GRAVI User Guide (v0.1.6)

Stephen Pederson

2023-07-09

Contents

1 Introduction 5

2 Workflow Description 7
2.1 Annotation Setup 7
2.2 Peak Calling 8
2.3 Differential Signal 10
2.4 Pairwise Comparisons 15

3 Quick Start Guide 19
3.1 Install Snakemake 19
3.2 Create the Directory Structure 19
3.3 Run the Pipeline 19
3.4 Tips And Tricks 20

4 Files and Directories 21
4.1 Directory Structure 21
4.2 Alignments 22
4.3 Sample Descriptions 22
4.4 Additional Files 23

5 YAML Configuration Files 25
5.1 The Main Configuration: `config.yml` 25
5.2 Colour Schemes: `colours.yml` [#colours-yml] 29
5.3 Additional Parameters: `params.yml` 29
5.4 HTML Settings: `rmarkdown.yml` 31

Chapter 1

Introduction

This book is the primary documentation for running the GRAVI (Gene Regulatory Analysis using Variable Inputs) workflow. This workflow is managed using `snakemake` (>v7.7) for running locally or on any HPC, and is designed to minimally take one ChIP target under at least two conditions. There is no theoretical upper limit to the number of ChIP targets which can be analysed, although the practicalities of interpretation will dictate this.

Required additional inputs are:

- A gtf with gene, transcript and exon-level annotations. The workflow has been tested extensively with Gencode annotations and these are preferred.
- A blacklist should also be provided as a bed file. These can easily be obtained from here

In addition to ≥ 1 ChIP targets, **optional** input includes:

- Differential Expression results from a single RNA-Seq experiment as a `tsv/csv`
- Any type of genomic feature derived or obtained externally (`*gtf`)
- HiC Interactions (`*.bedpe`)
- Additional coverage tracks for visualisation, such as those produced by key histone marks (`*bigwig`)

The GRAVI workflow itself will

- Annotate the genome using a custom, transcript-focussed approach
- Identify peaks using `macs2 callpeak`
- Perform differential binding analysis for each ChIP target & requested comparisons
- Compare differential binding results across ChIP targets or samples (Pairwise comparisons)
- Perform enrichment analyses at all steps of the workflow

The primary output is a series of `html` pages generated from `rmarkdown` files, along with key figures and tables able to be shared amongst collaborators and incorporated directly into publications.

Chapter 2

Workflow Description

The GRAVI workflow performs multiple steps, most of which depend on those conducted previously. The workflow management software `snakemake`[Mölder et al., 2021] is used to run the complete workflow, giving the capacity to be run on local servers or HPC systems.

The `snakemake` DAG (directed acyclic graph) of the workflow is always included in the compiled document, however a simplified version is presented below.

Simplified DAG of the GRAVI workflow. Key steps are numbered and shown in black. Inputs are shown in blue, whilst key outputs are shown in coral.

The workflow produces a series of HTML reports as a larger webpage, inspired by the excellent `workflowr`[Blischak et al., 2019] package but instead relying directly on `rmarkdown::render_site()`[Allaire et al., 2022]. The compiled html pages will be placed in the `docs` directory, with all source `rmarkdown` files being placed in the `analysis` directory. Key additional outputs (e.g. bed/csv files) are placed in the `output` directory. The standardised directory layout is shown in Section 4.1. The complete R environment from every compiled HTML page is also saved in `output/envs`, however these are marked as temporary files by the workflow and will be deleted when removing `temp` output (`snakemake --delete-temp-output --cores 1`)

2.1 Annotation Setup

Genomic Regions

Defining genomic regions is a key part of the setup for analysis. Under the GRAVI workflow, a series of *non-overlapping genomic regions* are defined which characterise the most likely role/aspect for each specific region. As these will be unique to each Gencode build, and also include any optional RNA-Seq data, this step is performed for every experiment. This step follows the default process as defined in the function `defineRegions()` from the Bioconductor package `extraChIPs`

1. *Promoters*: By default these are defined as -1500/+500bp from every distinct TSS. If promoters from two or more transcripts overlap, they are merged into a single promoter. These ranges can be easily changed to increase or decrease the size using the YAML as described in Section 5.3
2. *Upstream Promoters*: These are extended promoter regions up to 5kb upstream by default, which again can be changed to suit
3. *Exons* are defined as any exon not overlapping a promoter or upstream promoter
4. *Introns* are defined as any transcribed sequence not overlapping a promoter, upstream promoter or exon
5. *Intergenic Regions*: are divided into two subsets, neither of which are permitted to overlap any previously defined regions
   - Within 10kb of a gene. Again this distance is customisable
   - Beyond 10kb of a gene.

During the characterisation of all the above regions, mappings to associated genes and transcripts is retained and included in the subsequent `GenomicRanges` object.

Other Steps

Additional steps performed during preparation of the annotations are to ensure that all external features, treatment groups and defined genomic regions have colours assigned, which will then propagate through the workflow for consistency of visualisation. External features and/or HiC data is summarised if provided and the association between these datasets and defined genomic regions is also provided as part of the output.

Greylisted regions are also determined and prepared for exclusion through the workflow, along with the provided blacklisted regions.

2.2 Peak Calling

This section of the workflow uses `macs2 callpeak`[Zhang et al., 2008] with default parameters. Peaks are called on each individual sample and by merging samples within treatment groups for each provided ChIP target. One summary report will be produced each ChIP target specified in the `target` column of `samples.tsv`, which will assess all treatments within a ChIP target.

Quality Assessment

Basic QC statistics such as Library Size, Read Lengths, Total Detected Peaks and Fraction Of Reads In Peaks (FRIP) are provided in tabular and visual form. Plots showing sample-specific GC content and Cross Correlations are also provided to enable visual identification of any outlier samples which can be excluded manually, or handled in any other suitable manner.

