some files exemplifying the implementation of different cases and organisms.

In particular, the variables `filter1`, `filter2`, as well as, `delim1`, `delim2`, `delim3`; should be adjusted correspondingly to the organism of interest and the way the data is organized within the reference file. The following code shows an example of how this is done for *T.thermophila* and *O.trifallax*.

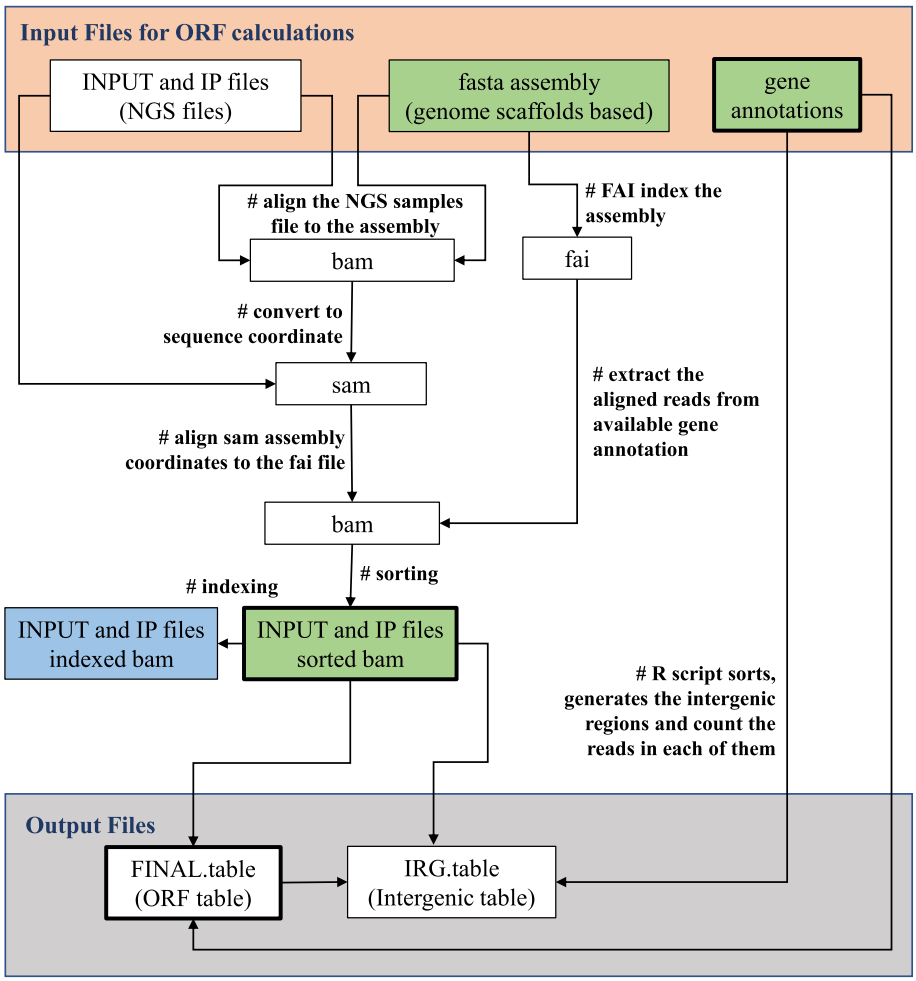

Fig. 1 Core RACS pipeline overview. This flowchart represents the logic steps implemented in the core pipeline. Boxes represent files and file types as indicated in the text. Files with thick boxes represent the Input Files for Intergenic calculations. Files in green are to be uploaded to IGV. File in blue is needed for IGV but it does not have to be uploaded to IGV. This file has to be kept in the same folder directory than the sorted bam

```
filter1=gene
filter2="Name=TTHERM_"
. . .
delim1="TTHERM"
delim2=";Note"
delim3="Note="
```

"defn" file for T.thermophila; see `RACS/core/defns/TT_gene.id`

```
filter1=gene
filter2="Name=Contig_"
. . .
delim1="Name=Contig"
delim2=";Note"
delim3="Note="
```

"defn" file for O.trifallax; see `RACS/core/defns/OXY_gene.id`

Determination of the intergenic regions

The intergenic regions were not available neither determined by the standard packages. For this reason, we developed an R script to determine these regions. In this pipeline these sequences are calculated during each run to account for further genome actualizations. The inputs for this script are the files generated by the genic regions pipeline discussed in the previous section (i.e. "`FINAL.table.`*INPUTfile-IPfile*", the INPUT and IP *.bam* files –which are generated as intermediate files of the Genic Region pipeline–) plus the gene annotation file (e.g. `T_thermophila_June2014.gff3`). First, the script determines the intergenic regions by calculating the beginning and end of each annotated gene within each available scaffold and subtracts these values. The algorithm only reports intergenic regions that are equal

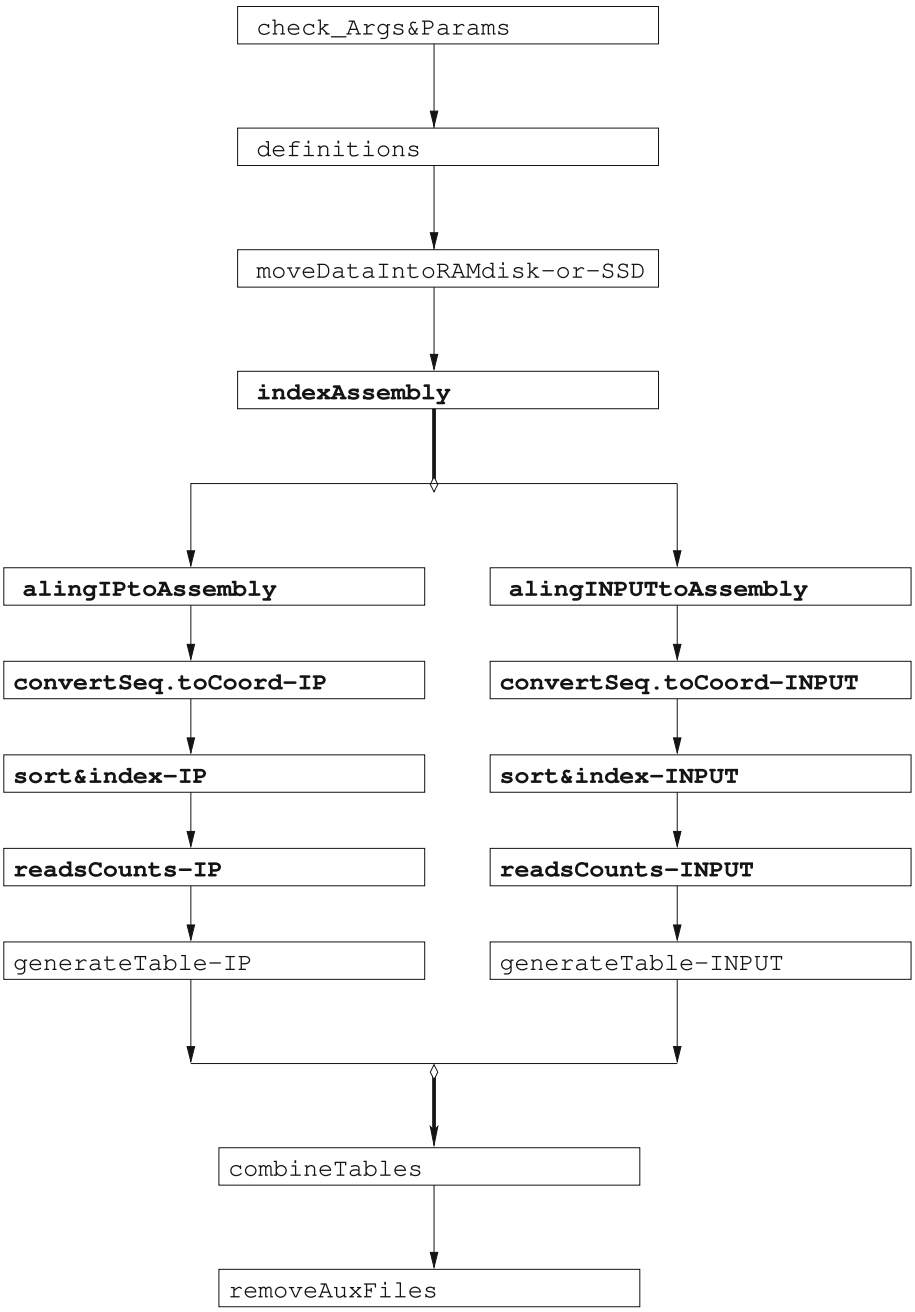

Fig. 2 Schematic diagram of the tasks implemented for the RACS *core pipeline*. Included are details of the processing stages in relation to the scaffold based genome and the breakdown of each these steps. Bold names, indicate bioinformatic specific modules while normal fonts represent generic ones. The bifurcation represents tasks that can be executed in parallel, as there is no data dependency among them

or greater to zero. In the earlier version of the pipeline this constraint was not included, and in some cases, it could result in the pipeline reporting regions with negative sizes. We noticed that 92 of these cases were presented in our previous study [16]; however, we should emphasize that there were not reads present in these regions thus did not affect these results. Second, the script uses the newly generated intergenic regions to count the number of reads in each of them. Finally, the data is deposited in the intergenic table for each of the intergenic regions Fig. 1.

