HT collections of genomic features: TUs, TSSs, TTSs

Claire Rioualen

2022-03-26

Table of Contents

library(DT)  
library(gridExtra)  
library(readxl)  
library(tidyverse)  
  
## Local installation  
devtools::load\_all("~/Desktop/Git/EcoliGenes")  
  
concat\_uniq <- function(x, sep = ","){  
 paste0(unique(na.omit(x)), collapse = sep)  
}  
  
## wrapper function to remove commented rows from excel sheets  
read\_excel\_cm <- function(file, sheet, comment.char = "#"){  
 tab <- read\_excel(file, sheet, col\_names = F)  
 skip\_rows <- grep(paste0("^", comment.char), unlist(c(tab[,1])))  
 if(length(skip\_rows) > 0) tab <- read\_excel(file, sheet, skip = max(skip\_rows))  
 return(tab)  
}

# RegulonDB-HT collections

*In this work, we offer facilitated access to HT collections. Each collection comprises all of the curated datasets generated using a given technology, and resulting in a specific type of object. A dataset is conceived as a set of data and its metadata. Metadata gathers the properties that enable the identification of an experiment, such as growth conditions (culture medium, medium supplements, aeration, temperature, pH, agitation, growth phase, optical density), genome version, features associated with the publications (author list, year of publication, PMID), as well as reported database identifiers, plus any additional pertinent information.* (Tierrafría, Rioualen et al., submitted)

In this report, I describe the generation of the collections of TUs, transcription start sites and termination sites HT datasets.

## Transcription units

### Format

TUs are collected from a variety of sources and processed to produce uniform datasets in a bed-like format, with the following fields:

* **chromosome:** NC\_000913.3
* **start:** Left genomic position
* **stop:** Right genomic position
* **id:** Unique TU ID
* **length:** TU length
* **strand:** TU strand reported by authors
* **gene\_number:** number of genes entirely contained in TU
* **genes:** bnumbers of genes entirely contained in TU
* **pseudo:** 1 if TU contains pseudo genes, else 0
* **phantom:** 1 if TU contains phantom genes, else 0

NB: Some of this information may differ from the originally published data. For example, we chose to report only the genes that are entirely contained in the TUs, while some authors chose to report all genes intersectting with the TU coordinates. When needed, gene names were actualized to the latest annotations available, and TU coordinates were actualized to the genome version NC\_000913.3.

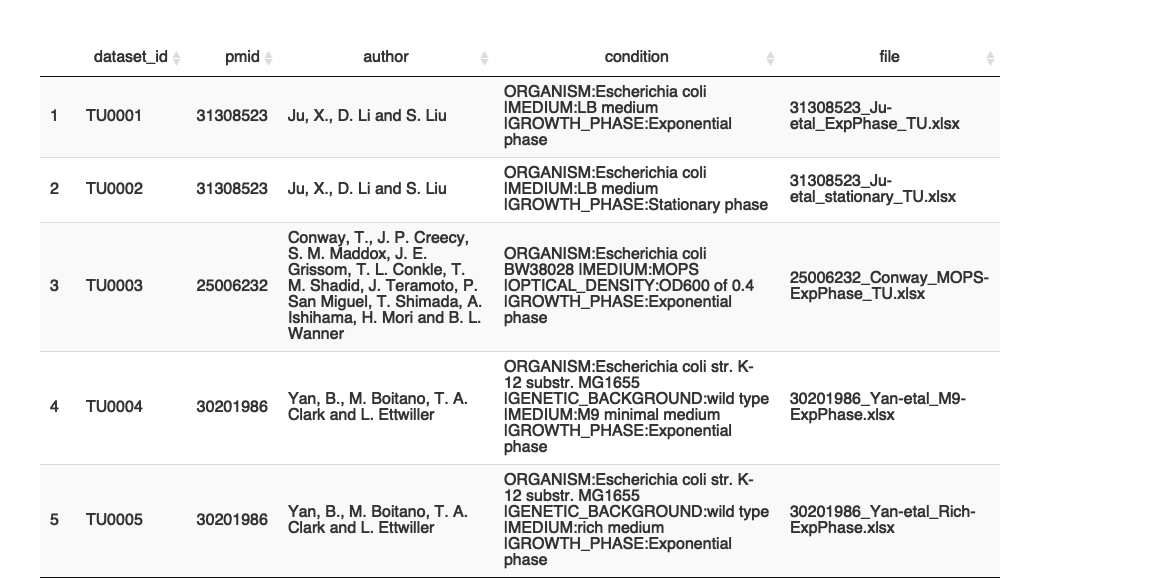
### Metadata

Each collection is thoroughly described in its corresponding metadata table, generated through curation of literature and database exploration.

ht\_tu\_dir <- paste0(ht\_collection\_dir, "/collection\_TU")  
  
## Read metadata table, filter out datasets that don't have a condition associated  
ht\_tu\_metadata <- read\_excel(paste0(ht\_tu\_dir, "/DatasetCollection-TU-HT.xlsx"), sheet = 1, skip = 1) %>%  
 dplyr::rename(dataset\_id = "Dataset ID",  
 pmid = "PMID",  
 author = "Authors",  
 condition = "Growth Conditions Experimental",  
 file = "Dataset File Name") %>%  
 dplyr::mutate(filepath = paste0(ht\_tu\_dir, "/author\_files/", file)) %>%  
 dplyr::mutate(first\_author = str\_split(author, ",")[[1]][1]) %>%  
 dplyr::select(dataset\_id, pmid, author, first\_author, condition, file, filepath) %>%  
 dplyr::filter(!is.na(condition))  
  
## Make dataset list from table  
ht\_tu\_dataset\_list <- setNames(split(ht\_tu\_metadata, seq(nrow(ht\_tu\_metadata))), ht\_tu\_metadata$dataset\_id)

These are the TU datasets available:

## Display datasets  
DT1 <- DT::datatable(ht\_tu\_metadata %>% dplyr::select(-filepath), options = list(dom = '', pageLength = 20, autoWidth = TRUE))  
DT1



### Datasets per author

This part is customized depending on the originally published files. Since those files are formatted each in their own way, this part semi manual and should be modified carefully.

It generates one table per dataset, with the following columns: start, stop, strand.