One feature of the GRAVI workflow is to compare between replicates within a treatment group to determine any low-quality samples. Taking the median number of peaks within a sample group (np_i), and sample with $>N$-fold or $<N$-fold np_i peaks will be marked as an outlier. By default, this is set to 10-fold using the parameter `outlier_threshold` (Section 5.1) and can be disabled by setting this parameter to `.inf`, which will be passed to `R` as the value ∞. Given that some ChIP targets may be cytoplasmic and yield zero peaks in some treatment groups, samples with zero peaks can also be allowed by setting `allow_zero` to be `true`.

Results

Individual Replicates: UpSet plots[Conway et al., 2017] of all samples are first produced to enable identification of any possible mislabelling. Venn Diagrams are then produced for all replicates within each treatment groups (if $n \leq 3$), or UpSet plots are produced for $n \geq 4$ replicates to show consistency between replicates.

Treatment Peaks: During this step of the workflow a set of *treatment-specific peaks* will be produced for each ChIP target. The set of peaks obtained by merging samples will be compared to all individual samples, and only those peaks identified in at least $100 * p\%$ of the samples which passed QC will be included in the set of Treatment peaks. This parameter can be set in `config.yml` as the parameter `min_prop_reps` (Section 5.1) with the default of 0.5 indicating that peaks must be detected in 1 or more samples (if N = 2), two or more samples (if N = 3, 4) etc.

Union Peaks: After defining the treatment-specific peaks, these are combined to define the set of *union peaks*. Treatment peaks are merged across conditions in an inclusive manner, such that the range of each union peak encompasses the entire region covered by all overlapping treatment peaks. Union peaks are then used as the 'universe' of peaks against which treatment-specific behaviours are compared.

Peaks overlapping a blacklisted or greylisted region are excluded at all steps of the analysis.

The final HTML report produced for each target will contain:

1. A Venn Diagram showing the overlap in Union peaks and whether one or more Treatment peaks overlaps the Union peak. In the case of 4 or more treatments, an UpSet plot will be produced instead
2. Distance to TSS plots as actual and cumulative distributions using union peaks.
3. A pie chart describing the distribution of union peaks within the genomic regions defined previously

Additional plots will be produced in the presence of external data

1. The distribution of union peaks within external features will be shown *if external features are provided*
2. The distribution of union peaks within external features and genomic regions will be shown *if external features are provided*

Highly-ranked union and treatment-specific Treatment peaks are also shown in their genomic context alongside regulatory/external features and nearby gene models.

Key Outputs

- Union peaks will be exported as `output/macs2/<target>/<target>_union_peaks.bed`
- Treatment peaks will be exported as individual bed files for each treatment group (`output/macs2/<Ctarget>/<target>_<treat>_treatment_peaks.bed`), as well as combined in a single `RDS` object for faster parsing into an `R` environment as `output/macs2/<target>/<target>_treatment_peaks.rds`
- BigWig files will be generated from the bedgraph files output by `mcas2 callpeak` using the `--SPMR` tag, for each individual sample and for each set of peaks called by merging samples from the same group
- Additional BigWig files will be produced from the bedgraph files output by `macs2 bdgcmp`, representing fold-enrichment over the input sample for each set of grouped samples

2.3 Differential Signal

This step provides much of the uniqueness to the GRAVI workflow combining approaches from `macs2`[Zhang et al., 2008], `qsmooth`[Hicks et al., 2018], `csaw`[Lun and Smyth, 2014], `limma`[Ritchie et al., 2015], `edgeR`[Robinson et al., 2010] and `ihw`[Ignatiadis et al., 2016] whilst relying heavily on the infrastructure provided by `extraChIPs`. A sliding window approach as advocated by Lun & Smyth (2014) is the primary strategy used for detection of differetial signal.

All two-factor comparisons of interest can specified via the YAML file (Section 5.1) with more complicated layouts specified as below

```
contrasts:
  - ["control", "treat1"]
  - ["control", "treat2"]
```

Differential signal analysis will be performed for every ChIP target where both treatment groups are found from each specified contrast. Two possible approaches are available, with these being 1) `sq-lt` (Smooth-Quantile limma-trend) or 2) `ls-ql` (Library-Size Quasi-Likelihood).

Overview of steps in the differential signal workflow. Primary R packages used for each step are indicated in brackets. Integration with differential expression results (RNA-Seq) is an option step only performed if RNA-Seq data is provided.

Sliding Windows

By default, sliding windows will be defined based on the estimated fragment length so that the window size is slightly wider than the fragment length (w_{size}), and the step size is $w_{\text{step}} = w_{\text{size}}/3$. This usually leads to window sizes which are multiples of 30nt, e.g. 90, 120, 150, 180, 210, 240 etc.

Alignments are initially counted across autosomes and sex chromosomes, explicitly excluding scaffolds and mitochondrial alignments, as well as regions contained in black/greylists. Windows are discarded if the total number of alignments across all samples are $< n$, where n represents the total number of samples in the current analysis. Windows which are more likely to contain noise than true signal are then discarded using `extraChIPs::dualFilter()` in combination with the *Union peaks* from Section 2.2. The set of Union peaks is used as a guide for setting the inclusion/exclusion thresholds which are based on 1) Overall signal intensity and 2) Enrichment over input. Thresholds for each measure are defined such that the proportion q of windows which overlap a Union peak are returned, with windows which pass *both* inclusion thresholds are returned. This parameter itself is set in `config.yml` as `filter_q` as described in Section 5.1. In general, values in the range $0.4 < q < 0.6$ perform well as the sliding windows around the margins of the Union peaks will be discarded at this point. Higher values will lead to large numbers of windows being retained which overlap peak margins and are relatively uninformative.

Normalisation

Under the `sq-lt` method, logCPM values are first calculated using the *total library size* for each sample. These values are then normalised using Smooth Quantile Normalisation[Hicks et al., 2018] specifiying treatment groups as the factor of interest and allowing for normalisation within and between groups. This can be particularly useful in the scenario where a transcription factor (TF) is primarily cytoplasmic in one treatment group, before shifting to the nucleus in response to treatment. Plots showing qsmooth weights, pre/post logCPM, pre/post RLE [Gandolfo and Speed, 2018] and pre/post PCA are produced in this section of the workflow.