Post-processing

Normalization of reads accumulation and enrichment calculation

To account for differences in the amount of clusters PF (reads) presented among samples, each of the obtained

INPUT and IP values were normalized by dividing them by the corresponding clusters PF value of the Flowcell summary (obtained from the NGS run) or from the Total Sequences (obtained from the fastQC file). These calculations can be done by the script "`normalizedORF.sh`" located in the core directory of RACS. Alternatively, it can also be calculated employing the following two spreadsheets: for the genic regions (*TET_Ibd1_MAC_Genome_Genic.xlsx*), and for the intergenic regions (*TET_Ibd1_MAC_Genome_Intergenic.xlsx*); that can be found in the "datasets" subdirectory within the RACS repository. Notice that there are several spreadsheets provided in this subdirectory, each of them will be used for different organisms/cases and can be used as templates for other datasets.

These spreadsheets contain the reads found in the untagged (or mock purification/negative control) samples in the *Untagged* tab. The user can also add the Flowcell summary details in the *Add_FCS_for_(SAMPLE_ID)* tab. The user can manually introduce the read values for the samples being analyzed in the *Add_(SAMPLE_ID)_ChiP_Seq* tab. In this tab the user can divide the number found by RACS by the corresponding cluster PF number found in the previous tab. This data can be deposited in the "Normalized_INPUT or _IP (FCS)" columns. After the required reads normalization, the accumulation can be obtained as the number of IP reads divided by the number of INPUT reads (IP/INPUT). This can be deposited in the "Enrichment_(N_IP(FCS)/N_INPUT(FCS)" column of the same tabs. The obtained values are filtered (*Filter 1*) by the user by subtracting the corresponding number found in the *Untagged* tab and deposit the values in the *Enrichment_Minus_AVERAGE_untagged* column. If there are more than two samples the values can be averaged and values that are less than 1.5 can be filtered (*Filter 2*) and deposited in the "Enrichment_Average_Sample" tab. For the genic region table, in this tab there is a column containing the Expression profile obtained from the *RNA_Seq* tab. We recommend to copy the filtered cells to the *Results* tab. The distribution of the protein of interest can be calculated in this tab. For the Intergenic table there is a *ORF_vs_IGR* (Intergenic) tab where the number of regions and reads can be calculated. The number of regions is represented by the number of genic and intergenic regions that passed the 2 filters. If there is data available for untagged samples, please refer to the "Utilities: validation and quality checks" section.

The number of reads found in the genic and intergenic regions can be calculated by adding all the available values from the "Normalized IP (FCS)" columns and deposit them in the *ORF_vs_IGR* tab of the Intergenic table.

During the post-processing steps, it is important to note that some regions presented in the processed table may have very few reads after subtracting the values obtained from untagged samples and they may seem as real interactors when they are not. For instance, a sample that has 2 reads in the INPUT and 10 reads in the IP will return an enrichment of 5 and it may pass the filter of 1.5× enrichment but they may not be significantly enriched.

Utilities: validation and quality checks

To account for biological and experimental variability in the wet lab, we typically perform ChIP-Seq using 2 independent samples for each distinct strain and average their Enrichment. To validate the findings, it is important to determine the genic and intergenic regions of the untagged (negative control) INPUT and IP samples. After this determination, we subtracted the obtained average enrichment from untagged to the obtained tagged average of the samples. Then we filtered for values that had an enrichment greater than or equal to 1.5 in the final enrichment column. These are the enriched regions and represent genomic regions to which the POI binds.

Visualization of reads accumulation

The browser IGV [26] can be used to visually inspect and validate the obtained reads based on their ranked enrichment. The files needed are illustrated in Fig. 1 and the 'README' file included in the RACS' repository. MACS2, a main-stream application to call peaks, can also be used as specified in [14]. MACS2 uses the same intermediate files (*.bam*) obtained from the RACS pipeline, hence it can be a good reference to be considered for comparison purposes.

RACS performance

RACS can be run in any normal Linux workstation; however, it can also take advantages of cluster-type environments. In particular, several stages of RACS can be run using multicore architectures with several threads in parallel. In addition to that, RAMdisk can be used to speed up file I/O operations. This is achieved by indicating thorugh a command line argument the specific location for the "working space" that RACS will use to place the input and temporary files to be generated. When we originally developed our pipeline, we tested it in our previous HPC cluster, GPC [1] consisting of 2.53 GHz Intel Xeon E5540, with 16 GB RAM per node (2 GB per core). By comparing the performance of RACS with a typical workstation we noticed a speed-up factor among 8 to 12×. We have also run our pipeline in our newest cluster, Niagara [27], of 1500 Lenovo SD350 servers each with 40 Intel "Skylake" cores at 2.4 GHz. Each node of the cluster has 188 GiB / 202 GB RAM per node, for which we have obtained a speed up of 5 to 10×. In other words, the whole processing of genic (ORF) and intergenic (IGR) for a typical INPUT/IP sample, took between 1 and 2 h. In addition to

that, in our new system is possible to bundle 40 (80 using multithreading) processes together.

Moreover, this first release of RACS utilizes the basic SAMtools and BAM codes, however it has been reported that improvements in processing SAM files could be achieved using SAMBAMBA [28]. One of the many advantages of dealing with an open source, modular pipeline like this, is that it allows interested users to explore this possibility as well, just by modifying the tool to process SAM files and selecting the one desired.

As mentioned above, one additional functionality that RACS offers is the ability to specify the "working space". When using the main script for counting reads in ORF, the user has the ability of indicating whether to use a faster "working space" than traditional spinning disks (ie. HDD) such as memory (ie. RAMdisk) or a solid state device (SSD). In general, utilizing RAMdisk or SSDs, would result in a speed-up of roughly 10 to 30%, depending on hardware specifications and the size of the dataset to be analyzed. The larger the dataset the more I/O operations (reads/writes) that would be needed, hence larger datasets would benefit the most of this. This is of course, assuming that the data and subsequent auxiliary files created during the analysis will fit in "memory". If that is not the case then depending on the system and how it is configured may result in decremental performance (e.g. some computers will begin *swapping* data –i.e. start using traditional HDD space–) or even crash (for instance, is common in many HPC clusters to do not allow for swapping techniques). Differences in performance among SSD vs RAMdisk, are almost negligible, again depending on hardware specs, this can be upmost of the order of few percentages. Finally, it should be noticed that by using RAMdisk (i.e. memory) as a working space, users will reduce the overall computational time, however this is will ultimately depend upon the amount of memory available as this technique will clearly increase the utilization of RAM. As a general estimate, at the moment of running the pipeline, users might estimate the amount of memory needed by one order of magnitude larger (i.e. $\times 10$) than the size of the dataset to be processed. Further details about memory utilization and walltimes as function of number of threads or cores, are presented in Table 2 and in the "doc" directory of the RACS repository.

Results

In this paper we introduce a one-stop methodology to analyze ChIP-Seq data to find the set of genome coordinates for a given POI. This methodology utilizes open-source tools such as BWA [22], SAMtools [23], Linux shell and R scripts [24] and techniques commonly employed in the HPC fields. RACS can be run either in a typical workstation or taking full advantage of HPC resources, such as, multicore architectures and use of RAMdisk, to improve the analysis times making it more efficient, (see details on "RACS performance" section). This pipeline was developed to answer whether the POI localized to a given set of cordinates (genes) or to the remaining regions in the genome that were not given by the user (intergenic). RACS was designed in a user-friendly manner to accommodate researchers with basic knowledge in Linux shell and R [24]. RACS provides accessible downstream analyses of ChIP-Seq data obtained from Illumina instruments. RACS follows a unique approach to tackle this problem, is widely applicable and useful enough to analyze ChIP-Seq related data from a variety of different organisms generated by NGS.

The RACS pipeline, Fig. 2, offers a solution that utilizes an available contig-based genome sequence file and a second annotation file that contains the coordinates for the annotated genes. After processing ChIP-Seq data, RACS will output two tables, the first containing all found reads accumulation in the genic region corresponding to the annotated genes and the second containing the accumulation of reads in the intergenic regions. An intergenic region will be calculated as a region that starts at the end of a given gene coordinate and ends at the beginning of the next contiguous given gene coordinate. RACS will calculate the beginning of a contig as the beginning of an intergenic region (as long as there is not an annotated gene at the beginning of the contig) which ends at the coordinate of the first encountered gene, and it will do the same at the end of each contig. These intergenic or adjacent regions are newly generated each time to account for modifications or improvements in the files containing gene annotations. The obtained results from both tables are normalized to the number of clusters that passed Illumina's "Chastity filter" also called clusters PF. These numbers represent the reads obtained per sample. The normalized values are further filtered by using the data obtained from the mock samples.

