

# CummeRbund workflow

Loyal A. Goff

This document is a work in progress and will continually be updated as new features or analyses are integrated into the cummeRbund pipeline. This guide is being released as is, with the understanding that existing gaps will be completed in due time. Please bear with us as we work to expand this resource.

## Contents

### 1 Overview

### 2 Workflow Summary

### 3 Reading cuffdiff output

```
> library(cummeRbund)

> cuff <- readCufflinks(dir=system.file("extdata", package="cummeRbund"))
> cuff
```

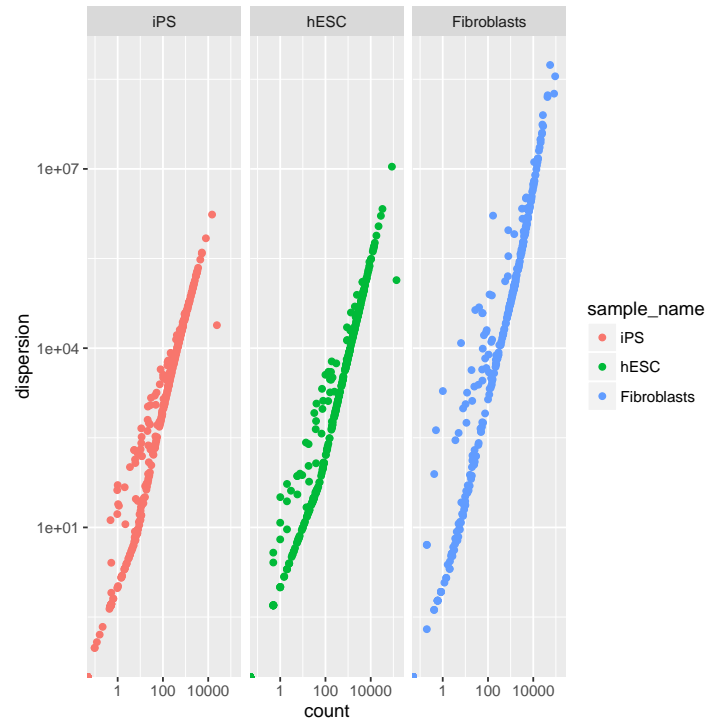
```
CuffSet instance with:
  3 samples
  400 genes
  1203 isoforms
  662 TSS
  906 CDS
  1062 promoters
  1986 splicing
  990 relCDS
```

This example uses the 'test' dataset that is included in the cummeRbund package. Importantly, if your current working directory contains the output from your cuffdiff analysis, simply calling `cuff<-readCufflinks()` will automatically find the default files and begin parsing your data.