Alternatively, under the `ls-ql` methodology, only the *total library size* is provided to the GLM fitting stage with no further normalisation strategies being employed. This is analogous to the default `edgeR` option provided using DiffBind [Stark and Brown, 2011].

Hypothesis Testing

Given the normalised logCPM values being analysed under the `sq-lt` approach, an appropriate strategy for detection of differential signal is the `limma-trend` method [Law et al., 2014] as established for RNA-Seq analysis. Under the `ls-ql` approach, raw counts are analysed using the Quasi-Likelihood approach for Negative Binomial data [Lun et al., 2016], again as developed for RNA-Seq data, and as offered when using DiffBind [Stark and Brown, 2011].

Both strategies are performed on *all retained windows* using a **range-based Null Hypothesis** [McCarthy and Smyth, 2009], such that a minimum change can be specified, below which changes in signal are not considered to be of interest. Under this statistical approach, the Null Hypothesis would be

$$H_0 : -\lambda \leq \mu \leq \lambda$$

with the alternate hypothesis being

$$H_A : |\mu| > \lambda$$

By default, the value $\lambda = \log_2(1.2)$ is used, which denotes a 20% change in signal as the point where sites become of interest. This value is set in `config.yml` as the parameter `fc` (Default: `fc = 1.2`), and setting `fc = 1` would return the statistical approach to be a point-based Null Hypothesis $H_0 : \mu = 0$.

After performing this statistical test, overlapping windows are *merged* taking the individual window with the **highest signal** as the representative window for the merged region. As this value is independent of the statistical test[Lun and Smyth, 2014], the resultant set of p-values are FDR-adjusted using the Benajmini-Hochberg approach[Benjamini and Hochberg, 1995], giving a set of FDR-adjusted p-values for all merged regions.

Independent Hypothesis Weighting

The Independent Hypothesis Weighting approach[Ignatiadis et al., 2016] suggests the a set of p-values can be partitioned by any independent variable, with weights assigned to each partition, and these weighted p-values can then be adjusted using conventional strategies, such as the FDR. Under the GRAVI workflow, four possibilities for partitioning the p-values obtained after merging regions are provided, and these can be specified in the `ihw` parameter of `config.yml`. The options are

1. `ihw: "targets"` where union peaks from all other ChIP targets included in the larger GRAVI workflow are used to partition p-values. As union peaks are treatment-agnostic, these simply provide a scaffold defining the presence/absence of all other ChIP targets under any treatment condition, in combination across all ChIP targets.
2. `ihw: "regions"` where previously defined genomic regions are used to partition the p-values

3. `ihw: "features"` where any external features supplied are used for the partitioning
4. `ihw: "none"` where no partitioning is performed and the standard FDR-adjusted p-values are used to determine differential signal status

The default approach implemented by the IHW authors suggests that p-value partitions must be > 1000, and as such, all the above approaches collapse smaller groups until the smallest group contains at least 1000 p-values (i.e. merged regions). In the case of ChIP targets which do not bind in a promiscuous manner, the IHW step may make minimal difference to the results.

Assigning Genes To Windows

After merging of neighbouring sliding windows, genes are assigned to each merged region. This is performed using annotated genomic regions, any external features and HiC data, via the function `extraChIPs::mapByFeatures()`.

Under this approach:

1. Regions which overlap a promoter are assigned to the *genes associated directly with that promoter*
2. Regions which overlap an enhancer are assigned to *all genes within 100kb*
3. Regions which overlap any HiC interactions are assigned to *all genes connected by the interactions*
4. Regions with no assignment from steps 1-3 are assigned to *all directly overlapping genes*, or the *nearest gene within 100kb*

If no HiC data is included, step 3 is not performed

Presentation of Key Results

The above steps essentially complete the detection of differential signal regions and assignment of these to regulatory target genes. A series of visualisations are then provided including:

1. MA & Volcano plots
2. Profile Heatmaps for sites with Increased/Decreased target signal
3. Results summarised by chromosome
4. Signal/logFC/differential signal partitioned by genomic region

If *external features are provided*, the plots from step 4 are replicated using external features to partition results. A combined summary of the results by genomic region and external features is also included.

A series of summary tables, along with the 200 most highly ranked regions are included. Results of differential signal windows and the genes assigned to them are also exported as a CSV for sharing with collaborators. Genomic plots for unique regions based on signal strength (logCPM), changed signal (logFC) and statistical support (FDR) are also provided as part of the default output.

Enrichment Testing

Using the gene-sets and pathways specified in `params.yml` four enrichment analyses are performed using `goseq`[Young et al., 2010]. These are

1. Comparison of genes mapped to a site with **target signal** and genes which are not mapped to a region with target signal
2. Comparison of genes mapped to a **differential signal site** against genes mapped to a site with detected signal, but which is considered unchanged
3. Comparison of genes mapped to a site with **increased signal** against genes mapped to a site, but with no differential signal
4. Comparison of genes mapped to a site with **decreased signal** against genes mapped to a site, but with no differential signal

Gene-width is used as an offset for biased sampling in the analysis of bound target only, as longer genes are more likely to have a peak mapped to them. For all other analyses, no offset term is included and the standard Hypergeometric test is performed.

All results are presented as searchable, interactive HTML tables including which genes from each pathway are mapped to the relevant set of regions from which signal was detected. Where enough pathways are detected as significantly enriched, network plots are also produced deriving distances between nodes based on shared, differential-signal genes using the *overlap coefficient*.

RNA-Seq Data

If results from a differential expression (DE) analysis are included (which can also be microarray results), a series of additional analyses are performed. Firstly, the relationship between detected genes and the genomic regions where target signal has been detected are characterised. If external features are also provided, this is repeated incorporating these features.

P-values from DE analysis are partitioned by differential signal status. No further IHW is performed, but this can still provide a clear visual clue as to the relationship between target signal and differential gene expression.

Genomic regions for the 5 genes most highly ranked for differential expression, and with > 1 target-bound window assigned to them are plotted.