Case study

In this section, we describe how RACS was used to analyze and generate the data presented in [16] in addition to its refinement by introducing ChIP-Seq data from untagged strain. The model organism used in [16] is the protist Alveolate *T. thermophila* which is the most experimentally amenable member of this taxonomical group. *T. thermophila* can be used in some cases to understand the basic biology of the parasitic and disease-causing members of the Alveolates. Members of *Plasmodium* species that causes malaria [29–31] and other related species that affect ecosystems [32] and aquiculture [33] can be examined by analogy through our selected model organism. In addition, *T.thermophila* has genes that present homology to human genes [34, 35] and characteristics that makes it an excellent candidate to study chromatin mainly because

of the segregation of transcriptionally active and silent chromatin into two distinct nuclei, macronucleus (MAC) and micronucleus (MIC) respectively [36]. In our recent study we identified a protein, Ibd1, that physically interacts with several chromatin remodeling complexes [16]. The *T.thermophila*'s genome [25] is contig based and contains almost 27 thousand annotated genes or genic regions [37]. To further the understanding of Ibd1, and to contribute to current understanding of how chromatin remodeling works, we analyzed its localization within the genome by ChIP-Seq (Fig. 3) [38–41]. This allowed us to identify the set of genes bound by Ibd1 to begin to understand its function.

Pre-processing of the fastq files and quality assessment

The ChIP samples were processed as described in [16] to make the library preparation using the TruSeq ChIP-Seq kit (Illumina). For the untagged (this study) and Ibd1 [16] strains, libraries were sequenced using the v4 chemistry in a HiSeq2500 instrument (Illumina) set for High Output mode. The obtained read lengths were of 66 base pairs, 6 base pairs corresponded to the adapters

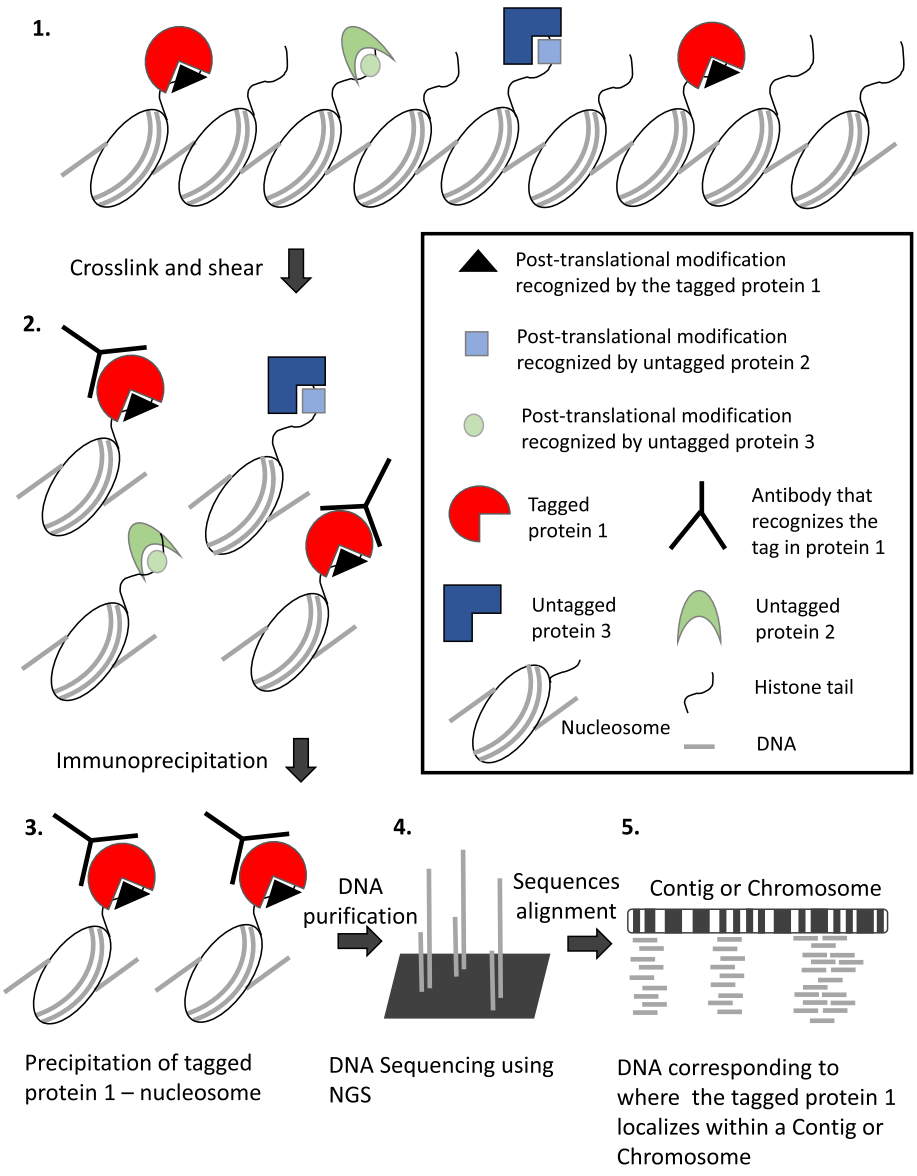

Fig. 3 Diagram summarizing the *ChIP-Seq* technique used to prepare the samples and generate the data from the "wet-lab": 1) Native state of chromatin. 2) Specific antibodies recognize the tagged proteins. 3) Isolation of tagged protein plus its interacting chromatin. 4) After DNA purification and library preparation NGS is performed. 5) The output data from NGS is aligned to *Tetrahymena thermophila*'s genome assembly

for demultiplexing. These files were demultiplexed in *fastq* format and the adapters were trimmed using the bcl2fastq2 Conversion Software v2.20.0. The obtained *fastq* files for the INPUTS and IP samples were assessed by fastQC version 0.11.5 [42]. Each dataset obtained from the ChIP-Seq experiments has a sequence of whole cell DNA (INPUT) and DNA sequenced from an immunoprecipitated (IP) sample. The Ibd1 NGS data generated in [16] can be found at the following Gene Expression Omnibus (GEO) link: GSE103318 [16]. In addition, untagged *T.thermophila* fastq files were generated and they can be found at the following GEO link: GSE125576.

We recommend assessing the quality of the data obtained from the NGS. This step is important to have general information regarding the run. This fastQC report also helps to understand the alerts present in each sample, these alerts do not necessarily mean that the NGS run failed [42]. In other words, this step is to verify if the *fastq* data has any alerts that have to be addressed before proceeding to the processing. For example, in our case our data for the *Per base sequence quality* fell into the very good quality reads (green) area of the y axis allowing us to avoid quality trimming. On the other hand, we obtained a flag for *Overrepresented sequences*, in particular the one that called our attention was the sequence containing only the nucleotide N. Since our reads are 35-58 base pairs long, the allowed maximum mismatch to the genome will be up to 3 base pairs according to the BWA algorithm [22] hence it will not consider these sequences for the alignment.

Visualization and list of reads

After, the determination of enriched regions we can further analyze them using a visualization tool, such as IGV. The region of interest can be copied from either the genic or intergenic table. This localization corresponds to where the protein of interest is localizing with respect to annotated genes (see Fig. 4 panel A) or an intergenic region (see Fig. 4 panel B). It is important to note that for Ibd1's ChIP-Seq [16] data we also used MACS2 a main-stream application to call peaks [14]. The visualization option for MACS2 and RACS are similar in that both provide a specific file that can be used for this purpose. In the case of RACS, our pipeline uses *.bam* and *.fai* files which are generated within the GENIC part of the pipeline (see Fig. 1). Such *.bam* files can be opened in IGV, although the *.fai* (index) file will not, however both files should be present in the same directory. The required files for IGV visualization are depicted in Fig. 1. In addition, the *.bam* files generated by RACS can be used as input for MACS2. When compared the MACS2 visualization file to the RACS *.bam* files using IGV (Fig. 4, panels A and B), we observed that the RACS files provide a visual of the portion enriched. Here we observed that the IP samples are clearly enriched regions showing peaks when compared to the INPUT samples. This can be determined by noting the numbers shown in each of the IGV tracks, which represent its corresponding reads accumulation.

The track corresponding to our pipeline in Fig. 4 panel A shows clear accumulation of reads on the gene that is on the right side as it is in the track form MACS2. However, for the gene and the intergenic region on the left side of the track RACS does not show a clear accumulation whereas MACS2 does. Figure 4 panel B shows a perfect match. The ability of comparing these two tools at the same time can help the user by providing more robust results that can lead or not to further investigation of the specific sequence.