#### Conway et al., 2014

Reference article: [Conway et al., 2014](https://doi.org/10.1128/mbio.01442-14)

Conditions:

* Exponential phase

Notes:

* Genome coordinates were converted from version U00096.2 to version U00096.3 using the library [EcoliGenes](https://github.com/rioualen/EcoliGenes)

TU0003 <- read\_excel(ht\_tu\_dataset\_list$TU0003$filepath, sheet = 1, skip = 1) %>%  
 dplyr::select(TU, STRAND) %>%  
 dplyr::rename(strand = STRAND) %>%  
 tidyr::separate(TU, c("left", "right"), sep = ":") %>%  
 dplyr::distinct() %>%  
 dplyr::mutate(left = as.numeric(gsub("^[A-Z]{1,2}-{1}", "", left)),  
 right = as.numeric(gsub("^[A-Z]{1,2}-{1}", "", right))) %>%  
 dplyr::mutate(start = EcoliGenes::convert\_coords(left),  
 stop = EcoliGenes::convert\_coords(right)) %>%  
 dplyr::select(start, stop, strand)

#### Ju et al., 2019

Reference article: [Ju et al., 2019](https://doi.org/10.1038/s41564-019-0500-z)

Conditions:

* exponential phase
* stationary phase

TU0001 <- read\_excel(ht\_tu\_dataset\_list$TU0001$filepath, skip = 1) %>%  
 dplyr::rename(start = Boundary\_of\_transcript\_unit...1,   
 stop = Boundary\_of\_transcript\_unit...2,   
 strand = Direction\_of\_transcript\_unit) %>%  
 dplyr::select(-Number\_of\_covered\_annotated\_genes, -Name\_of\_covered\_Genes, -Length)  
  
TU0002 <- read\_excel(ht\_tu\_dataset\_list$TU0002$filepath, skip = 1) %>%  
 dplyr::rename(start = Boundary\_of\_transcript\_unit...1,   
 stop = Boundary\_of\_transcript\_unit...2,   
 strand = Direction\_of\_transcript\_unit) %>%  
 dplyr::select(start, stop, strand)

#### Yan et al., 2018

Reference article: [Yan et al., 2018](10.1038/s41467-018-05997-6)

Conditions:

* Rich (rich growth medium)
* M9 (minimal growth medium)

Notes:

* Two types of TU termination are reported for each condition: detected TTS or longest read end. They are grouped in 2 files according to growth conditions

TU0004 <- read\_excel(ht\_tu\_dataset\_list$TU0004$filepath, skip = 0) %>%  
 dplyr::rename(start = TU\_left\_position,   
 stop = TU\_right\_position,   
 strand = TU\_strand) %>%  
 dplyr::select(start, stop, strand)  
  
TU0005 <- read\_excel(ht\_tu\_dataset\_list$TU0005$filepath, skip = 0) %>%  
 dplyr::rename(start = TU\_left\_position,   
 stop = TU\_right\_position,   
 strand = TU\_strand) %>%  
 dplyr::select(start, stop, strand)

### Uniformization

This part assumes that the specificities of each individual dataset were dealt with previously. What it does:

* Generate unique IDs for TUs
* Generate additional columns (TU length, TU genes, TU gene number, flag columns for pseudo genes and phantom genes)
* Write one file per dataset

Notes:

* Only the genes that are entirely contained in the TUs are taken into account, regardless of what was reported by the authors in the original datasets.
* The step that consists in getting bnumbers from genes that are contained in each TU is a little slow.

## Write datasets to files  
tu\_datasets\_list <- list()  
for(ds\_id in names(ht\_tu\_dataset\_list)) {  
 df <- get(ds\_id) %>%  
 dplyr::arrange(start) %>%  
 dplyr::mutate(id = paste0("TU\_", ds\_id, "\_", dplyr::row\_number()),  
 chromosome = "NC\_000913.3") %>%  
 dplyr::rowwise() %>%  
 dplyr::mutate(length = stop - start + 1,  
 genes = EcoliGenes::what\_genes(start, stop, strand)) %>%  
 tidyr::separate\_rows(genes, sep = ",") %>%  
 dplyr::mutate(pseudo = EcoliGenes::is\_pseudogene(genes),  
 phantom = EcoliGenes::is\_phantomgene(genes)) %>%  
 dplyr::group\_by(id) %>%  
 dplyr::summarise(across(where(is.character), concat\_uniq),  
 across(where(is.numeric), min),  
 gene\_number = ifelse(genes == "", 0, n())) %>%  
 dplyr::arrange(start)   
  
 write.table(df %>% dplyr::select(chromosome, start, stop, id, length, strand, gene\_number, genes, pseudo, phantom),  
 file = paste0(ht\_tu\_dir, "/uniform\_files/", ds\_id, ".tsv"),  
 sep = "\t", quote = F, col.names = T, row.names = F)  
 assign(ds\_id, df)  
 tu\_datasets\_list[[ds\_id]] <- df  
}

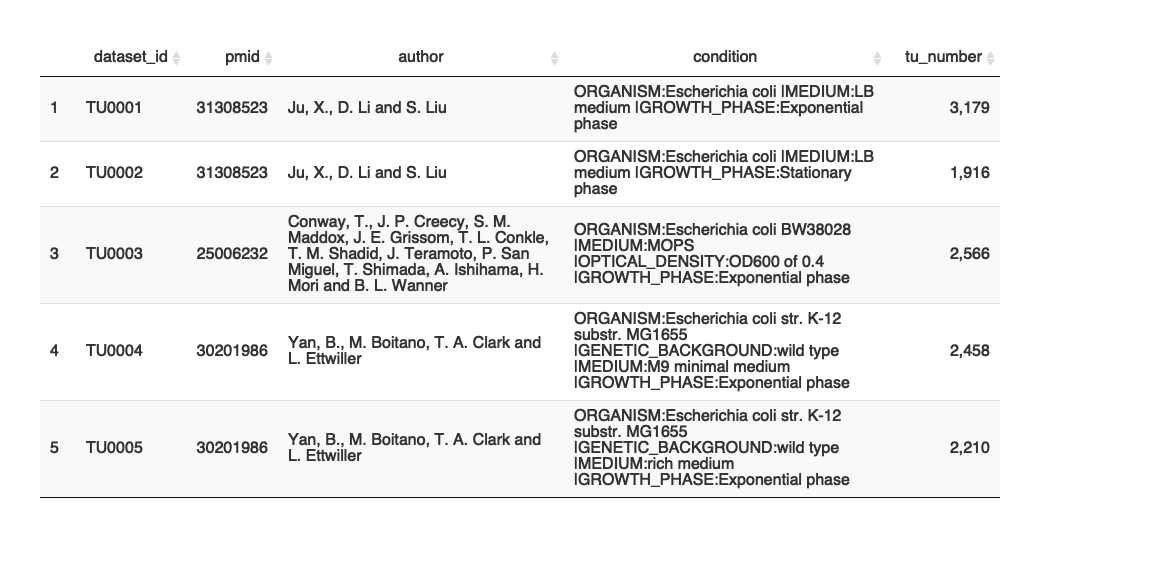
* Write a summary file for TU datasets

ht\_tu\_stats <- ht\_tu\_metadata %>%  
 dplyr::rowwise() %>%  
 dplyr::mutate(tu\_number = nrow(get(dataset\_id))) %>%  
 dplyr::select(dataset\_id, pmid, author, condition, tu\_number)  
  
write.table(ht\_tu\_stats, file = paste0(ht\_tu\_dir, "/TU\_datasets.tsv"), sep = "\t", quote = F, col.names = T, row.names = F)

