# Rcade: R-based analysis of ChIP-seq And Differential Expression data

Shamith Samarajiwa

September 3, 2014

## Contents

## 1 Introduction

**This tutorial is based on the Rcade Bioconductor vignette created by Jonathan Cairns.**

*Rcade* is a bioconductor tool that analyses ChIP-seq data by coupling the ChIP-seq results to an existing Differential Expression (DE) analysis. *Rcade* enables inferring **direct targets** of a transcription factor (TF) - these targets should exhibit TF binding activity, and their expression levels should change in response to TF binding to gene associated regulatory elements.

The ChIP-seq analysis component of *Rcade* is performed through methods from the *baySeq* package, with respect to a user-defined universe of potential binding sites. This means that *Rcade* avoids the noise issues associated with peak-calling and focuses instead on robust quantification of binding activity. **Any** universe can be selected, and *Rcade* provides functionality to accommodate the common case where bins are defined relative to genomic annotation features.

In some situations, it may be appropriate to define the binding site universe based on a set of peak-calls from other data sets. However, it is **inappropriate** to use peak-calls from the same ChIP-seq data used in the Rcade analysis - such an analysis would be prone to **confirmation bias**.

# 2 Rcade Analysis

## 2.1 Install packages

The following packages are required. Use `install.packages()` or `biocLite()` to install them.

- *rgl*
- *biomaRt*
- *Rcade*

## 2.2 TF binding and gene transcription

Each gene is assumed to have some number of associated binding sites, each of which can be active or inactive as inferred from the ChIP-seq data. Additionally, every gene has one or more expression values, each of which is either DE or not DE under some perturbation - for example, after the knockdown or stimulation of a TF of interest. It is assumed that, conditional on a gene having both a ChIP-seq signal and a DE signature, the ChIP-seq and expression data associated with that gene are independent. All pairwise interactions between ChIP and DE data are considered for a given gene.

**Use Case:**

In this tutorial, we will use the example data provided in the *Rcade* package. Ideally you would use ChIP and expression data generated from the same experiment. In this example, data were obtained from two sources, each pertaining to the transcription factor **STAT1**. All of the experiments were performed with the HeLa cell line. The two data sets are:

1. Differential Expression data from ¡Array Express¿, under accession number E-GEOD-11299.
2. STAT1 ChIP-seq data from the Snyder lab, as part of the ENCODE consortium Input DCC accession numbers: wgEncodeEH000611 and wgEncodeEH000612 ChIP DCC accession number: wgEncodeEH000614

To keep file size down for the tutorial, all of these files have been truncated. Thus, they only contain data pertinent to a handful of selected genes.

The location of these data files will vary from system to system. To find the data directory on your computer, use the following code:

```
library("Rcade")
dir <- file.path(system.file("extdata", package="Rcade"), "STAT1")
dir
```

## 2.3 DE Analysis

The DE data (which was pre-analysed using the *limma* package) must contain the following fields:

*geneID* - gene IDs used to link DE results to genes. *logFC* - The log fold change associated with each gene. *B* - B values (log-odds).

```
DE <- read.csv(file.path(dir, "DE.csv"))
DElookup <- list(GeneID="ENSG", logFC="logFC", B="B","Genes.Location", "Symbol")
```

## 2.4 ChIP-seq analysis

These reads should have already undergone quality control and sequence-level pre-processing, such as any read trimming and adaptor removal. Moreover, they should have appropriate index files - for example, using the indexBam function in the package *Rbamtools*. Example .bam and .bai files are provided in the package:

```
dir(dir, pattern = ".bam")
targets <- read.csv(file.path(dir, "targets.csv"), as.is = TRUE)
targets
```

## 2.5 Annotation

In ChIP-seq analysis, Rcade performs its analysis based on the read counts in user-defined bin regions. These regions are specified with a GRanges object from the *GenomicRanges* package.