The output lists provided by RACS are segregated into two *.csv* files. The first file contains the genic and the second the intergenic regions. Both lists contain the all reads obtained from the INPUT and IP samples. This obtained data should be filtered with the data obtained from Mock samples. We found that the output of MACS2 provides a list of peaks. MACS2 does not classify the peaks based on the localization to a gene or an intergenic region as RACS does. However, this can be addressed using BEDTools [15] after MACS2 analysis. The datasets generated by MACS2 and RACS can be found at the GEO link GSE103318.

Mock samples facilitate the analysis

In [16] we found that the majority of genes regulated by Ibd1 were highly expressed housekeeping genes. For that analysis a filtering step using ChIP-Seq data from untagged samples were not employed, and instead a cut-off was implemented based on accumulation of reads. However, since the cut-off was arbitrary, there was some degree of uncertainty in regards of its astringency. To overcome this limitation and further facilitate the analysis, for this study we performed a mock ChIP-Seq experiment using untagged control *T.thermophila* cells in order to reveal the identity of the set of specific DNA sequences that have affinity for the antibody-conjugated chromatography resin either directly or mediated through unknown protein(s) in the chromatin extract. RACS analysis of the Ibd1 ChIP-Seq dataset filtered by two mock IP ChIP-Seq replicas from untagged *T.thermophila* enhanced RACS' ability to discriminate non-specific DNA binding (see "Post-processing" section Post-processing for further details). In addition, the use of mock ChIP-Seq samples eliminated the uncertainty associated with using the arbitrary cut-off. Between both the analysis presented in [16], and this new analysis (RACS), there are not major statistical differences regarding Ibd1 localization to genes that are highly, moderate, or low to no-expressed (Table 1). The statistical analysis presented in Table 1 shows that the hypothesis generated in [16] regarding an Ibd1 function related to transcription of

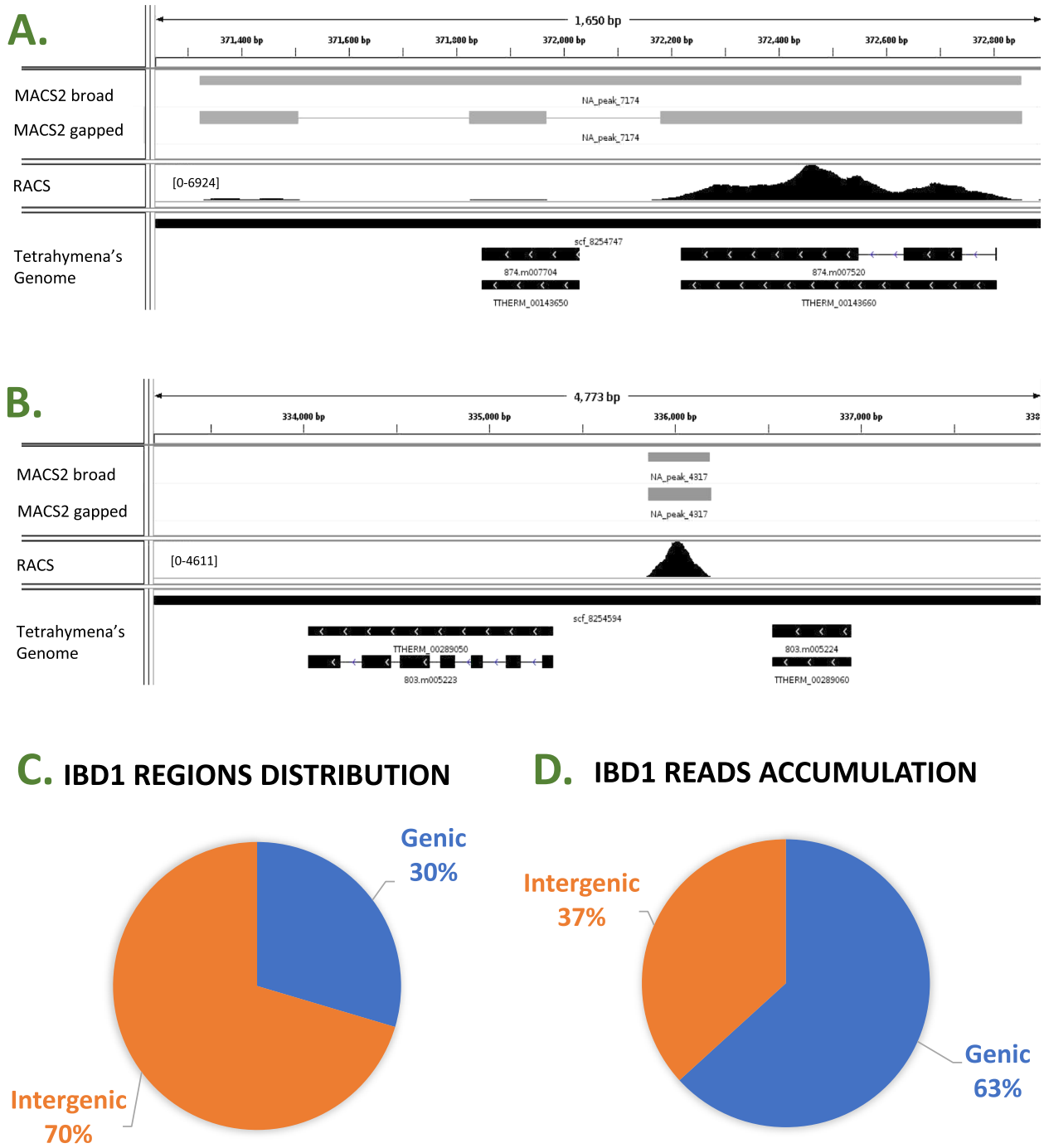

Fig. 4 a and **b** Visualization genic and intergenic region using IGV. The top track shows MACS2 broad and gapped peaks. The middle track shows RACS visual representation of reads accumulation. Note that RACS shows graphical reads behaviour and accumulation preferences. The bottom track shows *T.thermophila*'s genes. On the other hand, MACS2 found two weak peaks that can be interpreted as background by our pipeline. The range inside the brackets represents the highest number of reads for that specific track. **c** Ibd1 localizes to more intergenic than genic regions. **d** The majority of reads are found in genic regions. These results take into consideration the updated information provided by the mock samples

Table 1 Comparison of Ibd1 localization presented in [16] (without untagged controls) and analyzed by RACS using untagged controls (current study)

(a) Ibd1 localization presented in both studies.

Expression level	Ibd1 localization % using untagged controls	Ibd1 localization % without untagged controls
High expression	51	54
Moderate expression	6	16
Low to no-expression	16	14
Non-available expression for the TTHERMs in the GSM692081 data set	27	16

(b) t-Test: Paired Two Sample for Means

t-Test: Paired Two Sample for Means

	Ibd1 localization % using untagged controls	Ibd1 localization % without untagged controls
Mean	25	25
Variance	374	374.6667
Observations	4	4
Pearson Correlation	0.89581514	
Hypothesized mean difference	0	
df	3	
t Stat	0	
$P(T \leq t)$ one-tail	0.5	
t Critical one-tail	2.35336343	
$P(T \leq t)$ two-tail	1	
t Critical two-tail	3.18244631	

There is a correlation of 0.896 and non-statistical differences between the two data sets. The data presented in [16] uses an arbitrary cut-off. The data presented in this paper does not use the arbitrary cut-off and instead uses as cut-off the values obtained by the analyses of the untagged samples

highly expressed genes stands. The calculation of this result can be found in the Result tab of the genic table (RACS/datasets/TET_Ibd1_MAC_Genome_Genic.xlsx).

RACS aids in the determination of the protein of interest function

To gain insights in the POI function, we segregated its localization between genic and intergenic. After analyzing Ibd1's raw ChIP-Seq data with RACS, tables with the total number of reads found in each of the 26,996 genic and 27,780 intergenic regions were generated [16]. From the genic and intergenic tables we observed that Ibd1 localizes to more individual intergenic regions than genic regions (Fig. 4 panel C). However, the majority of reads accumulation are in the genic regions (Fig. 4 panel D), suggesting that Ibd1 primary localization is within the genic implicating Ibd1 function in transcription regulation.

The function of Ibd1 was further inferred based on the GO annotations for biological process [43] categories of genes to which it binds. From the genic table (RACS/datasets/TET_Ibd1_MAC_Genome_Genic.xlsx) we observed that Ibd1 mostly localizes to genes that are highly expressed and related to housekeeping function; such as, cellular function, translation, gene expression, biogenesis, cytoplasmic translation among others (see Fig. 5). The calculation of this result can be found in the *Gene_Ontology* tab of the genic table (RACS/datasets/TET_Ibd1_MAC_Genome_Genic.xlsx).

RACS for *T.thermophila*'s rDNA minichromosome

The obtained data from Ibd1 ChIP-Seq was used against the rDNA minichromosome sequence [44]. Ibd1 is not enriched in any of the 3 genic or 4 intergenic regions. The generated tables can be found in the repository, under the datasets subdirectory: *TET_Ibd1_rDNA_Genic.xlsx* and *TET_Ibd1_rDNA_Intergenic.xlsx*.