### Results

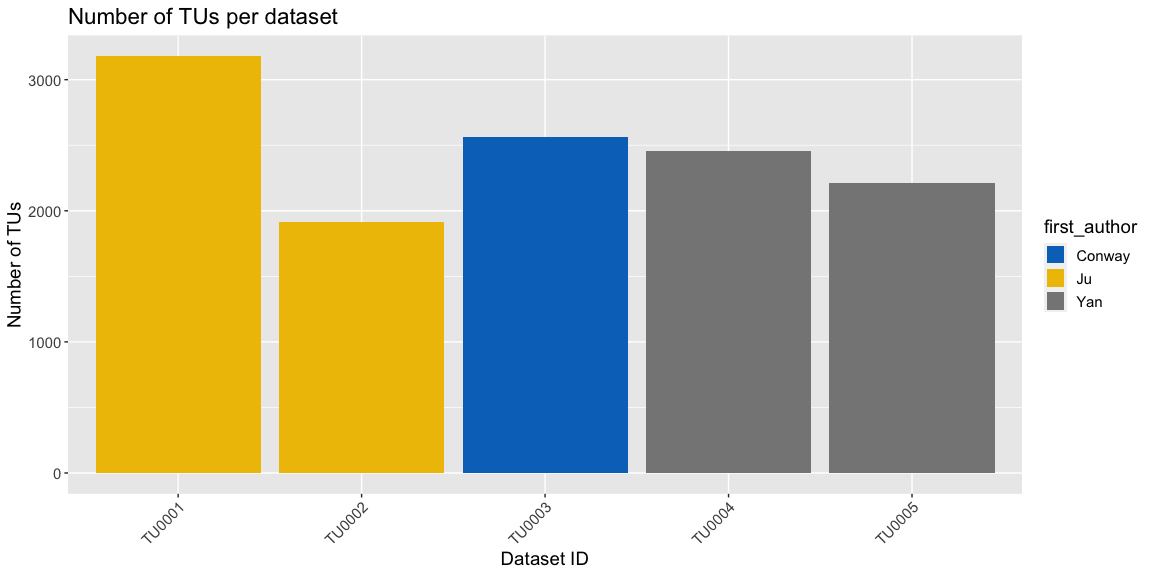
Summary of the TU datasets:

DT2 <- DT::datatable(ht\_tu\_stats, options = list(dom = '', pageLength = 20)) %>% DT::formatRound('tu\_number', digits=0)  
DT2



Number of TUs per dataset:

ht\_tu\_stats\_legends <- ht\_tu\_stats %>%  
 dplyr::mutate(first\_author = str\_split(author, ",")[[1]][1])  
  
ggplot(ht\_tu\_stats\_legends, aes(x = dataset\_id, y = tu\_number, fill = first\_author)) +  
 geom\_col(size = 2) +  
 theme(axis.text.x = element\_text(angle = 45, hjust=1), text = element\_text(size = 14)) +  
 ylab("Number of TUs") +  
 xlab("Dataset ID") +  
 labs(title = "Number of TUs per dataset") +  
 ggsci::scale\_fill\_jco()



## Transcription start sites (TSS)

### Format

TSSs are collected from a variety of sources and processed to produce uniform bed-like dataset files with the following fields:

* **chromosome:** NC\_000913.3
* **start:** Left genomic position. If not reported, will be the same as pos\_1
* **stop:** Right genomic position. If not reported, will be the same as pos\_1
* **id:** Unique TSS ID
* **pos\_1:** TSS +1 position reported by authors
* **strand:** TSS strand reported by authors

### Metadata

ht\_tss\_dir <- paste0(ht\_collection\_dir, "/collection\_TSS")  
  
## Read metadata table  
ht\_tss\_metadata <- read\_excel(paste0(ht\_tss\_dir, "/DatasetCollection-TSS-HT.xlsx"), sheet = 1, skip = 1) %>%  
 dplyr::rename(dataset\_id = "Dataset ID",  
 pmid = "PMID",  
 author = "Authors",  
 condition = "Growth Conditions Experimental",  
 file = "Dataset File Name") %>%  
 dplyr::mutate(filepath = paste0(ht\_tss\_dir, "/author\_files/", file)) %>%  
 dplyr::mutate(first\_author = str\_split(author, ",")[[1]][1]) %>%  
 dplyr::select(dataset\_id, pmid, author, first\_author, condition, file, filepath)   
#dplyr::filter(!is.na(condition))  
   
## Make list from table  
ht\_tss\_dataset\_list <- setNames(split(ht\_tss\_metadata, seq(nrow(ht\_tss\_metadata))), ht\_tss\_metadata$dataset\_id)

Datasets:

## Display datasets  
DT3 <- DT::datatable(ht\_tss\_metadata %>% dplyr::select(-filepath, -author), options = list(dom = '', pageLength = 20, autoWidth = TRUE, columnDefs = list(list(width = '200px', targets = c(4)))))  
DT3



### Datasets per author

#### Conway et al., 2014

Reference article: [Conway et al., 2014](https://doi.org/10.1128/mbio.01442-14)

Conditions: NA

Notes:

* Genome coordinates were converted from version U00096.2 to version U00096.3 using the library [EcoliGenes](https://github.com/rioualen/EcoliGenes).