Given the unpredictability of an RNA-Seq dataset and the number of genes considered to be differentially expressed, bound and differential signal windows are used as gene sets to perform GSEA[Korotkevich et al., 2019] using sites directly, and sites partitioned by genomic region and external feature (if provided). GSEA is performed 1) taking direction of differential expression into account, 2) ranking genes purely by significance (i.e. without direction).

Finally, standalone GSEA results from the RNA-seq dataset (incorporating direction of change and all requested `MSigDB` pathways) are compared to enrichment (i.e. `goseq`) results for differential signal regions. As these are essentially orthogonal viewpoints on gene regulation, P-values from each analysis are combined using Wilkinson's `maximump()` method and the resultant set of p-values adjusted using the specified method and requested threshold for α Any common pathways are given in an interactive HTML table along with network plots based on shared regulatory targets also found in the leading edge fro GSEA.

Key Outputs

Key output files produced by this step of the workflow are:

- Differential signal results (by window): `output/differential_binding/<target>/<target>_<control>_<tre`
- Differential signal results (by gene): `output/differential_binding/<target>/<target>_<control>_<treat`
- Regions with increased target signal: `output/differential_binding/<target>/<target>_<control>_<treat`
- Regions with decreased target signal: `output/differential_binding/<target>/<target>_<control>_<trea`

2.4 Pairwise Comparisons

By default, pair-wise comparisons are performed between all differential signal results. If ChIP target TF1 has samples in Control and Treat1, whilst ChIP target TF2 has samples in Control, Treat1 and Treat2, with comparisons being requested for Treat1 vs. Control and Treat2 Vs Control, the following pair-wise comparisons will be automatically performed:

1. TF1 (Treat1 Vs Control) and TF2 (Treat1 Vs Control)
2. TF1 (Treat1 Vs Control) and TF2 (Treat2 Vs Control)
3. TF2 (Treat1 Vs Control) and TF2 (Treat2 Vs Control)

Given the automated nature of the workflow, this represents the simplest approach and any redundant comparisons are simply able to be ignored by the user.

Comparison of Peaks

The pair-wise comparisons module initially compares union peaks between the two targets as a Venn Diagram, with the sets of (up to) four treatment-specific peaks being compared using an UpSet plot.

Comparison of Differentially Bound Windows

Pair-wise comparison of two ChIP targets requires more nuance than simply looking for sites where both are changed. A universal set of windows is first obtained across all windows retained in both targets. These are then classified as either *Up*, *Down*, *Unchanged* or *Undetected* for each ChIP target. Using both targets, the universal windows are then classified based on **both targets**. This is particularly important given the hypothesised role of ChIP targets which can act as pioneer factors[Zaret and Carroll, 2011], or those which bind in complex and sequester other factors[Hickey et al., 2021].

The classification of each window is initially based on significant FDR-adjusted p-value using the range-based H_0 in at least one comparison. In order to ensure more accurate assignment of windows in the secondary comparison, the FDR-adjusted p-values using a *point-based* H_0 are used, in conjunction with an estimated logFC beyond the range $\pm\lambda$, i.e. $|\widehat{\text{logFC}}| > \lambda$. This reduces the number of regions incorrectly classified as unchanged in one comparison due to the use of the range-based H_0, as is common for all situations where threshold are applied.

The combined behaviours of both ChIP targets is then compared directly, described by genomic region and external features (if provided). Distances between the windows with representative statistics (i.e. maximal signal) are determined where both targets are present. The combined changes in signal are then compared as a complete set for all windows where both targets are detected, as well as broken down by genomic region and external features. Distances between the windows where the maximal signal was observed are also presented

Combined Visualisations

Profile heatmaps for all behavioural groups which are detected (e.g. TF1 Up - TF2 Up, TF1 Up - TF2 - Unchanged etc) are all presented.

The genomic windows for which the two factors are present are visualised based on the combined strongest signal, and the combined largest change. Using only combinations of Up/Down/Unchanged, six windows for each are presented. As with all genomic visualisations, any coverage provided as an external track (e.g H3K27ac or ATAC-seq signal) will be added to all plots.

Enrichment Analysis

The same gene-sets from MSigDB[Liberzon et al., 2015] as used previously are then used for enrichment testing, using genes as mapped to windows during previous steps. Enrichment testing again uses `goseq` with gene length as the biased-sampling term. Enrichment is performed at multiple levels:

1. Genes Mapped To All Windows
    1. Genes mapped in *either comparison*
    2. Genes mapped in *comparison 1 but not the second comparison*
    3. Genes mapped in *comparison 2 but not the first comparison*
    4. Genes mapped in *both comparisons*
2. Genes Mapped to Differential Signal Windows
    - All pair-wise combinations of Up/Down/Unchanged/Undetected are tested across both factors. If no enrichment is found, results are not presented.

Where enrichment for pathways are found, network plots are again generated

Integration With RNA-Seq Results

The set of genes mapped to each pair-wise combination of Up/Down/Unchanged/Undetected are then compared to the external DE results using GSEA incorporating direction of change, and overall significance. Differential expression is displayed as a multi-panel volcano plot, separating genes by detected signal patterns. Along with barcode plots for the 9 most highly-ranked groups, genes in the Leading Edge for *all* significant combined groups are also provided in the results table, along with network plots. Significantly enriched pathways are again determined by combining p-values from GSEA performed on RNA-Seq and enrichment testing performed sites returning signal, using Wilkinson's method.

2.4.1 Key Outputs

Key output files produced by this step of the workflow include

- Pairwise differential signal patterns across both comparisons (`output/pairwise_comparisons/<target1>_<tar`
- The above file as an RDS object for easier parsing into an `R` environment.

If RNA-Seq data is provided, the file `output/pairwise_comparisons/<target1>_<target2>/<target1>_<comparis` will contain mappings between DE genes and combined ChIP target differential signal patterns. All lists of enriched pathways will also be exported as csv files.