Outliers

During the Post-processing stage, we found a great number of reads for the following three genic regions: TTHERM_02141639, TTHERM_02641280, TTHERM_02653301; in the tagged and untagged

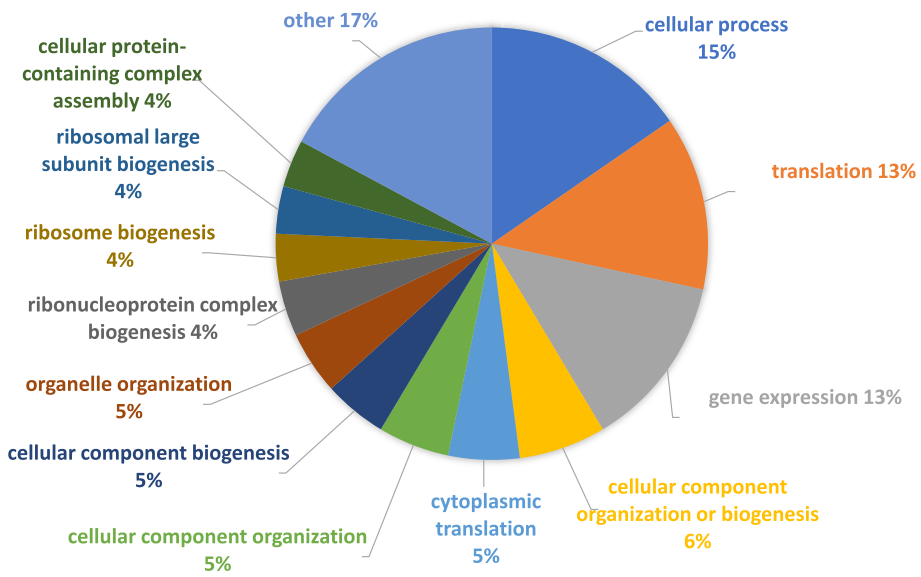

Fig. 5 Gene Ontology (GO) analysis of genes controlled by Ibd1. GO predicted that the majority of Ibd1 bounded genes are related to housekeeping functions

ChIP-Seq samples. After applying the 1.5× cut-off for enrichment of Ibd1 we found that the first two mentioned genic region were filtered out after the subtraction step described in the ("Utilities: validation and quality checks" section). The third gene passed the control, thus, seems that Ibd1 localizes to this region. In other words, the accumulation of the first two regions are due to nonspecific binding and the third to specific binding. This is another example of why the untagged strains can help to determine if this found accumulated DNA in the untagged and tagged samples are or not due to specific binding.

Performance

By implementing this pipeline as described here, we obtained roughly a factor of 4× faster in comparison to a serial and non-I/O optimized (i.e. not using RAMdisk), in an equivalent hardware to the node used in the cluster. This is something we have also observed by using similar techniques (e.g. RAMdisk) in other type of bioinformatics pipelines where the hierarchy of the computational scales is dominated by the I/O parts of the code. Moreover, we processed a second set of data, that was roughly 3 times larger than the original data –which would not fit in memory (> 64 GB)–, utilizing a more modern node (i7 core) with a solid-state device (SSD), we were able to further reduce the processing time approximately by another factor of ∼ 4. This type of trend is typical in cases where performance is dominated by computations and I/O operations (e.g. reading and writing files), for which the combination of faster processing plus faster access to the data is essential for improving the overall performance. Nevertheless, we should emphasize that even when RAMdisk or an SSD can be a solution that could in principle be thrown to similar type of problems, i.e. intensively I/O demanding ones, the best approach would always be to try to mitigate and reduce as much as possible the I/O operations, as these usually represent the slowest part in any computational implementation.

Other points to notice are: i) in most of the cases, increasing the number of cores, improves performance in terms of speed-up factors; ii) speed-up factors, also depend on the size of the data sets, although in general they follow a very similar trend; iii) larger data sets require larger processing times, while –in general– smaller data sets show better scaling performance, which in principle can be understood as the pipeline has no communication parallelism implemented; iv) there are limitations to these scaling trends, for instance when the amount of data/work to be splitten is not big enough with respect to the overhad cost of organizing the work distribution (an example of this can be seen in Table 2 with the MED31-1 dataset when attempting to run with 64 cores).

RACS for *O.trifallax*

To test RACS in a different model organism we utilized the data generated for the following study [17]. The used data set can be found at the GEO link: GSE55703 and the tables generated by RACS are

Table 2 RACS scaling and performance trends for the ORF part of the pipeline: we performed the standard *strong scaling* analysis, as well as a function of different dataset sizes

Initial data size	Number of procesors	Workspace usage	Walltime time
≈3 GB[a]	1	≈27 GB	7037 secs
	2	"	5059 secs
	4	"	3856 secs
	8	"	3238 secs
	16	"	2940 secs
	32	"	2801 secs
	64[b]	"	2463 secs
≈2.4 GB[c]	1	≈20 GB	5477 secs
	2	"	4005 secs
	4	"	3128 secs
	8	"	2678 secs
	16	"	2456 secs
	32	"	2344 secs
	64	"	2161 secs
≈6.8 GB[d]	1	≈50.3 GB	6987 secs
	2	"	5662 secs
	4	"	4864 secs
	8	"	4451 secs
	16	"	4245 secs
	32	"	4148 secs
	64	"	4155 secs
≈7.1 GB[e]	1	≈53.4 GB	7728 secs
	2	"	6191 secs
	4	"	5255 secs
	8	"	4740 secs
	16	"	4529 secs
	32	"	4413 secs
	64	"	4249 secs
≈1.4 GB[f]	1	≈8.3 GB	2874 secs
	2	"	1796 secs
	4	"	1218 secs
	8	"	920 secs
	16	"	773 secs
	32	"	702 secs
	64	"	639 secs

[a] Ibd1-1 data set for *T.thermophila* [16].
[b] Although there are 40 physical cores in the TDS/Niagara nodes, *hyperthreading* is enabled so it can be used up to 80 logical cores.
[c] Ibd1-2 data set for *T.thermophila* [16].
[d] MED31-1 data set for *T.thermophila* [48].
[e] MED31-2 data set for *T.thermophila* [48].
[f] Data set for *O.trifallax*.
As it can be seen, the working space (in this case *memory utilization*) can reach up to a factor of 9-10× the size of the initial data to be processed. Further details about memory consumption can be found in the README document and the "doc" directory, included within the RACS repository. These tests were run in the TDS system (i.e. one Lenovo SD530 node with 40 cores and 192GB of RAM with CentOS 7.4 operating system) of the Niagara supercomputer [27], utilizing RAMDISK as working space

available in the repository, under the datasets subdirectory: *OXY_Rpb1_MAC_Genic.xlsx* and *OXY_Rpb1_MAC_Intergenic.xlsx*. After analyzing this published data, we found that Rpb1 binds to 90% of the 24,885 annotated genes and to 54% of the 43,326 RACS generated intergenic regions. This result concludes that Rpb1 has a genome wide distribution and it is consistent to what it was published previously. In Fig. 6 (panels A and B) it is shown that the DNA distribution throughout the gene is consistent to what it was found in [17]. A new result found by RACS for this study is presented in Fig. 6 panels C and D. Figure 6 panel C shows that Rpb1 interacts with the same amount of genic and intergenic regions. However, Fig. 6 panel D shows that 81% of all reads are distributed in the genic regions. This shows that Rpb1 has a preference for gene bodies.

Discussion

In this paper, we have presented RACS, a pipeline implementation utilizing open source tools for the rapid analysis of ChIP-Seq data for a POI from an organism with a contig-based genome sequence. RACS utilizes the predicted gene coordinates and groups the reads accumulation for genic and intergenic regions in ranked form. The objective is to be able to infer POI function based on its chromatin occupancy. This pipeline has been applied to *Tetrahymena thermophila* and *Oxytricha trifallax*'s ChIP-Seq data, but its application can be extended to ChIP-Seq datasets generated in any other organisms.

Initially, when we called peaks for Ibd1 ChIP-Seq data using MACS2, the output did not indicate whether the protein localizes to a genic or an intergenic region. MACS2 calls all peaks regardless of their position in a genic or intergenic region, which makes interpretation difficult when combined with the minimal annotation of the *Tetrahymena* genome. RACS segregates ChIP-Seq ranked peaks between genic and intergenic which can help to quickly assign biological function to a POI. We note that other programs, such as BEDTools, can be used to perform this task in combination with MACS2. Without the need of any additional "external" software, RACS calls peaks and segregates them in two tables based on the given set of coordinates (genes) or the remaining regions in the genome that were not provided by the user (intergenic) . Thus, our pipeline is appropriate to address biological questions regarding function based on genome position.