DS0006 <- read\_excel(ht\_tss\_dataset\_list$DS0006$filepath, sheet = 1, skip = 4) %>%  
 dplyr::mutate(pos\_1 = gsub("^[PSIAS]{1,2}-", "", Promoter), strand = STRAND) %>%  
 dplyr::select(strand, pos\_1) %>%  
 dplyr::distinct() %>%  
 dplyr::mutate(pos\_1 = EcoliGenes::convert\_coords(pos\_1)) %>%  
 dplyr::mutate(start = pos\_1, stop = pos\_1)

#### Ju et al., 2019

Reference article: [Ju et al., 2019](https://doi.org/10.1038/s41564-019-0500-z)

Conditions:

* exponential phase
* stationary phase

DS0004 <- read\_excel\_cm(ht\_tss\_dataset\_list$DS0004$filepath, sheet = 1) %>%  
 dplyr::rename(pos\_1 = TSS\_site,   
 strand = TSS\_direction) %>%  
 dplyr::mutate(start = pos\_1, stop = pos\_1) %>%  
 dplyr::select(start, stop, strand, pos\_1) %>%  
 dplyr::distinct()  
  
DS0005 <- read\_excel\_cm(ht\_tss\_dataset\_list$DS0005$filepath, sheet = 1) %>%  
 dplyr::rename(pos\_1 = TSS\_site,   
 strand = TSS\_direction) %>%  
 dplyr::mutate(start = pos\_1, stop = pos\_1) %>%  
 dplyr::select(start, stop, strand, pos\_1) %>%  
 dplyr::distinct()

#### Mendoza-Vargas et al., 2009; Salgado et al., 2013

Reference articles:

1. [Mendoza-Vargas et al., 2009](https://doi.org/10.1371/journal.pone.0007526)
2. [Salgado et al., 2013](https://pubmed.ncbi.nlm.nih.gov/23203884/)

Conditions/methodology:

* 454 data (1)
* RACE data (1)
* 5 tri or monophosphate enrichment (1, 2)

Notes:

* Files were downloaded on 2021/03/19 [here](http://regulondb.ccg.unam.mx/highthroughputdatasetssearch?term=all)
* Genome coordinates were converted from version U00096.2 to version U00096.3 using the library [EcoliGenes](https://github.com/rioualen/EcoliGenes)
* 3 datasets files formatted differently:
  + One file has TSS left and right positions, as well as “max frequency” position.It also has “orientation” information.
  + Two files have gene’s left and right coordinates, and relative position of the TSS to the gene (unique position is calculated depending on strand). These files don’t have relative orientation information

#--------------------------------------------------------------------------------------  
DS0009 <- read\_excel\_cm(ht\_tss\_dataset\_list$DS0009$filepath, sheet = 1, comment.char = "#") %>%  
 dplyr::rename(strand = Orientation,   
 left = `TSS Absolute genome left position`,   
 right = `TSS Absolute genome right position`,   
 relative\_pos = `Relative distance to gene`) %>%  
 dplyr::mutate(tss\_pos = ifelse(strand == "forward", left + relative\_pos, ifelse(strand == "reverse", right - relative\_pos, NA))) %>%  
 dplyr::mutate(pos\_1 = EcoliGenes::convert\_coords(tss\_pos)) %>%  
 dplyr::mutate(start = pos\_1,   
 stop = pos\_1,  
 strand = ifelse(strand == "reverse", "-", ifelse(strand == "forward", "+", NA))) %>%  
 dplyr::select(start, stop, strand, pos\_1) %>%  
 dplyr::distinct()   
  
DS0010 <- read\_excel\_cm(ht\_tss\_dataset\_list$DS0010$filepath, sheet = 1, comment.char = "#") %>%  
 dplyr::rename(strand = Orientation,   
 left = `TSS Absolute genome left position`,   
 right = `TSS Absolute genome right position`,   
 relative\_pos = `Relative orientation to gene`) %>%  
 dplyr::mutate(tss\_pos = ifelse(strand == "forward", left + relative\_pos, ifelse(strand == "reverse", right - relative\_pos, NA))) %>%  
 dplyr::mutate(pos\_1 = EcoliGenes::convert\_coords(tss\_pos)) %>%  
 dplyr::mutate(start = pos\_1,   
 stop = pos\_1,  
 strand = ifelse(strand == "reverse", "-", ifelse(strand == "forward", "+", NA))) %>%  
 dplyr::select(start, stop, strand, pos\_1) %>%  
 dplyr::distinct()   
  
#--------------------------------------------------------------------------------------  
DS0011 <- read\_excel\_cm(ht\_tss\_dataset\_list$DS0011$filepath, sheet = 1, comment.char = "#") %>%  
 dplyr::rename(strand = Orientation,   
 left = `TSS Absolute genome left position`,   
 right = `TSS Absolute genome right position`,   
 tss\_pos = `TSS Max Frequency`) %>%  
 dplyr::mutate(pos\_1 = EcoliGenes::convert\_coords(tss\_pos)) %>%  
 dplyr::mutate(start = EcoliGenes::convert\_coords(left)) %>%  
 dplyr::mutate(stop = EcoliGenes::convert\_coords(right)) %>%  
 dplyr::mutate(strand = ifelse(strand == "reverse", "-", ifelse(strand == "forward", "+", NA))) %>%  
 dplyr::select(start, stop, strand, pos\_1) %>%  
 dplyr::distinct()

#### Cho et al., 2014

Reference article: [Cho et al., 2014](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3923258/)

Conditions:

* Exponential phase (from [2012 paper](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3415461/)\*)
* Glutamine as source of nitrogen
* Heat shock
* Stationary phase

Notes:

* All TSSs come in the supplementary table 6 from the 2014 article, and are separated in 4 distinct datasets
* Genome coordinates were converted from version U00096.2 to version U00096.3 using the library [EcoliGenes](https://github.com/rioualen/EcoliGenes)

##---------------------------------------------------------------------------------  
Palsson\_all <- read\_excel(ht\_tss\_dataset\_list$DS0013$filepath, sheet = 1) %>%  
 dplyr::rename(strand = Strand, condition = Conditions, tss\_pos = `TSS position`) %>%  
 dplyr::mutate(pos\_1 = EcoliGenes::convert\_coords(tss\_pos)) %>%  
 tidyr::separate\_rows(condition, sep = "") %>%  
 dplyr::mutate\_all(dplyr::na\_if,"") %>%  
 dplyr::filter(!is.na(condition)) %>%  
 dplyr::mutate(start = pos\_1, stop = pos\_1) %>%  
 dplyr::select(start, stop, strand, pos\_1, condition) %>%  
 dplyr::distinct()   
  
##---------------------------------------------------------------------------------  
DS0013 <- Palsson\_all %>% dplyr::filter(condition == "E") %>% dplyr::select(-condition)  
  
DS0014 <- Palsson\_all %>% dplyr::filter(condition == "G") %>% dplyr::select(-condition)  
  
DS0015 <- Palsson\_all %>% dplyr::filter(condition == "H") %>% dplyr::select(-condition)  
  
DS0016 <- Palsson\_all %>% dplyr::filter(condition == "S") %>% dplyr::select(-condition)