Chapter 3

Quick Start Guide

3.1 Install Snakemake

You will need a snakemake installation to begin. Please see here for help setting this up. If you are running the pipeline on an HPC and are unsure, please consult with your HPC support team about setting up a snakemake profile on your specific cluster.

3.2 Create the Directory Structure

1. Create a new github repository on your account by going to the github template repository
2. Download your new repository to your local server or HPC using `git clone <myrepository>`
3. Place your bam files in the subdirectory `data/bam` or `data/aligned` as described in section 4.2
4. Edit `samples.tsv` in the `config` directory as described in section 4.3
5. Ensure you have the blacklist as a bed file and annotations as gtf
6. Modify any parameters in `config/config.yml`

3.3 Run the Pipeline

3.3.1 Run On A Local Server

To run using 16 cores without any queuing system (e.g. on a local machine), enter the following

```
snakemake -p --use-conda --notemp --keep-going --rerun-triggers mtime --cores 16
```

3.3.2 Run On An HPC

Please consult with your local support team for their advice running a `snakemake` workflow. In essence, the above command will need to be provided to your queuing system through the preferred strategy. The `snakemake` profile required will generally be stable across all workflows but may require expertise from the technical support team.

3.4 Tips And Tricks

3.4.1 Removing Large files

Some large files, such as R Environments and BedGraph files are marked as `temp` files internally and these can be removed after completion of the workflow using

```
snakemake --delete-temp-output --cores 1
```

3.4.2 Shared Conda Environments

`conda` environments can easily become bloated and if running multiple GRAVI analyses it may be simpler to host a common set of `conda` environments to avoid their constant recreation. This can be performed by adding the argument `--conda-prefix '/path/to/my/envs/'` in the call to `snakemake`

Conda environments can also be built prior to running the workflow using the standalone command

```
snakemake \
    --use-conda \
    --conda-prefix '/path/to/my/envs/' \
    --conda-create-envs-only \
    --cores 1
```

3.4.3 Running Restricted Sections of the Worklow

Snakemake has the capacity to run a workflow up until a certain point and this can be easily done using the argument `--until` and specifying the stage you wish to terminate the workflow at. For example, the argument `--until compile_macs2_summary_html` would only run the workflow until the macs2 summaries are compiled, which may be preferable for checking QC before proceeding to differential expression and pairwise comparisons.

Chapter 4

Files and Directories

4.1 Directory Structure

The GRAVI workflow requires a set directory structure. If using the template repository, as advised, this will be mostly taken care of. The required directory structure is

```
project_home/
    analysis
    config
    data
    docs
    output
    workflow
```

- Rmarkdown scripts will be added to and executed from the `analysis` directory
- Key configuration files are provided in the `config` directory
- *Your data* should be placed in the `data` directory as described below
- The `html` output summarising all results will be produced in the `docs` directory
- Additional output files will be placed in `output` with all figures written within the respective folder for each html page, or with `docs/assets`.
- The workflow itself is run by all code supplied in the `workflow` directory
- The complete `R Environment` for each compiled RMarkdown document is also saved in `output/envs`, and given their larger sizes, these can be deleted using `snakemake --delete-temp-output --cores 1` to conserve storage space.

4.2 Alignments

The GRAVI workflow currently takes `bam` files as the primary input. Multiple workflows exist for quality control, adapter removal and de-duplication and it is assumed that supplied reads will have been pre-processed with the above steps, then aligned to the genome of interest.

Files should be placed in the `data/bam` directory as set in `config.yml`, although this can be changed if desired.

```
project_home/
    data
         bam
              target1_control_rep1.bam
              target1_control_rep2.bam
              target1_control_rep3.bam
              target1_treat_rep1.bam
              target1_treat_rep2.bam
              target1_treat_rep3.bam
              target2_control_rep1.bam
              target2_control_rep2.bam
              target2_control_rep3.bam
              target2_treat_rep1.bam
              target2_treat_rep2.bam
              target2_treat_rep3.bam
              input1.bam
```

4.3 Sample Descriptions

The file `samples.tsv` defines the set of files which the workflow will be applied to. Any files placed in the `data/aligned` directory, but not specified in this file will be ignored. The desired layout should be a *tab-delimited file* (i.e. tsv). These can be generated using Excel, Notepad++, R, Visual Studio, or any other software you are comfortable with. A brief example would follow the layout

sample	target	treat	replicate	input
target1_control_rep1	Target1	Control	1	input1
target1_control_rep2	Target1	Control	2	input1
target1_control_rep3	Target1	Control	3	input1
target1_treat_rep1	Target1	Treat1	1	input1
target1_treat_rep2	Target1	Treat1	2	input1
target1_treat_rep3	Target1	Treat1	3	input1

4.3.1 Required columns

This file must contain all four of the columns `sample`, `target`, `treat`, `input`, in any order. If supplied, optional columns such as `replicate`, `passage` etc can be referenced in the workflow. As well as defining all required steps for the workflow, labels for plots will be generated from combinations of these columns.

- `sample`: This **must** be identical to the filename, but without the `.bam` extension.
- `treat`: This is used to define all comparisons
- `input`: All files must correspond to a file in `data/bam` but without the `.bam` suffix. Each sample can have a separate input sample, or input samples can be shared across all or some of the samples.

4.3.2 Optional Columns

Any additional columns can be used to denote batches, or passages if running a nested/paired model. These column names will be automatically detected at the appropriate steps of the workflow and incorporated into figures and tables. Common column names may be `replicate` or `passage` (for cell lines)

4.4 Additional Files

Additional, optional files can also be supplied and is it customary to place these in `data/external` with paths (relative to `project_home`) added to `config.yml`. Full details are demonstrated in Section 5.1 Names can be any informative name chosen by the user.

```
project_home/
   data
        bam
        external
             gencode_annotation.gtf.gz
             blacklist.bed.gz
             rnaseq_topTable.tsv
             external_features.gtf.gz
             hic_interactions.bedpe
             additional_coverage_control.bw
             additional_coverage_treat.bw
```

4.4.1 RNA-Seq

Files provided with differential expression analysis results from a relevant RNA-Seq experiment should follow the layout as produced by `topTable()` from the `limma` package[Ritchie et al., 2015]. Gene IDs should match those in the Gencode GTF (Ensembl IDs) and should be contained in a column called `gene_id`.