We hold the opinion that MACS2 and RACS are complementary to each other, but empathize that they are not dependent on each other for analysis. For example, MACS2 can be used to establish or to generate a set of coordinates for a specific transcription protein binding to the genome. Thus, we can infer that the POI is attaching to specific areas in the genome to control transcription

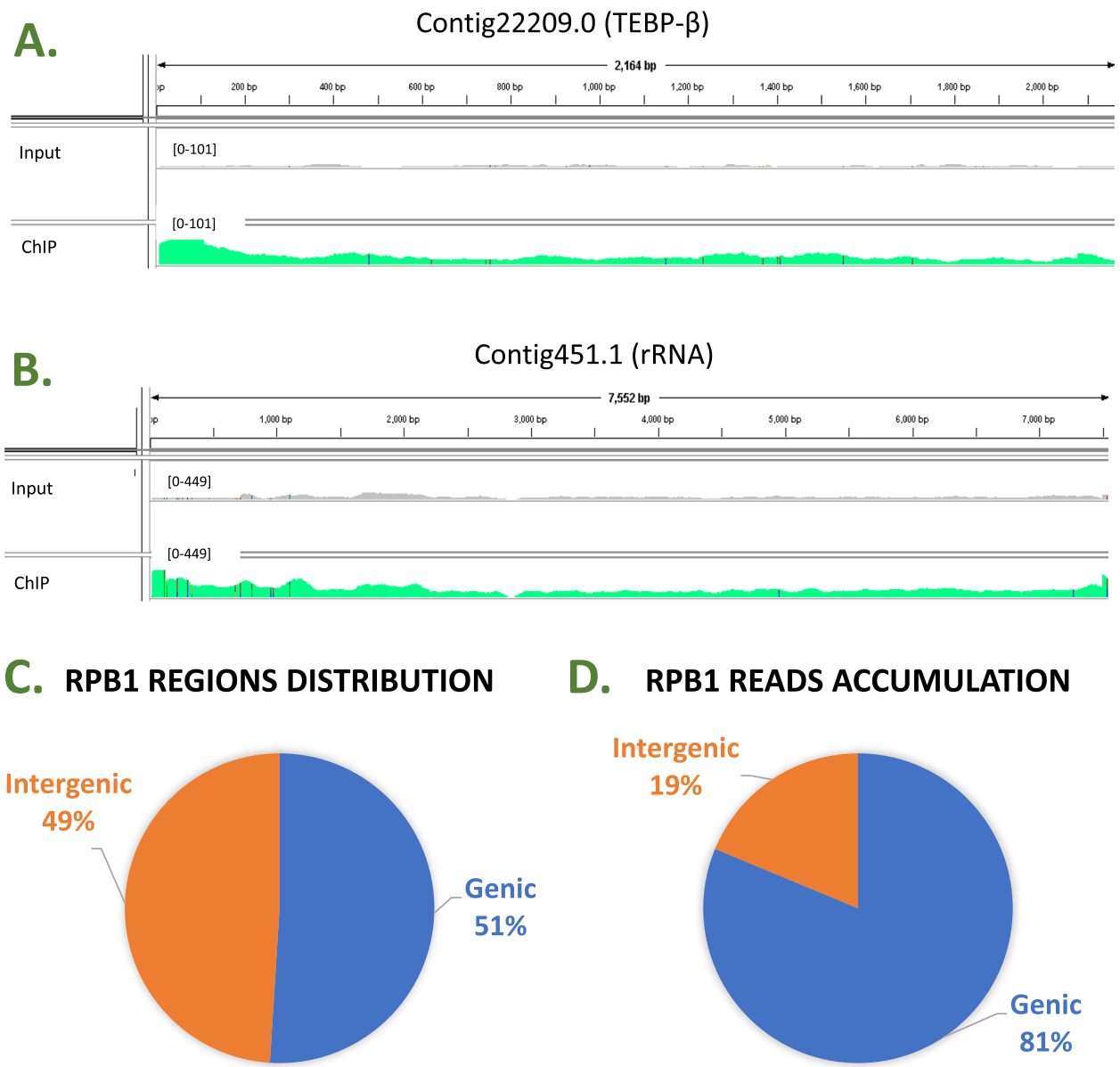

Fig. 6 RACS analysis using *Oxytricha trifallax* ChIP-Seq Rpb1 data gives results comparable to the reported in [17]. **a** and **b** Rpb1 enriches along Contig22209.0 and Contig451.1. The range inside the brackets represents the highest number of reads for that specific track. **c** Rpb1 can be found binding a similar amount of genic and intergenic regions throughout the genome; however, **d** most of the reads that were pulled down by Rpb1 are within genic regions

and we could annotate these regions as binding sites for the specific transcription protein. Then, if we perform ChIP-Seq on a different protein that has also been shown to physically interact with the transcription protein previously mentioned, we could use the coordinates given by MACS2 to generate a *.gff3* file to input it alongside the genome file to the RACS pipeline. This will allow us to rapidly determine the degree of overlap and potential co-localization in some or all binding sites. In that, MACS and RACS can synergize to provide a powerful tool for the analysis of less developed genome sequences.

Even when there are many computational tools available for processing ChIP-Seq data, RACS is particularly suitable for the analysis of contig-based genome sequence with associated minimal annotation. Other tools, such as, MACS2 and metagene using deepTools analysis [45] complement RACS. Recently several ChIP-Seq studies [46–48] have emerged for *T.thermophila*. However, there is a lack of standardized computational methods for this model organism, hence it becomes difficult to reliably reach at the same conclusions when replicating the findings. Our tool is the first effort in *T.thermophila* to

provide a community resource for genome-wide ChIP-Seq studies, therefore it has the potential to contribute to standardization of ChIP-Seq analyses in ciliates. We intend to continue refinement of RACS based on community need. For example, recently, single-molecule sequencing based on nanopores has emerged as a promising technology with a potential to revolutionize the genomics [49]. The nanopore sequencing provides the advantages of 1) long reads, enabling the de novo transcriptome analysis [50], 2) point-of-care, making real-time analysis possible [51], and 3) PCR free, allowing the direct identification of epigenetics [52]. Considering its promising outcomes, studies using model organisms with divergent genomes, such ciliates and parasitic organisms including *Trypanosoma*, will be particularly benefited from the nanopore sequencing technology [53]. Currently, a major challenge is to develop sophisticated and high-performance computational tools to interpret and analyze the nanopore sequencing data [54–56]. In future, we aim to improve and implement the RACS pipeline for the analysis of nanopore sequencing data.

Conclusions

RACS is an excellent tool for genomes that are contig-based and/or have poor annotations, it permits the segregation of reads accumulation between genic and intergenic region after ChIP-Seq processing. RACS is complementary to other tools, such as MACS2, as it can help to discriminate complex regions improving the overall analysis.

RACS offers an alternative tool with a different approach focused on a simple, modular and open approach. RACS offers a versatile, agile and modular pipeline that cover many of the steps needed in the process of analyzing ChIP-Seq data.

The pipeline uses HPC tools, such as RAMdisk or batch processing via scheduling in cluster type environments, so that the data analysis can be done for large datasets. The scripts are reusable and generic enough that can be simply modified and utilized in other pipelines as well.

The modular approach we followed when developing RACS, also allows for future developments as this pipeline could be easily ported as a backend of a web interface, or a *gateway* portal, serving a larger group of researchers from different disciplines.

Abbreviations

ChIP-Seq: Chromatin immunoprecipitation coupled to next generation sequencing; ENCODE: Encyclopedia of DNA elements; FCS: Flowcell summary; GEO: Gene expression omnibus; GO: Gene oncology; GPC: General purpose cluster; HDD: Hard disk device; HPC: High performance computing; I/O: Input/output; IGR: Intergenic region; IP: Immunoprecipitated; MAC: Macronucleus; MIC: Micronucleus; N: Normalized; NGS: Next generation sequencing; ORF: Open region frame; OXY: Oxytricha trifallax; PCR: Polymerase chain reaction; PF: Passing filtering; RACS: Rapid analysis of ChIP-Seq data; TET: *Tetrahymena thermophila*

Acknowledgements

We acknowledge Tanja Durbic and Graham O'Hanlon from the Donnelly Sequencing Centre (DSC - http://ccbr.utoronto.ca/donnelly-sequencing-centre) at the University of Toronto for sequencing our ChIP samples.

Authors' contributions

AS defined the alignment steps using BWA and SAMtools and the reads extraction methodology and MP automated these steps by developing the scripts. AS conceived the reads extraction methodology for the intergenic regions and MP developed and automated it. AS and MP analysed the data, designed the study and wrote the manuscript. AS and SNS prepared the ChIP samples from untagged strains. JF conceived the study, coordinated and edited the manuscript. All authors read and approved the final manuscript.

Funding

Work in the Fillingham laboratory was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants RGPIN-2015-06448. SciNet is funded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund—Research Excellence; and the University of Toronto. Design of the study, collection, analysis and interpretation of data was carried out by the authors of this paper.