#### Thomason et al., 2014

Reference article: [Thomason et al., 2014](https://doi.org/10.1128/jb.02096-14)

Conditions:

* LB growth medium, OD = 2.0 (stationary phase)
* Minimum growth medium, OD = 0.4 (exponential phase)
* LB growth medium, OD = 0.4 (exponential phase)

Notes:

* Genome coordinates were converted from version U00096.2 to version U00096.3 using the library [EcoliGenes](https://github.com/rioualen/EcoliGenes)

DS0001 <- read\_excel\_cm(ht\_tss\_dataset\_list$DS0001$filepath, sheet = 1, comment.char = "#") %>%  
 dplyr::filter(detected == 1) %>%  
 dplyr::select(Strand, Pos) %>%  
 dplyr::distinct() %>%  
 dplyr::mutate(pos\_1 = EcoliGenes::convert\_coords(Pos)) %>%  
 dplyr::mutate(start = pos\_1, stop = pos\_1, strand = Strand) %>%  
 dplyr::select(start, stop, strand, pos\_1)  
  
DS0002 <- read\_excel\_cm(ht\_tss\_dataset\_list$DS0002$filepath, sheet = 1, comment.char = "#") %>%  
 dplyr::filter(detected == 1) %>%  
 dplyr::select(Strand, Pos) %>%  
 dplyr::distinct() %>%  
 dplyr::mutate(pos\_1 = EcoliGenes::convert\_coords(Pos)) %>%  
 dplyr::mutate(start = pos\_1, stop = pos\_1, strand = Strand) %>%  
 dplyr::select(start, stop, strand, pos\_1)  
  
DS0003 <- read\_excel\_cm(ht\_tss\_dataset\_list$DS0003$filepath, sheet = 1, comment.char = "#") %>%  
 dplyr::filter(detected == 1) %>%  
 dplyr::select(Strand, Pos) %>%  
 dplyr::distinct() %>%  
 dplyr::mutate(pos\_1 = EcoliGenes::convert\_coords(Pos)) %>%  
 dplyr::mutate(start = pos\_1, stop = pos\_1, strand = Strand) %>%  
 dplyr::select(start, stop, strand, pos\_1)

#### Wade lab

Data not published, refer to [this article](https://doi.org/10.1007/978-1-62703-730-3_1) for details on experimental methods.

Conditions:

* M9 + 0.2% glycerol, cells grown with shaking at 30°C

Notes:

* TSSs queried from private database on 2021-09-21

DS0012 <- read\_excel(ht\_tss\_dataset\_list$DS0012$filepath, sheet = 1) %>%  
 dplyr::mutate(pos\_1 = start) %>%  
 dplyr::select(start, stop, strand, pos\_1) %>%  
 dplyr::distinct()

#### Yan et al., 2018

Reference paper: [Yan et al., 2018](10.1038/s41467-018-05997-6)

Conditions:

* M9
* Rich

Notes:

* HT-inferred TSSs from PacBio long read data

DS0007 <- read\_excel(ht\_tss\_dataset\_list$DS0007$filepath, sheet = 1) %>%  
 dplyr::rename(start = TSS\_left\_position,   
 stop = TSS\_right\_position,   
 pos\_1 = TSS\_site,  
 strand = TSS\_strand) %>%  
 dplyr::select(start, stop, strand, pos\_1) %>%  
 dplyr::distinct()  
  
DS0008 <- read\_excel(ht\_tss\_dataset\_list$DS0008$filepath, sheet = 1) %>%  
 dplyr::rename(start = TSS\_left\_position,   
 stop = TSS\_right\_position,   
 pos\_1 = TSS\_site,  
 strand = TSS\_strand) %>%  
 dplyr::select(start, stop, strand, pos\_1) %>%  
 dplyr::distinct()

### Uniformization

* Generate unique IDs for TSSs
* Write one file per dataset

tss\_datasets\_list <- list()  
  
## Uniformize format, add TSS ID column, write file  
for(ds\_id in names(ht\_tss\_dataset\_list)) {  
 df <- get(ds\_id) %>% ##### OJO con estos que no se actualizan ## ?  
   
 dplyr::arrange(start) %>%  
 dplyr::mutate(id = paste0("TSS\_", ds\_id, "\_", dplyr::row\_number())) %>%  
 dplyr::mutate(chromosome = "NC\_000913.3")  
  
 write.table(df %>% dplyr::select(chromosome, start, stop, id, pos\_1, strand),  
 file = paste0(ht\_tss\_dir, "/uniform\_files/", ds\_id, ".tsv"),  
 sep = "\t", quote = F, col.names = T, row.names = F)  
   
