GRAVI User Guide (Under Development)

Stephen Pederson

2022-05-04

Contents

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` 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.

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

- Results from a single RNA-Seq experiment
- 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
- *TO BE COMPLETED*: Perform motif analysis based on key binding patterns within and across ChIP targets
- Add key output files to any local `git` repository

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

Input Files and Directories

2.1 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 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`
- The workflow itself is run by all code supplied in the `workflow` directory

2.2 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/aligned` directory as set in `config.yml`, although this can be changed if desired. Each ChIP target should be placed in a separate directory using the example layout as given below. As IgG input may be shared between ChIP targets, these can all be placed in the directory `data/Input` and this is hard-wired into the workflow.

```
project_home/
    data
         aligned
              TF1
                   control_rep1.bam
                   control_rep2.bam
                   control_rep3.bam
                   treat_rep1.bam
                   treat_rep2.bam
                   treat_rep3.bam
              TF2
                   control_rep1.bam
                   control_rep2.bam
                   control_rep3.bam
                   treat_rep1.bam
                   treat_rep2.bam
                   treat_rep3.bam
              Input
                    pooled_input.bam
```

2.3 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
control_rep1	TF1	Control	1	pooled_input
control_rep2	TF1	Control	2	pooled_input
control_rep3	TF1	Control	3	pooled_input
treat_rep1	TF1	Treatment	1	pooled_input
treat_rep2	TF1	Treatment	2	pooled_input
treat_rep3	TF1	Treatment	3	pooled_input

2.3.1 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.
- `target`: This **must** be the directory name within `data/aligned` which contains the sample
- `treat`: This is used to define all comparisons
- `input`: All files must correspond to a file in `data/aligned/Input` 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.

2.3.2 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)

2.4 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`. Names can be any informative name chosen by the user.

```
project_home/
    data
         aligned
              TF1
              TF2
              Input
         external
              rnaseq_topTable.tsv
              external_features.gtf
              hic_interactions.bedpe
              additional_coverage_control.bw
              additional_coverage_treat.bw
```

2.4.1 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.

2.4.2 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.

2.4.3 HiC Interactions

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

2.4.4 External Coverage

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

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 `snakemake` on your specific cluster. `Snakemake` itself is in widespread use globally, so they should be able to provide the support you need.

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/aligned` as described in section 2.2
    - These should be placed in separate directories for each target, such `data/bam/target1` and `data/bam/target2`
4. Edit `samples.tsv` in the `config` directory as described in section 2.3
5. 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 --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.

Chapter 4

Editing The 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

4.1 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.

An example layout of `config.yml` is:

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

paths:
  bam: "data/aligned"
  bigwig: "data/bigwig"
  macs2: "data/macs2"

genome:
  build: "GRCh37"
  gencode: "33"

external:
```

```
  rnaseq: "data/external/results.tsv"
  features: "data/external/h3k27ac_features.gtf"
  hic: "data/external/encode_hic.bedpe"
  coverage:
    H3K27Ac:
      control: "data/external/H3K27Ac_control.bw"
      treat: "data/external/H3K27Ac_treat.bw"

comparisons:
  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:
    min_prop_peaks: 0.1
    min_prop_reps: 0.3
```

4.1.1 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`. In theory, this can be changed but this may adversely impact the pipeline.

`paths:`

- `bam:` (Default: "data/aligned") Alignments should be placed in `data/aligned` as advised in section 2.2, although this can be changed to `data/bam` or any other relative path as desired. Again, changes to the default layout may adversely affect the pipeline stability
- `macs2:` (Default: "data/macs2") Output from `macs2 callpeak` will be placed in this directory, mirroring the input structure from `data/aligned` where each ChIP target (e.g. TF1, TF2) will have results written to separate directories.

- `bigwig:` (Default: "data/bigwig") After running `macs2 callpeak`, `bedGraph` files will be converted to the more space efficient `bigwig` files. The directory structure from both `data/aligned` and `data/macs2` will again be mirrored such that each ChIP target has all samples written to the same directory

4.1.2 Settings Which Should Be Modified

`genome:`

Specify the genome build used for alignments and for gene annotations. Build must match that used exactly for performing alignments. The gtf corresponding to the specified Gencode release will be downloaded as part of the workflow

`external:`

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

- `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 be in `csv` or `tsv` format. Excel-specific (`xls`, `xlsx`) formats are not supported.
- `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.
- `hic` HiC interactions must be provided as a `bedpe` file
- `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.

- `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.
- `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 `macs2` consensus peak using all other ChIP targets in combination
  - `"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 consensus/oracle peaks for each target and treatment group.

- `min_prop_peaks:` (Default: 0.1) Samples within a treatment group must contain more than this proportion of peaks which overlap peaks from the "best" sample within the relevant treatment group, i.e. the sample with the most number of peaks identified by `macs2 callpeak`. This is an inclusive threshold which will only discard clearly poor samples, which themselves are more likely to occur in tissue/organoid than in cell-line derived data.
- `min_prop_reps` (Default: 0.3) When forming **oracle** peaks within each *target and treatment group*, peaks must be represented in at least this proportion of samples. This defaults to 0.3 which would equate to 1 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.

4.2 colours.yml: Visualisation Settings

Defines all colour schemes for plots. Required YAML list elements are `qc`, `regions`, `direction` and `treat`. Optional colours can be provided for `features`.

4.3 params.yml: Additional Parameters

Default settings for annotations and enrichment testing

4.4 rmarkdown.yml: HTML Settings

Default settings for rmarkdown documents

Bibliography

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.

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.

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.

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.