Availability of data and materials

ChIP-Seq data used to develop this methodolgy can be found online at Gene Expression Omnibus (GEO) http://www.ncbi.nlm.nih.gov/geo/ GSE103318 and GSE125576. NGS and peak files produced in this study were deposited at https://www.ncbi.nlm.nih.gov/geo/ with unique identifiers GSE103318 and GSE125576. Direct links: http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE103318 and http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE125576. RACS pipeline (including a README file with instructions in how to use the scripts) can be accessed through any of the following repositories: https://gitrepos.scinet.utoronto.ca/public/?a=summary&p=RACS https://bitbucket.org/mjponce/RACS.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Chemistry and Biology, Ryerson University, 350 Victoria St, M5B 2K3 Toronto, Canada. [2]SciNet High Performance Computing Consortium, University of Toronto, 661 University Ave, M5G 1M1 Toronto, Canada. [3]Department of Molecular Genetics, University of Toronto, 1 King's College Cir, M5S 1A8 Toronto, Canada.

Received: 22 March 2019 Accepted: 13 September 2019
Published online: 29 October 2019

References

1. Loken C, Gruner D, Groer L, Peltier R, Bunn N, Craig M, Henriques T, Dempsey J, Yu C-H, Chen J, Dursi LJ, Chong J, Northrup S, Pinto J, Knecht N, Zon RV. SciNet: Lessons Learned from Building a Power-efficient Top-20 System and Data Centre. J Phys Conf Ser. 2010;256(1):012026.
2. Ponce M, Spence E, Gruner D, van Zon R. Scientific computing, high-performance computing and data science in higher education. J Comput Sci Educ. 2019. https://doi.org/10.22369/issn.2153-4136/10/1/5.1604.05676.
3. Venter ea, Craig J. The sequence of the human genome. Science. 2001;291(5507):1304–51. https://doi.org/10.1126/science.1058040.

4. The NIH HMP Working Group, Peterson J, et al. The nih human microbiome project. Genome Res. 2009;19(12):. https://doi.org/10.1101/gr.096651.109.
5. Wang Z, Gerstein M, Snyder M. Rna-seq: a revolutionary tool for transcriptomics. Genetics. 2009;10(1):. https://doi.org/10.1038/nrg2484.
6. Ng SB, Turner EH, Robertson PD, Flygare SD, Bigham AW, et al. Targeted capture and massively parallel sequencing of 12 human exomes. Nature. 2009;461(7261):.
7. Schuster SC. Next-generation sequencing transforms today's biology. Nat Methods. 2008;5:. https://doi.org/10.1038/nmeth1156.
8. Loman N, Misra R, Dallman T, Constantinidou C, Gharbia S, Wain J, Pallen M. Corrigendum: Performance comparison of benchtop high-throughput sequencing platforms. Nat Biotechnol. 2012;30:. https://doi.org/10.1038/nbt0612-562f.
9. Johnson DS, Mortazavi A, Myers RM, Wold B. Genome-wide mapping of in vivo protein-dna interactions. Science. 2007;316(5830):1497–502. https://doi.org/10.1126/science.1141319. http://arxiv.org/abs/http://science.sciencemag.org/content/316/5830/1497.full.pdf.
10. Park PJ. Chip–seq: advantages and challenges of a maturing technology. Nat Rev Genet. 2009;10:. https://doi.org/10.1038/nrg2641.
11. Mardis ER. The impact of next-generation sequencing technology on genetics. Trends Genet. 2008;24:. https://doi.org/0.1016/j.tig.2007.12.007.
12. Schmidt D, Wilson MD, Ballester B, Schwalie PC, Brown GD, Marshall A, Kutter C, Watt S, Martinez-Jimenez CP, Mackay S, Talianidis I, Flicek P, Odom DT. Five-vertebrate chip-seq reveals the evolutionary dynamics of transcription factor binding. Science. 2010;328(5981):1036–40. https://doi.org/10.1126/science.1186176.
13. Cremer T, Cremer C. Chromosome territories, nuclear architecture and gene regulation in mammalian cells. Nat Rev Genet. 2001;2(4):292–301. Copyright - Copyright Nature Publishing Group Apr 2001; Last updated - 2013-01-27.
14. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein B. E., Nusbaum C, Myers RM, Brown M, Li W, Liu XS. Model-based analysis of chip-seq (macs). Genome Biol. 2008;9(9):137. https://doi.org/10.1186/gb-2008-9-9-r137.
15. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26(6):841–2. https://doi.org/10.1093/bioinformatics/btq033.
16. Saettone A, Garg J, Lambert J-P, Nabeel-Shah S, Ponce M, Burtch A, Thuppu Mudalige C, Gingras A-C, Pearlman RE, Fillingham J. The bromodomain-containing protein ibd1 links multiple chromatin-related protein complexes to highly expressed genes in tetrahymena thermophila. Epigenet Chromatin. 2018;11(1):10. https://doi.org/10.1186/s13072-018-0180-6.
17. Khurana JS, Wang X, Chen X, Perlman D, Landweber LF. Transcription-independent functions of an rna polymerase ii subunit, rpb2, during genome rearrangement in the ciliate, oxytricha trifallax. Genetics. 2014;197(3):839–849. https://doi.org/10.1534/genetics.114.163279.
18. Feng H, Misra V, Rubenstein D. Pbs: A unified priority-based scheduler. Sigmetrics Perform Eval Rev. 2007;35(1):203–14. https://doi.org/10.1145/1269899.1254906.
19. Steine M, Bekooij M, Wiggers M. A priority-based budget scheduler with conservative dataflow model. In: Proceedings of the 2009 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools. DSD '09. Washington: IEEE Computer Society; 2009. p. 37–44. https://doi.org/10.1109/DSD.2009.148.
20. McLay R, Schulz KW, Barth WL, Minyard T. Best practices for the deployment and management of production hpc clusters. In: State of the Practice Reports. SC '11. New York: ACM; 2011. p. 9–1911. https://doi.org/10.1145/2063348.2063360.
21. Yoo AB, Jette MA, Grondona M. Slurm: Simple linux utility for resource management. In: Feitelson D, Rudolph L, Schwiegelshohn U, editors. Job Scheduling Strategies for Parallel Processing. Berlin: Springer; 2003. p. 44–60.
22. Li H, Durbin R. Fast and accurate short read alignment with Burrows–Wheeler transform. Bioinformatics. 2009;25(14):1754. https://doi.org/10.1093/bioinformatics/btp324.
23. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25(16):2078. https://doi.org/10.1093/bioinformatics/btp352.
24. R Core Team. R: A Language and Environment for Statistical Computing. Vienna: R Foundation for Statistical Computing; 2016. R Foundation for Statistical Computing. https://www.R-project.org/.
25. Stover NA, Krieger CJ, Binkley G, Dong Q, Fisk DG, Nash R, Sethuraman A, Weng S, Cherry JM. Tetrahymena genome database (tgd): a new genomic resource for tetrahymena thermophila research. Nucleic Acids Res. 2006;34(suppl_1):500–3. https://doi.org/10.1093/nar/gkj054.
26. Robinson JT, Thorvaldsdóttir H, Winckler W, Guttman M, Lander ES, Getz G, Mesirov JP. Integrative Genomics Viewer. Nat Biotechnol. 2011;29(1):24–26.
27. Ponce M, van Zon R, Northrup S, Gruner D, Chen J, Ertinaz F, Fedoseev A, Groer L, Mao F, Mundim BC, Nolta M, Pinto J, Saldarriaga M, Slavnic V, Spence E, Yu C-H, Peltier WR. Deploying a top-100 supercomputer for large parallel workloads: The niagara supercomputer. In: Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning). PEARC '19. New York: ACM; 2019. p. 34–1348. https://doi.org/10.1145/3332186.3332195.
28. Tarasov A, Vilella AJ, Cuppen E, Nijman IJ, Prins P. Sambamba: fast processing of ngs alignment formats. Bioinformatics. 2015;31(12):2032–4. https://doi.org/10.1093/bioinformatics/btv098.
29. Peterson DS, Gao Y, Asokan K, Gaertig J. The circumsporozoite protein of plasmodium falciparum is expressed and localized to the cell surface in the free-living ciliate tetrahymena thermophila. Mol Biochem Parasitol. 2002;122(2):119–26. https://doi.org/10.1016/S0166-6851(02)00079-8.
30. Linder JU, Engel P, Reimer A, Krüger T, Plattner H, Schultz A, Schultz JE. Guanylyl cyclases with the topology of mammalian adenylyl cyclases and an n-terminal p-type atpase-like domain in paramecium, tetrahymena and plasmodium. EMBO J. 1999;18(15):4222–32. https://doi.org/10.1093/emboj/18.15.4222. http://arxiv.org/abs/http://emboj.embopress.org/content/18/15/4222.full.pdf.
31. Roelofs J, Smith JL, Haastert PJMV. cgmp signalling: different ways to create a pathway. Trends Genet. 2003;19(3):132–4. https://doi.org/10.1016/S0168-9525(02)00044-6.
32. Tang YZ, Egerton TA, Kong L, Marshall HG. Morphological variation and phylogenetic analysis of the dinoflagellate gymnodinium aureolum from a tributary of chesapeake bay. J Eukaryot Microbiol. 2008;55(2):91–99. https://doi.org/10.1111/j.1550-7408.2008.00305.x.
33. Buchmann K, Sigh J, Nielsen CV, Dalgaard M. Host responses against the fish parasitizing ciliate ichthyophthirius multifiliis. Vet Parasitol. 2001;100(1):105–16. https://doi.org/10.1016/S0304-4017(01)00487-3. Vaccination and Immunity against Parasites.
34. Eisen JA, Coyne RS, Wu M, Wu D, Thiagarajan M, Wortman JR, Badger JH, Ren Q, Amedeo P, Jones KM, Tallon LJ, Delcher AL, Salzberg SL, Silva JC, Haas BJ, Majoros WH, Farzad M, Carlton JM, Smith Jr. RK, Garg J, Pearlman RE, Karrer KM, Sun L, Manning G, Elde NC, Turkewitz AP, Asai D. J, Wilkes DE, Wang Y, Cai H, Collins K, Stewart BA, Lee S. R, Wilamowska K, Weinberg Z, Ruzzo WL, Wloga D, Gaertig J, Frankel J, Tsao C.-C., Gorovsky MA, Keeling PJ, Waller RF, Patron N. J, Cherry JM, Stover NA, Krieger CJ, del Toro C, Ryder H. F, Williamson SC, Barbeau RA, Hamilton EP, Orias E. Macronuclear genome sequence of the ciliate tetrahymena thermophila, a model eukaryote. PLoS Biol. 2006;4(9):1–23. https://doi.org/10.1371/journal.pbio.0040286.
35. Ashraf K, Nabeel-Shah S, Garg J, Saettone A, Derynck J, Gingras A-C, Lambert J-P, Pearlman RE, Fillingham J. Proteomic analysis of histones H2A/H2B and variant Hv1 in Tetrahymena thermophila reveals an ancient network of chaperones. Mol Biol Evol. 2019;msz039:. https://doi.org/10.1093/molbev/msz039. http://arxiv.org/abs/http://oup.prod.sis.lan/mbe/advance-article-pdf/doi/10.1093/molbev/msz039/27974900/msz039.pdf.
36. Martindale DW, Allis CD, Bruns PJ. Conjugation in tetrahymena thermophila: A temporal analysis of cytological stages. Exp Cell Res. 1982;140(1):227–36. https://doi.org/10.1016/0014-4827(82)90172-0.
37. Xiong J, Lu X, Zhou Z, Chang Y, Yuan D, Tian M, Zhou Z, Wang L, Fu C, Orias E, Miao W. Transcriptome analysis of the model protozoan, tetrahymena thermophila, using deep rna sequencing. PLoS ONE. 2012;7(2):1–13. https://doi.org/10.1371/journal.pone.0030630.
38. Bailey T, Krajewski P, Ladunga I, Lefebvre C, Li Q, Liu T, Madrigal P, Taslim C, Zhang J. Practical Guidelines for the Comprehensive Analysis of ChIP-seq Data. PLoS Comput Biol. 2013;9(11):1–8. https://doi.org/10.1371/journal.pcbi.1003326.
39. Landt SG, Marinov GK, Kundaje A, Kheradpour P, Pauli F, Batzoglou S, Bernstein BE, Bickel P, Brown JB, Cayting P, Chen Y, DeSalvo G, Epstein C, Fisher-Aylor KI, Euskirchen G, Gerstein M, Gertz J, Hartemink