 tss\_datasets\_list[[ds\_id]] <- df  
}

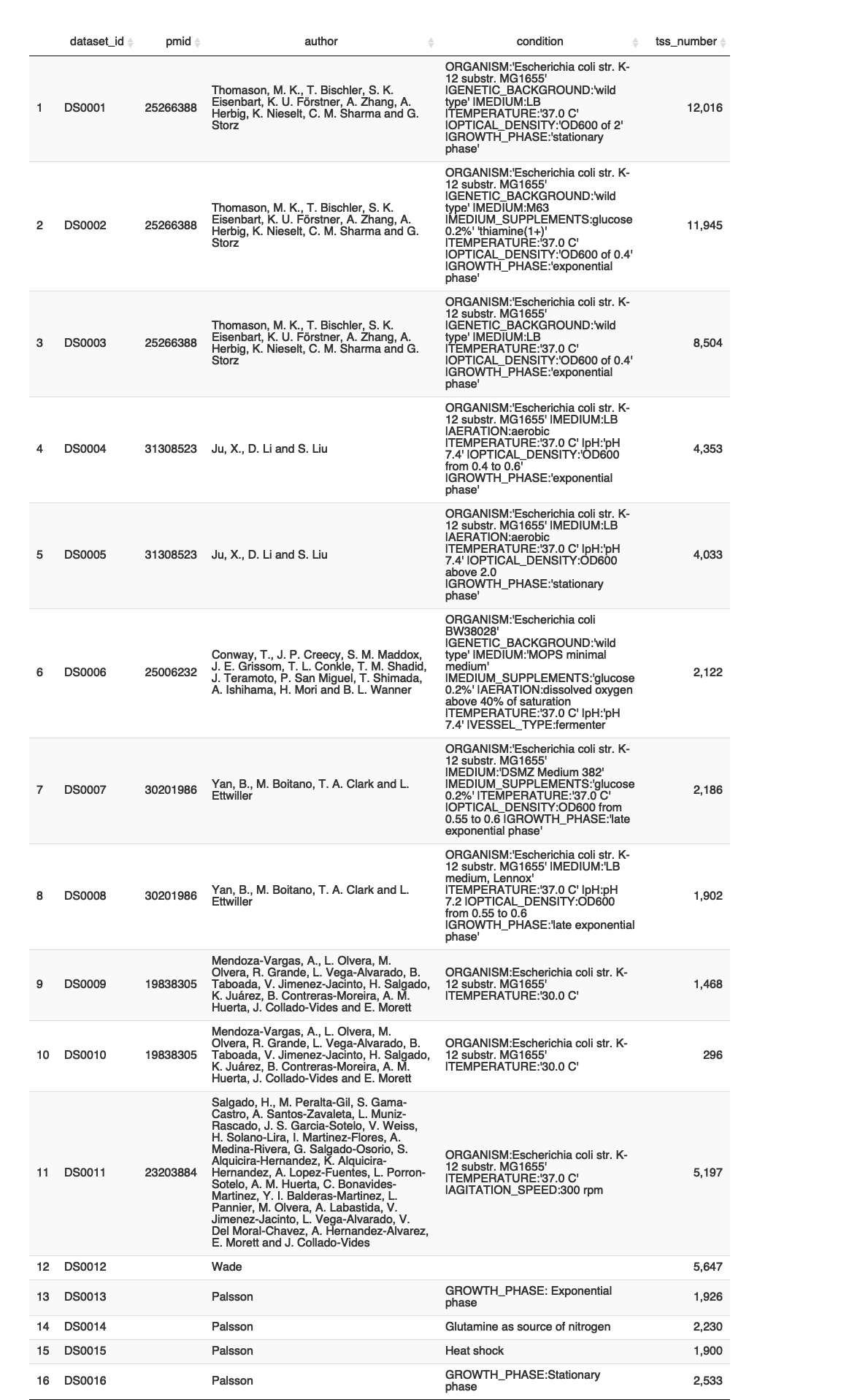
* Write a summary file for TSS datasets

ht\_tss\_stats <- ht\_tss\_metadata %>%  
 dplyr::rowwise() %>%  
 dplyr::mutate(tss\_number = nrow(get(dataset\_id))) %>%  
 dplyr::select(dataset\_id, pmid, author, condition, tss\_number)  
  
write.table(ht\_tss\_stats, file = paste0(ht\_tss\_dir, "/TSS\_datasets.tsv"), sep = "\t", quote = F, col.names = T, row.names = F)

### Results

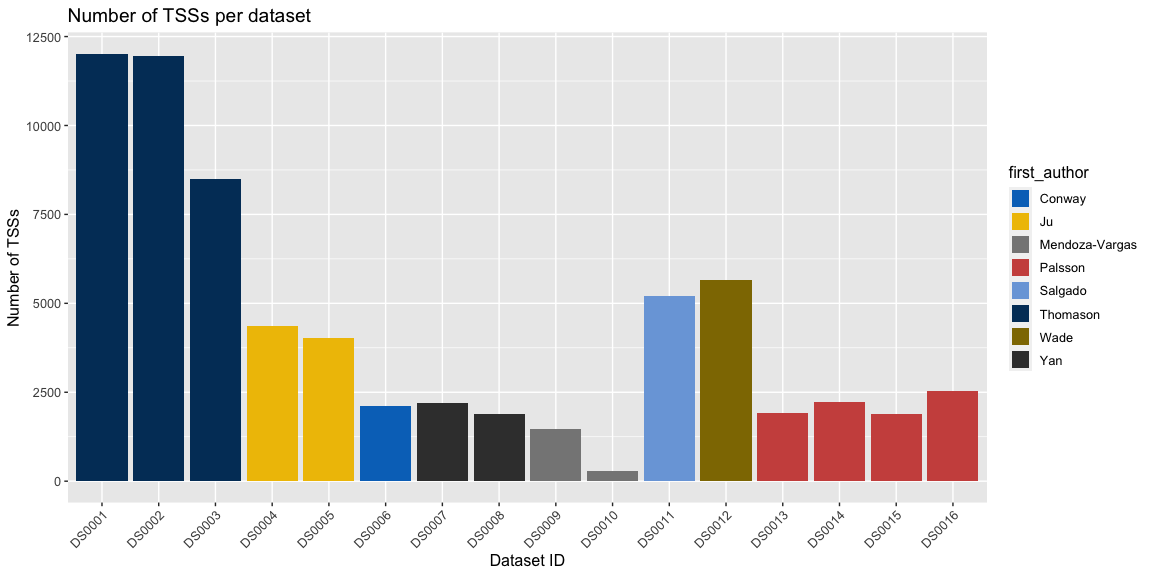
Summary of datasets

DT4 <- DT::datatable(ht\_tss\_stats, options = list(dom = '', pageLength = 20)) %>% DT::formatRound('tss\_number', digits=0)  
DT4



Distribution

ht\_tss\_stats\_legends <- ht\_tss\_stats %>%  
 dplyr::mutate(first\_author = str\_split(author, ",")[[1]][1])  
  
ggplot(ht\_tss\_stats\_legends, aes(x = dataset\_id, y = tss\_number, fill = first\_author)) +  
 geom\_col(size = 2) +  
 theme(axis.text.x = element\_text(angle = 45, hjust=1), text = element\_text(size = 12)) +  
 ylab("Number of TSSs") +  
 xlab("Dataset ID") +  
 labs(title = "Number of TSSs per dataset") +  
 ggsci::scale\_fill\_jco()