A common requirement is to define bins about genomic annotation features: Rcade provides functionality to generate a GRanges object via the defineBins() function. In this tutorial **a very reduced annotation** file is generated. Do not use this file for your own analysis. The *biomaRt* package can be used to generate custom annotation of genomic regions. The zone=c(-1500,1500) argument defines the zone of interest: it starts 1500bp 5' of each Transcription Start Site (TSS) (-1500), and ends 1500bp 3' of each TSS (+1500). This range was selected because preliminary analysis of read and peak distributions indicated that STAT1 binding are sites are enriched in this window and are mainly distributed around core promoter regions associated with the TSS.

```
anno <- read.csv(file.path(dir, "anno.csv"))
anno <- anno[order(anno$chromosome_name),]
colnames(anno) <- c("ENSG","chr","start","end","str")
ChIPannoZones <- defineBins(anno, zone=c(-1500, 1500), geneID="ENSG")
```

The object ChIPannoZones can now be used in the Rcade analysis.

## 2.6 Prior specification

By default, Rcade's prior belief is that each gene's DE and ChIP-seq statuses are independent. This is unlikely to be true in real data, as genes with ChIP-seq signal are more likely to be DE than other genes.

The prior values 0.05 and 0.005 were selected arbitrarily, before analysis. An advanced user might select priors in a less arbitrary manner for example, by looking at the overlap between ChIP-seq and DE in similar datasets. We do not go into further details of such an analysis here.

```
DE.prior = 0.01
prior.mode = "keepChIP"
prior = c("D|C" = 0.05, "D|notC" = 0.005)
```

## 2.7 Analysis

Use `RcadeAnalysis` function:

```
Rcade <- RcadeAnalysis(DE, ChIPannoZones, annoZoneGeneidName="ENSG",
ChIPtargets=targets, ChIPfileDir = dir, DE.prior=DE.prior,
prior.mode=prior.mode, prior=prior,DElookup=DElookup)
Rcade
```

The `Rcade` object stores information from the analysis. The DE, ChIP and Rcade analysis can be then stored in seperate objects.

```
xDE <- getDE(Rcade)
xChIP <- getChIP(Rcade)
xRcade <- getRcade(Rcade)
```

## 2.8 Plotting and QC

You can also generate plots, for example:

1. Principle Component Analysis on the counts with the plotPCA function

   ```
   plotPCA(Rcade)
   ```

2. The MM plot shows log-ratios from DE plotted against log-ratios from the ChIP-seq

   ```
   plotPCA(Rcade)
   plotMM(Rcade)
   ```

3. PlotBBB() plots log-odds values for ChIP-seq, DE and the combined ChIP-seq/DE analysis. You need to install the rgl package.

   ```
   library(rgl)
   plotBBB(Rcade)
   ```

# 3 Export Results

Results can then be exported into ”csv” text files. For example the top 2000 genes are exported by:

```
exportRcade(Rcade, directory="RcadeOutput",cutoffArg=2000)
```

more advanced export options are available:

```
exportRcade(Rcade, directory="RcadeOutput", cutoffMode="top", cutoffArg = 1000,
justGeneID=FALSE, removeDuplicates="beforeCutoff")
```

- cutoffMode = ”all” cutoff ignored, all results written to disk.
- cutoffMode = ”top”
  Take the top N genes, where N is specified by cutoffArg.
- cutoffMode = ”B”
  Take all genes with that satisfy B ¿ cutoffArg, where B is the log-odds.
- cutoffMode = ”FDR”

The expected false positive rate, FPR, and the expected false negative rate, FNR, are calculated using B values. The cutoff chosen is the one that maximizes the value of FPR + cutoffArg*FNR.

## 3.1 Exported analysis files

This function exports Rcade output to disk - specifically, it creates the following files:

- ChIP.csv
- ChIPonly.csv
- DEandChIP.csv
- DownChIP.csv
- Down.csv
- DownNoChIP.csv
- Nothing.csv
- UpChIP.csv
- Up.csv
- UpNoChIP.csv

Each file contains genes appropriate to its hypothesis, sorted by descending B value (i.e. ranked from most interesting to least interesting). For example, if you wanted the genes that display DE (either up or down) and also have ChIP signal present, you would look at the top rows of DEandChIP.csv. For genes that have a ChIP signal but explicitly show no DE, use ChIPonly.csv.

Rcade is an extremely useful package for downstream analysis of ChIP-seq data. It goes beyond peak or gene lists and enables identification of direct targets of transcription factors. These can be used in combination with Network Biology methods to unravel complexities of biological systems.