AJ, Hoffman MM, Iyer VR, Jung YL, Karmakar S, Kellis M, Kharchenko PV, Li Q, Liu T, Liu XS, Ma L, Milosavljevic A, Myers RM, Park PJ, Pazin MJ, Perry MD, Raha D, Reddy TE, Rozowsky J, Shoresh N, Sidow A, Slattery M, Stamatoyannopoulos JA, Tolstorukov MY, White KP, Xi S, Farnham PJ, Lieb JD, Wold BJ, Snyder M. ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. Genome Res. 2012;22(9):1813–31. https://doi.org/10.1101/gr.136184.111.

40. Cormier N, Kolisnik T, Bieda M. Reusable, extensible, and modifiable R scripts and Kepler workflows for comprehensive single set ChIP-seq analysis. BMC Bioinformatics. 2016;17(1):270. https://doi.org/10.1186/s12859-016-1125-3.
41. Qin Q, Mei S, Wu Q, Sun H, Li L, Taing L, Chen S, Li F, Liu T, Zang C, Xu H, Chen Y, Meyer CA, Zhang Y, Brown M, Long HW, Liu XS. ChiLin: a comprehensive ChIP-seq and DNase-seq quality control and analysis pipeline. BMC Bioinformatics. 2016;17(1):404. https://doi.org/10.1186/s12859-016-1274-4.
42. Andrews S, et al. FastQC: a quality control tool for high throughput sequence data. 2010.
43. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, et al. Gene ontology: tool for the unification of biology. Nat Genet. 2000;25(1):25.
44. Reischmann KP, Zhang Z, Kapler GM. Long range cooperative interactions regulate the initiation of replication in the Tetrahymena thermophlla rDNA minichromosome. Nucleic Acids Res. 1999;27(15):3079–89. https://doi.org/10.1093/nar/27.15.3079.
45. Ramírez F, Dündar F, Diehl S, Grüning BA, Manke T. deepTools: a flexible platform for exploring deep-sequencing data. Nucleic Acids Res. 2014;42(W1):187–91. https://doi.org/10.1093/nar/gku365. http://arxiv.org/abs/.
46. Wang Y, Chen X, Sheng Y, Liu Y, Gao S. N6-adenine dna methylation is associated with the linker dna of h2a. z-containing well-positioned nucleosomes in pol ii-transcribed genes in tetrahymena. Nucleic Acids Res. 2017;45(20):11594–606.
47. Kataoka K, Mochizuki K. Phosphorylation of an hp1-like protein regulates heterochromatin body assembly for dna elimination. Dev Cell. 2015;35(6):775–88.
48. Garg J, Saettone A, Nabeel-Shah S, Cadorin M, Ponce M, Marquez S, Pu S, Greenblatt J, Lambert J-P, Pearlman RE, Fillingham J. The med31 conserved component of the divergent mediator complex in tetrahymena thermophila participates in developmental regulation. Curr Biol. 2019;29(14):2371–96. https://doi.org/10.1016/j.cub.2019.06.052.
49. Loose MW. The potential impact of nanopore sequencing on human genetics. Human Mol Genet. 2017;26(R2):202–7. https://doi.org/10.1093/hmg/ddx287. http://arxiv.org/abs/http://oup.prod.sis.lan/hmg/article-pdf/26/R2/R202/20425119/ddx287.pdf.
50. Byrne A, Beaudin AE, Olsen HE, Jain M, Cole C, Palmer T, DuBois RM, Forsberg EC, Akeson M, Vollmers C. Nanopore long-read rnaseq reveals widespread transcriptional variation among the surface receptors of individual b cells. Nat Commun. 2017;8:16027.
51. Lu H, Giordano F, Ning Z. Oxford nanopore minion sequencing and genome assembly. Genomics Proteomics Bioinforma. 2016;14(5):265–279. https://doi.org/10.1016/j.gpb.2016.05.004. SI: Big Data and Precision Medicine.
52. Simpson JT, Workman RE, Zuzarte P, David M, Dursi L, Timp W. Detecting dna cytosine methylation using nanopore sequencing. Nat Methods. 2017;14(4):407.
53. Díaz-Viraqué F, Pita S, Greif G, de Souza RdCM, Iraola G, Robello C. Nanopore Sequencing Significantly Improves Genome Assembly of the Protozoan Parasite Trypanosoma cruzi. Genome Biol Evol. 2019;11(7):1952–7. https://doi.org/10.1093/gbe/evz129. http://arxiv.org/abs/.
54. Han R, Li Y, Gao X, Wang S. An accurate and rapid continuous wavelet dynamic time warping algorithm for end-to-end mapping in ultra-long nanopore sequencing. Bioinformatics. 2018;34(17):722–31. https://doi.org/10.1093/bioinformatics/bty555. http://oup.prod.sis.lan/bioinformatics/article-pdf/34/17/i722/25702439/bty555.pdf.
55. Li Y, Han R, Bi C, Li M, Wang S, Gao X. DeepSimulator: a deep simulator for Nanopore sequencing. Bioinformatics. 2018;34(17):2899–908. https://doi.org/10.1093/bioinformatics/bty223.
56. Teng H, Cao MD, Hall MB, Duarte T, Wang S, Coin LJM. Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning. GigaScience. 2018;7(5). https://doi.org/10.1093/gigascience/giy037.

Publisher's Note