## Transcription termination sites (TTS)

### Format

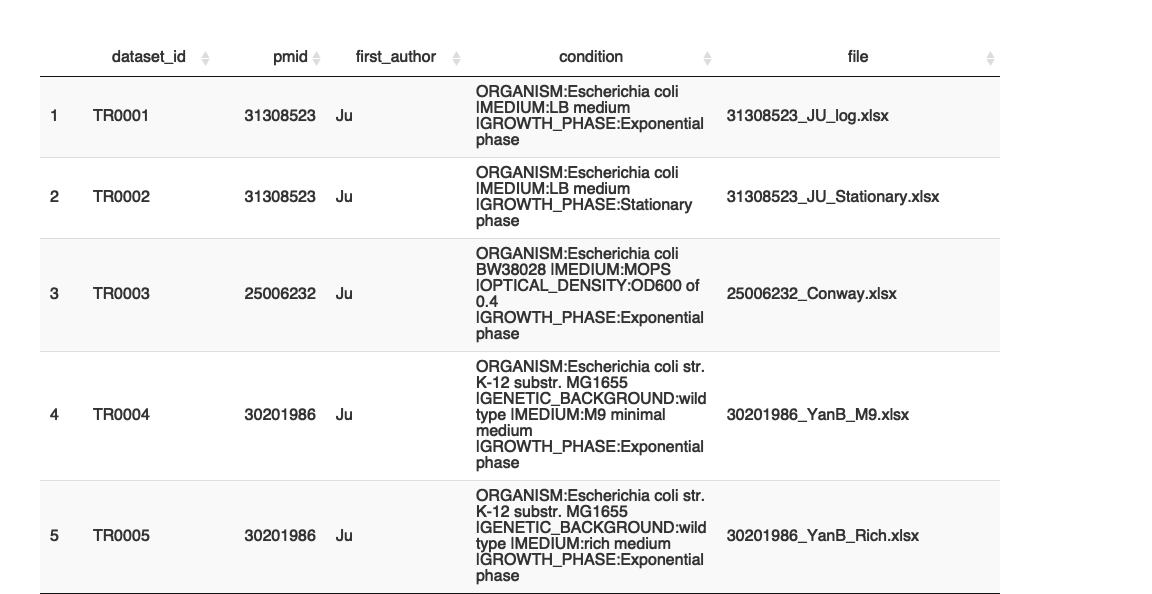
TTSs are collected from a variety of sources and processed to produce uniform bed-like dataset files with the following fields:

* **chromosome:** NC\_000913.3
* **start:** Left genomic position. If not reported, will be the same as term\_pos
* **stop:** Right genomic position. If not reported, will be the same as term\_pos
* **id:** Unique TSS ID
* **term\_pos:** Terminal position reported by authors
* **strand:** TTS strand reported by authors

### Metadata

ht\_tts\_dir <- paste0(ht\_collection\_dir, "/collection\_TTS")  
  
## Read metadata table, filter out datasets that don't have a condition associated  
ht\_tts\_metadata <- read\_excel(paste0(ht\_tts\_dir, "/DatasetCollection-TTS-HT.xlsx"), sheet = 1, skip = 1) %>%  
 dplyr::rename(dataset\_id = "Dataset ID",  
 pmid = "PMID",  
 author = "Authors",  
 condition = "Growth Conditions Experimental",  
 file = "Dataset File Name") %>%  
 dplyr::mutate(filepath = paste0(ht\_tts\_dir, "/author\_files/", file)) %>%  
 dplyr::mutate(first\_author = str\_split(author, ",")[[1]][1]) %>%  
 dplyr::select(dataset\_id, pmid, author, first\_author, condition, file, filepath) %>%  
 dplyr::filter(!is.na(condition))  
  
## Make dataset list from table  
ht\_tts\_dataset\_list <- setNames(split(ht\_tts\_metadata, seq(nrow(ht\_tts\_metadata))), ht\_tts\_metadata$dataset\_id)

## Display datasets  
DT5 <- DT::datatable(ht\_tts\_metadata %>% dplyr::select(-filepath, -author), options = list(dom = '', pageLength = 20, autoWidth = TRUE, columnDefs = list(list(width = '200px', targets = c(4)))))  
DT5



### Datasets per author

This part is customized depending on the originally published files. Since those files are formatted each in their own way, this part is partly manual and should be modified carefully.

#### Conway et al., 2014

Reference article: [Conway et al., 2014](https://doi.org/10.1128/mbio.01442-14)

Conditions:

* Exponential phase

Notes:

* Genome coordinates were converted from version U00096.2 to version U00096.3 using the library [EcoliGenes](https://github.com/rioualen/EcoliGenes)

TR0003 <- read\_excel\_cm(ht\_tts\_dataset\_list$TR0003$filepath, sheet = 1, comment.char = "#") %>%  
 dplyr::mutate(term\_pos = gsub("^[T]{1,2}-", "", Term), strand = Strand) %>%  
 dplyr::select(strand, term\_pos) %>%  
 dplyr::distinct() %>%  
 dplyr::mutate(term\_pos = EcoliGenes::convert\_coords(term\_pos)) %>%  
 dplyr::mutate(start = term\_pos, stop = term\_pos) %>%  
 dplyr::select(start, stop, strand, term\_pos)

#### Ju et al, 2019

Reference article: [Ju et al., 2019](https://doi.org/10.1038/s41564-019-0500-z)

Conditions:

* exponential phase
* stationary phase

TR0001 <- read\_excel\_cm(ht\_tts\_dataset\_list$TR0001$filepath, sheet = 1, comment.char = "#") %>%  
 dplyr::rename(term\_pos = TTS\_position,  
 strand = TTS\_strand) %>%   
 dplyr::mutate(start = term\_pos,   
 stop = term\_pos) %>%  
 dplyr::select(start, stop, strand, term\_pos) %>%  
 dplyr::distinct()  
  
TR0002 <- read\_excel\_cm(ht\_tts\_dataset\_list$TR0002$filepath, sheet = 1, comment.char = "#") %>%  
 dplyr::rename(term\_pos = TTS\_position,  
 strand = TTS\_strand) %>%   
 dplyr::mutate(start = term\_pos,   
 stop = term\_pos) %>%  
 dplyr::select(start, stop, strand, term\_pos) %>%  
 dplyr::distinct()

#### Yan et al., 2018

Reference article: [Yan et al., 2018](10.1038/s41467-018-05997-6)

Conditions:

* M9 (minimal growth medium)
* Rich (rich growth medium)

Notes:

* Two types of TU termination are reported for each condition: detected TTS or longest read end. The latter was not included.

TR0004 <- read\_excel(ht\_tts\_dataset\_list$TR0004$filepath, skip = 0) %>%  
 dplyr::mutate(strand = TU\_strand,   
 term\_pos = ifelse(strand == "-", TU\_left\_position, TU\_right\_position),  
 start = term\_pos,  
 stop = term\_pos) %>%  
 dplyr::select(start, stop, strand, term\_pos) %>%  
 dplyr::distinct()  
  
  
TR0005 <- read\_excel(ht\_tts\_dataset\_list$TR0005$filepath, skip = 0) %>%  
 dplyr::mutate(strand = TU\_strand,   
 term\_pos = ifelse(strand == "-", TU\_left\_position, TU\_right\_position),  
 start = term\_pos,  
 stop = term\_pos) %>%  
 dplyr::select(start, stop, strand, term\_pos) %>%  
 dplyr::distinct()

### Uniformization

This part assumes that the specificities of each individual dataset were dealt with previously. It generates unique IDs for terminators and writes one file per dataset.