Additional expected columns will be `logFC` and `FDR` or similar names which could be reasonably found by `regex` matching within the workflow.

4.4.2 External Features

These must be provided as a GTF which can be prepared by any method. The feature types should be defined in a **field named `feature`.** Non-overlapping features are optimal but not essential, and this is left to the users discretion. For example, if providing features such as enhancers and super-enhancers[Whyte et al., 2013], it may be more sensible to provide these as mutually exclusive groups.

4.4.3 HiC Interactions

Significant interactions can be sourced using any methodology, however these must be provided in `bedpe` format.

4.4.4 External Coverage

Additional coverage files should be provided in `bigwig` format.

Chapter 5

YAML Configuration Files

All YAML files which ruin the workflow are located in the `project_home/config` directory. The standard YAML structure is used in all files with the primary objective being passing workflow parameters to the various steps of the workflow. There are four files which control various aspects: `config.yml`, `colours.yml`, `params.yml` and `rmarkdown.yml`

5.1 The Main Configuration: `config.yml`

This file sets many of the primary parameters and is the file which *will need editing for any new dataset.* Many settings should remain unchanged as changing default locations of files may lead to unexpected instability in the workflow, whilst other setting **should** be changed, such as those which determine which comparisons to perform. This is the only file parsed directly by `snakemake` and subsequent rules, whilst all others are used to pass parameters to `R` environments.

An example layout of `config.yml` might be:

```
samples:
  file: "config/samples.tsv"

paths:
  bam: "data/bam"

genome:
  build: "GRCh37"

external:
  blacklist: "data/external/blacklist.bed.gz"
  gtf: "data/external/gencode.v33lift37.annotation.gtf.gz"
  rnaseq: "data/external/some_results.tsv"
  features: "data/external/custom_features.gtf.gz"
  hic: "data/external/hic_interactions.bedpe"
  coverage:
    H3K27Ac:
      control: "data/external/H3K27Ac_control.bw"
      treat: "data/external/H3K27Ac_treat.bw"

comparisons:
  method: 'sq-lt'
  fc: 1.2
  fdr: 0.05
  paired: false
  filter_q: 0.6
  contrasts:
    - ["control", "treat"]
  ihw: "regions"

peaks:
  macs2:
    gsize: "hs"
    fdr: 0.05
    keep_duplicates: "all"
  qc:
    outlier_threshold: 10
    allow_zero: true
    min_prop_reps: 0.5
```

Settings Which Don't Need To Be Modified

In general, the paths to key files don't need to be changed and default configurations are well tested. Whilst varying these has been intermittently attempted successfully, unexpected instability may occur and as such, is discouraged.

`samples:` (Default: "config/samples.tsv")

By default, the file which contains all sample-level information is defined as `config/samples.tsv`. This can be changed to reflect any requirements after inspecting QC reports.

`paths:`

- `bam:` (Default: "data/bam") Alignments should be placed in `data/bam` as advised in section 4.2, although this can be changed to any other path as desired.

Settings Which Should Be Modified

`genome:`

Specify the genome build used for alignments and for gene annotations.

`external:`

Provide paths to all additional data files here. Only files provided will be included in the workflow.

- `blacklist` Should be obtained from here
- `gtf`: Should ideally be obtained from Gencode
- (Optional) `rnaseq` should be the results as output by `limma::topTable()` or similar. Gene IDs should match those provided in the Gencode GTF (e.g. Ensembl IDs) and these should be in a column names `gene_id`. Columns such as `logFC` and `FDR` will be searched for during the workflow using regular expressions to find the best match. Can **only** be in `csv` or `tsv` format. Excel-specific (`xls`, `xlsx`) formats are not supported.
- (Optional) `features` can be determined by any method, with common choices being relevant histone marks, or promoters, enhancers and super-enhancers determined by H3K27ac marks. Features should be non-overlapping with the field `feature` defining the different feature types. Must be provided in GTF format.
- (Optional) `hic` HiC interactions must be provided as a `bedpe` file
- (Optional) `coverage` Tracks provided in this argument will be added to all genomic plots showing binding peaks or differential binding. Multiple files provided within each YAML list element will be overlaid as a single track. There is no upper limit to the number of tracks, however more tracks generally detract from an informative figure.

`comparisons:`

These settings determine how the differential binding analysis is performed.

- `method` (Default: "sq-lt") Defines the strategy used for differential binding. The currently implemented options are "sq-lt" and "ls-ql" as described previously
- `fc` (Default: 1.2) The setting of 1.2 indicates a 20% change in binding as the threshold below which we are not interested, or below which we consider binding changes to be inconsequential. This parameters is passed to `limma::treat()` [McCarthy and Smyth, 2009] in all differential binding analyses.
- `fdr` (Default: 0.05) Windows with significance below this threshold are considered to provide supporting evidence of differential binding.
- `paired` (Default: `false`) If set to `true` the values in the optional column (e.g. replicate, passage etc.) are used to perform a paired analysis as described in the `limma` manual
- `filter_q` (Default: 0.6) Passed to `extraChIPs::dualFilter()`. When

filtering (i.e. discarding) genomic sliding windows which are unlikely to contain true binding signal, determine thresholds which will retain this proportion of windows which overlap a peak identified by `macs2 callpeak`.

- `contrasts`: Define all contrasts for differential binding. Any ChIP target containing both treatment groups will be included for differential binding. Values must match those in the `treat` column of `samples.tsv`. Each differential binding analysis will be performed using the limma-trend method in the context of A vs B, such that complex models are not supported. Use new YAML list elements to define additional contrasts
- `ihw` (Default: "regions") Options used to stratify p-values for Independent Hypothesis Weighting[Ignatiadis et al., 2016] of differential binding results. Can take values `"regions"`, `"features"`, `"targets"` or `"none"`
  - `"regions"` P-values will be stratified by annotated genomic regions as determined in the initial steps of the workflow
  - `"features"` P-values will be stratified by provided external features
  - `"targets"` P-values will be stratified by the presence of a union peak using all other ChIP targets
  - `"none"` No Independent Hypothesis Weighting will be performed on the results from differential binding

```
peaks:
```

`macs2` settings are passed to `macs2 callpeak`. Only the arguments `gsize`, `fdr` and `keep_duplicates` are accepted. Please see the `macs2` manual for more detailed explanations.