## 4 Quality Assessment of data

### 4.1 Evaluating model fit

```
> d<-dispersionPlot(genes(cuff))  
> d
```

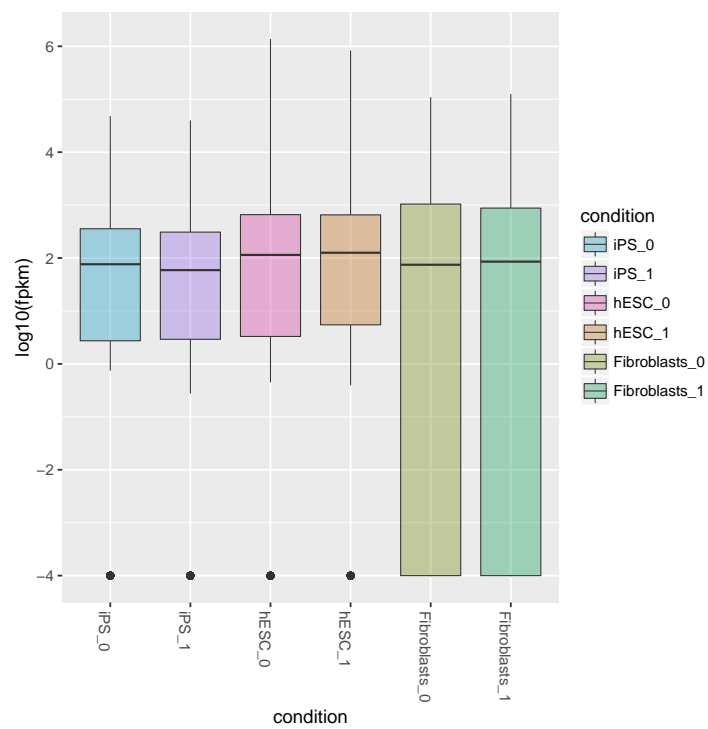


### 4.2 Identifying outlier replicates

```
> pBoxRep<-csBoxplot(genes(cuff),replicates=T)  
> pBoxRep
```

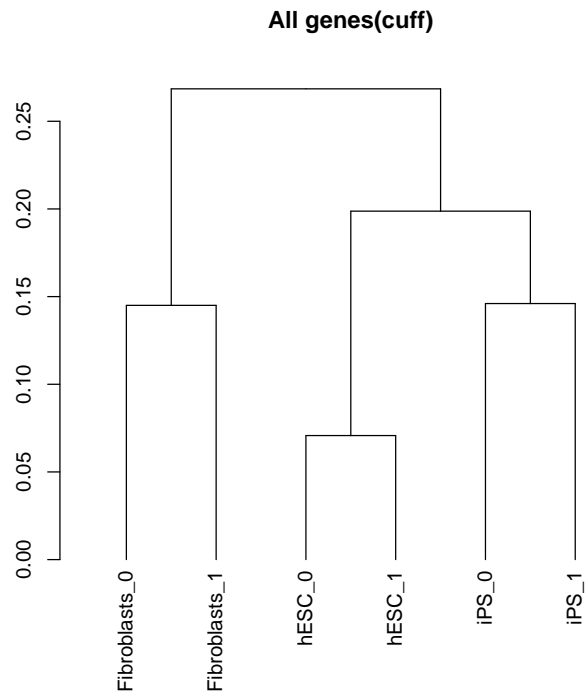
```
> pDendro<-csDendro(genes(cuff),replicates=T)  
> pDendro
```

'dendrogram' with 2 branches and 6 members total, at height 0.2685017



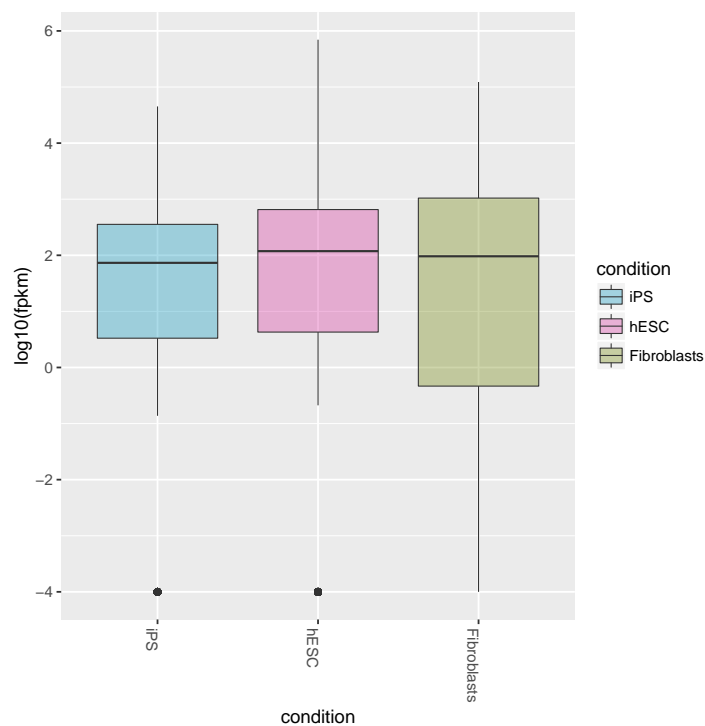
'dendrogram' with 2 branches and 6 members total, at height 0.2685017