tts\_datasets\_list <- list()  
  
for(ds\_id in names(ht\_term\_dataset\_list)) {  
 df <- get(ds\_id) %>%  
 dplyr::mutate(id = paste0("TTS\_", ds\_id, "\_", dplyr::row\_number()),  
 start = term\_pos,  
 stop = term\_pos,  
 chromosome = "NC\_000913.3") %>%  
 dplyr::arrange(start)  
  
 write.table(df %>% dplyr::select(chromosome, start, stop, id, term\_pos, strand), file = paste0(ht\_tts\_dir, "/uniform\_files/", ds\_id, ".tsv"), sep = "\t", quote = F, col.names = T, row.names = F)  
   
 assign(ds\_id, df)  
 tts\_datasets\_list[[ds\_id]] <- df  
}

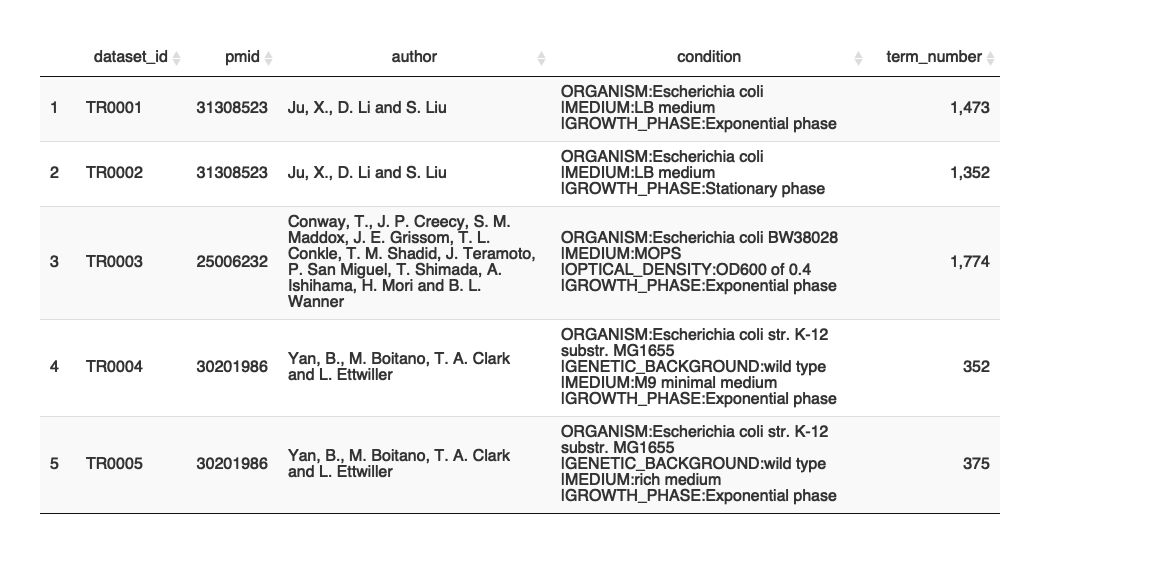
* Write a summary file for TTS datasets

ht\_tts\_stats <- ht\_tts\_metadata %>%  
 dplyr::rowwise() %>%  
 dplyr::mutate(term\_number = nrow(get(dataset\_id))) %>%  
 dplyr::select(dataset\_id, pmid, author, condition, term\_number)  
  
write.table(ht\_tts\_stats, file = paste0(ht\_tts\_dir, "/TTS\_datasets.tsv"), sep = "\t", quote = F, col.names = T, row.names = F)

### Results

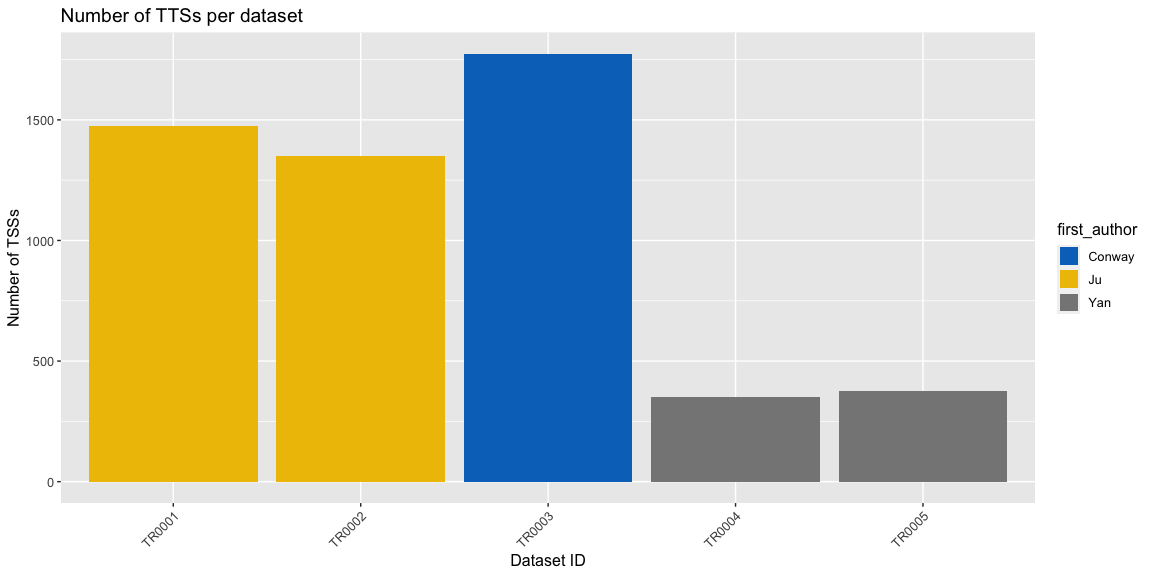
Summary of datasets:

DT6 <- DT::datatable(ht\_tts\_stats, options = list(dom = '', pageLength = 20)) %>% DT::formatRound('term\_number', digits=0)  
DT6



Distribution

ht\_tts\_stats\_legends <- ht\_tts\_stats %>%  
 dplyr::mutate(first\_author = str\_split(author, ",")[[1]][1])  
  
 ggplot(ht\_tts\_stats\_legends, aes(x = dataset\_id, y = term\_number, fill = first\_author)) +  
 geom\_col(size = 2) +  
 theme(axis.text.x = element\_text(angle = 45, hjust=1), text = element\_text(size = 12)) +  
 ylab("Number of TSSs") +  
 xlab("Dataset ID") +  
 labs(title = "Number of TTSs per dataset") +  
 ggsci::scale\_fill\_jco()



save.image(file = paste0("HT\_collections.Rdata"))