`qc` parameters are used for determining if samples are of a high enough quality, and how to determine union/treatment peaks for each target and treatment group.

- `outlier_threshold` (Default: 10) As previously described, samples with peak numbers beyond this value relative to the median peak numbers in each treatment group are marked for removal. To ignore this parameter, set this to `.inf` which effectively sets the threshold to ∞
- `allow_zero` (Default: `true`) Allow samples with no identified peaks. If a ChIP target is expected to be cytoplasmic in one condition, this can allow samples with no peaks to be retained.
- `min_prop_reps` (Default: 0.5) When forming **treatment-specific** peaks within each *target and treatment group*, peaks must be represented in at least this proportion of samples. This defaults to 0.5 which would equate to 2 of 3 samples, 2 of 4 samples etc. This value may need to be altered pending the results of a complete run after Quality Assessment has performed.

5.2 Colour Schemes: `colours.yml` [#colours-yml]

Defines all plotting colour schemes for consistency throughout the entire workflow. Colours can be any standard colours able to be interpreted by `R`, such as `'blue'` or `'#0000FF'`. Recommended YAML list elements are `qc`, `regions`, `direction` and `treat`. A gradient through al colours provided in the `heatmaps` element will be generated during generation of heatmaps. Any unspecified colours will be automatically assigned and will propagate through the workflow. As is standard across most programming languages, names are case-sensitive. An example file is given below:

```
qc:
  pass: "#0571B0"
  fail: "#CA0020"
regions:
  promoter: '#FF3300'
  upstream_promoter: '#E1EE05'
  exon: '#7EDD57'
  intron: '#006600'
  proximal_intergenic: '#000066'
  distal_intergenic: '#551A8B'
treat:
  Input: "#33333380"
  E2: "#3333CC"
  E2DHT: "#C52240"
features:
  enhancer: "#FFFF00"
  no_feature: "#E5E5E5"
direction:
  up: "#CA0020"
  down: "#0571B0"
  unchanged: "#7F7F7F"
  undetected: "#E5E5E5"
heatmaps: ['white', 'red']
```

5.3 Additional Parameters: `params.yml`

Default settings for defining initial annotations, mapping of peaks to genes and enrichment testing. In general, these will not need to be changed, but can be if required. In particular, the value **enh2gene** can be set to zero if using data such as H3K27ac HiChIP to determine long-range interactions.

For enrichment testing, the preferred p-value adjustment method can be specified using any value that match `p.adjust.methods` with an appropriate threshold also set in this section. Default settings are relatively inclusive and more stringent setting may be preferred by some users.

Parameters for msigdb[Liberzon et al., 2015] are passed to msigdbr[Dolgalev, 2022] and fields should match this layout. Any categories passed to gs_cat will lead to all subcategories being used from that category. Specific sub-categories of larger databases can be passed using gs_subcat element.

Values which determine the minimum and maximum size of any network plots are also defined here. Node-pair distances above the given distance will be removed from the plot and these edges excluded.

```
gene_regions:
  promoters:
    upstream: 1500
    downstream: 500
  upstream: 5000
  intergenic: 10000

## The values used when mapping peaks to genes.
## Passed to `extraChIPs::mapByFeature()`
## If including H3K27ac HiChIP for long range-interactions, it is advised to
## set `enh2gene` as zero, given that long range interactions in this case
## will more accurately map long-range enhancer interactions than using all
## genes within a given distance
mapping:
  gr2gene: 100000
  prom2gene: 0
  enh2gene: 100000
  gi2gene: 0

enrichment:
  adj: "fdr"
  alpha: 0.05
  ## Only gene-sets between these two values will be retained before testing
  min_size: 5
  max_size: 1000
  ## Only gene-sets above this size will be shown in the results
  min_sig: 3
  ## The categories to use from MSigDB. These are passed to msigdbr in the
  ## columns of the same name in an 'OR' approach
  species: "Homo sapiens"
  msigdb:
    gs_cat: "H"
    gs_subcat:
      - "CP:KEGG"
      - "CP:REACTOME"
      - "CP:WIKIPATHWAYS"
      - "TFT:GTRD"

## Used for network plots
networks:
  min_size: 4
  max_size: 80
  max_distance: 0.9
  layout: 'fr'
```

5.4 HTML Settings: `rmarkdown.yml`

The main workflow will produce a compiled set of HTML pages using `rmarkdown::render_site()`[Allaire et al., 2022]. The two available fields to supply here are:

1. `knitr_opts` which are passed to `knitr::opts_chunk$set()`[Xie, 2022] at the beginning of every compiled Rmarkdown document, and
2. `rmarkdown_site` which determines the layout and style of the final HTML report. All *left* elements of the `navbar` are determined automatically during the workflow and will be ignored if supplied here, whilst all other parameters are passed via the file `_site.yaml` which will be generated during the workflow.

Figure height and width parameters must be passed to `knitr::opts_set()` in inches, and the workflow defaults to `svg` output for most figures. Some figures which can become excessively large under this setting, such as those with many thousands of overlapping points, are set to always produce `png` output. Multiple output formats can also be set using values such as `['png', 'pdf']` if preferred.

```
knitr_opts:
  echo: TRUE
  message: FALSE
  warning: FALSE
  dev: ["png", "pdf"]
  fig.align: "center"
  fig.width: 10
  fig.height: 8

rmarkdown_site:
  name: "GRAVI: Gene Regulatory Analysis"
  output_dir: "../docs"
  navbar:
    title: "GRAVI"
    right:
      - icon: fa-github
        href: "https://github.com/smped/GRAVI"
```

```
output:
  html_document:
    toc: yes
    toc_float: yes
    code_folding: hide
    self_contained: false
    theme: sandstone
    highlight: textmate
    css: gravi.css
    includes:
      after_body: footer.html
```

Bibliography

JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. *rmarkdown: Dynamic Documents for R*, 2022. URL https://CRAN.R-project.org/package=rmarkdown. R package version 2.14.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995. ISSN 00359246. URL http://www.jstor.org/stable/2346101.