'dendrogram' with 2 branches and 6 members total, at height 0.2685017



### 4.3 Determining relationships between conditions

```
> pBox<-csBoxplot(genes(cuff))
> pBox
```



## 5 Analysis of differential expression

### 5.1 Identifying differentially expressed genes

```
> sigGeneIds<-getSig(cuff,alpha=0.05,level="genes")
> head(sigGeneIds)

[1] "XLOC_000004" "XLOC_000005" "XLOC_000008" "XLOC_000009"
[5] "XLOC_000011" "XLOC_000013"

> length(sigGeneIds)

[1] 207

> hESCvsFibroblast.sigGeneIds<-getSig(cuff,"hESC","Fibroblasts",alpha=0.05,level="genes")
> head(hESCvsFibroblast.sigGeneIds)

[1] "XLOC_000005" "XLOC_000009" "XLOC_000013" "XLOC_000014"
[5] "XLOC_000018" "XLOC_000025"

> length(hESCvsFibroblast.sigGeneIds)

[1] 151
```

### 5.1.1 Creating significant gene sets

```
> sigGenes<-getGenes(cuff,sigGeneIds)
> sigGenes
```

CuffGeneSet instance for 207 genes

Slots:

annotation	
fpkm	
repFpkm	
diff	
count	
isoforms	CuffFeatureSet instance of size 717
TSS	CuffFeatureSet instance of size 399
CDS	CuffFeatureSet instance of size 577
promoters	CuffFeatureSet instance of size 207
splicing	CuffFeatureSet instance of size 399
relCDS	CuffFeatureSet instance of size 207

### 5.1.2 Visualization of significant gene sets

## 5.2 Identifying differentially expressed features

```
> sigGeneIds<-getSig(cuff,alpha=0.05,level="isoforms")
> head(sigGeneIds)
```

```
[1] "TCONS_00000007" "TCONS_00000008" "TCONS_00000021"
[4] "TCONS_00000022" "TCONS_00000033" "TCONS_00000041"
```

```
> length(sigGeneIds)
```

```
[1] 256
```

### 5.2.1 Creating significant feature sets

### 5.2.2 Visualization of significant feature sets

## 5.3 Analysis of individual genes

### 5.3.1 Subsetting individual genes

### 5.3.2 Creating a CuffGene object

```
> Pink1<-getGene(cuff,'PINK1')
> Pink1
```

CuffGene instance for gene XLOC\_000172

Short name: PINK1

Slots:

annotation	
features	
fpkm	
repFpkm	
diff	
count	
isoforms	CuffFeature instance of size 2
TSS	CuffFeature instance of size 2
CDS	CuffFeature instance of size 2

### 5.3.3 Visualization of individual genes

## 6 Data Exploration

### 6.1 Identifying patterns of gene expression

Partitioning around medoids as e.g.

### 6.2 Similarity analysis

*findSimilar*

### 6.3 Specificity analysis

*csSpecificity*

## 7 Analysis of differential splicing

## 8 Analysis of differential promoter usage

## 9 Linking with additional analyses

### 9.1 Gene Ontology Analysis (clusterProfiler)

In this particular example, the reference .gtf file that was used has the official gene symbol (HUGO) stored in the *gene name* attribute field. This value is carried over by cuffdiff and cummeRbund into the *gene\_short\_name* field for each feature. We will exploit this ID and use it to map genes to their *Entrez ID* to be able to use the very nice clusterProfiler package.

The first step is to translate

### 9.2 Gene Set Enrichment Analysis (Preranked)

We will generate .rnk and .gmx files for use in the GSEA Preranked analysis.

- 9.3 Principal component analysis
- 10 Visualization Tips/Tricks
  - 10.1 Faceting
- 11 Session Info