John D Blischak, Peter Carbonetto, and Matthew Stephens. Creating and sharing reproducible research code the workflowr way [version 1; peer review: 3 approved]. *F1000Research*, 8(1749), 2019. doi: 10.12688/f1000research.20843.1. URL https://doi.org/10.12688/f1000research.20843.1.

Jake R. Conway, Alexander Lex, and Nils Gehlenborg. Upsetr: An r package for the visualization of intersecting sets and their properties. *Bioinformatics*, 33(18):2938–2940, 2017. doi: 10.1093/bioinformatics/btx364.

Igor Dolgalev. *msigdbr: MSigDB Gene Sets for Multiple Organisms in a Tidy Data Format*, 2022. URL https://CRAN.R-project.org/package=msigdbr. R package version 7.5.1.

L. C. Gandolfo and T. P. Speed. RLE plots: Visualizing unwanted variation in high dimensional data. *PLoS One*, 13(2):e0191629, 2018.

T. E. Hickey, L. A. Selth, K. M. Chia, G. Laven-Law, H. H. Milioli, D. Roden, S. Jindal, M. Hui, J. Finlay-Schultz, E. Ebrahimie, S. N. Birrell, S. Stelloo, R. Iggo, S. Alexandrou, C. E. Caldon, T. M. Abdel-Fatah, I. O. Ellis, W. Zwart, C. Palmieri, C. A. Sartorius, A. Swarbrick, E. Lim, J. S. Carroll, and W. D. Tilley. The androgen receptor is a tumor suppressor in estrogen receptor-positive breast cancer. *Nat Med*, 27(2):310–320, 02 2021.

Stephanie C. Hicks, Kwame Okrah, Joseph N. Paulson, John Quackenbush, Rafael A. Irizarry, and Hector Corrado Bravo. Smooth quantile normalization. *Biostatistics*, 19(2), 2018.

N. Ignatiadis, B. Klaus, J. B. Zaugg, and W. Huber. Data-driven hypothesis weighting increases detection power in genome-scale multiple testing. *Nat Methods*, 13(7):577–580, 07 2016.

Gennady Korotkevich, Vladimir Sukhov, and Alexey Sergushichev. Fast gene set enrichment analysis. *bioRxiv*, 2019. doi: 10.1101/060012. URL http://biorxiv.org/content/early/2016/06/20/060012.

C. W. Law, Y. Chen, W. Shi, and G. K. Smyth. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol*, 15(2):R29, Feb 2014.

A. Liberzon, C. Birger, H. Thorvaldsdóttir, M. Ghandi, J. P. Mesirov, and P. Tamayo. The Molecular Signatures Database (MSigDB) hallmark gene set collection. *Cell Syst*, 1(6):417–425, Dec 2015.

Aaron T L Lun and Gordon K Smyth. De novo detection of differentially bound regions for ChIP-seq data using peaks and windows: controlling error rates correctly. *Nucleic Acids Res.*, 42(11):e95, 2014.

Aaron T. L. Lun, Yunshun Chen, and Gordon K. Smyth. *It's DE-licious: A Recipe for Differential Expression Analyses of RNA-seq Experiments Using Quasi-Likelihood Methods in edgeR*, pages 391–416. Springer New York, New York, NY, 2016. ISBN 978-1-4939-3578-9. doi: 10.1007/978-1-4939-3578-9_19. URL https://doi.org/10.1007/978-1-4939-3578-9_19.

Davis J. McCarthy and Gordon K. Smyth. Testing significance relative to a fold-change threshold is a TREAT. *Bioinformatics*, 25(6):765–771, 01 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp053. URL https://doi.org/10.1093/bioinformatics/btp053.

F. Mölder, K. P. Jablonski, B. Letcher, M. B. Hall, C. H. Tomkins-Tinch, V. Sochat, J. Forster, S. Lee, S. O. Twardziok, A. Kanitz, A. Wilm, M. Holtgrewe, S. Rahmann, S. Nahnsen, and J. Köster. Sustainable data analysis with Snakemake. *F1000Res*, 10:33, 2021.

Matthew E Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7):e47, 2015. doi: 10.1093/nar/gkv007.

Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010. doi: 10.1093/bioinformatics/btp616.

Rory Stark and Gordon Brown. *DiffBind: differential binding analysis of ChIP-Seq peak data*, 2011. URL http://bioconductor.org/packages/release/bioc/vignettes/DiffBind/inst/doc/DiffBind.pdf.

W. A. Whyte, D. A. Orlando, D. Hnisz, B. J. Abraham, C. Y. Lin, M. H. Kagey, P. B. Rahl, T. I. Lee, and R. A. Young. Master transcription factors and mediator establish super-enhancers at key cell identity genes. *Cell*, 153 (2):307–319, Apr 2013.

Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2022. URL https://yihui.org/knitr/. R package version 1.39.

Matthew D Young, Matthew J Wakefield, Gordon K Smyth, and Alicia Oshlack. Gene ontology analysis for rna-seq: accounting for selection bias. *Genome Biology*, 11:R14, 2010.

K. S. Zaret and J. S. Carroll. Pioneer transcription factors: establishing competence for gene expression. *Genes Dev*, 25(21):2227–2241, Nov 2011.

Y. Zhang, T. Liu, C. A. Meyer, J. Eeckhoute, D. S. Johnson, B. E. Bernstein, C. Nusbaum, R. M. Myers, M. Brown, W. Li, and X. S. Liu. Model-based analysis of ChIP-Seq (MACS). *Genome Biol*, 9(9):R137, 2008.