```
---
title: "RNA-seq analysis of *Pseudomonas* S5 genes associated with motility - Supplementary materials"
author: ""
header-includes:
- \pagestyle{plain}
- \usepackage{booktabs}
- \usepackage{longtable}
- \usepackage{floatrow}
- \floatsetup[table]{capposition=top}
output:
  pdf_document:
  toc: false
number_sections: false
---

```{r setup, include = FALSE, cache = FALSE, eval=T}
rm(list=ls())
library(RefManageR)
bib <- ReadBib("suppD.bib")
BibOptions(check.entries = FALSE, style = "markdown", cite.style = "authoryear",
           bib.style = "numeric")
```

<div style="text-align: justify">

<!-- ** -->
<!-- RNA-SEQ DATA ANALYSIS MOTILITY-ASSOCIATED GENES PSEUDOMONAS S5 - SUPPLEMENTARY DATA -->
<!-- Spring 2016 - MLS - UNIL - Marie Zufferey -->
<!-- ** -->

```{r data_preparation, echo=FALSE, eval=T, include=F}
setwd("/home/user/Documents/UNI/SP16/SAGE2/scripts")
outfolder = "report_SM"
system(paste("rm -rf", outfolder))
system(paste("mkdir", outfolder)) #not overwritten if already existing
source("functions_4.R")
library(edgeR)
# library(readr)
library(ggplot2)
library(pheatmap)
library(reshape2)
# library(rtracklayer)
library(magrittr)
library(dplyr)
library(VennDiagram)
library(vegan)
library(genoPlotR)
library(knitr)
library(phia)

#*******************************************
# DATA PREPARATION
#*******************************************

annot <- read.csv("../data/annot_mot.csv", sep=",")
rawannot <- read.csv("../data/annot_mot.csv", sep=",")
S5_stat <- read.csv("../data/Pseud_S5_stat.txt", sep="\t")
S5_stat3d <- read.csv("../data/Pseud_S5_stat_3d.txt", sep="\t")
S5_stat12d <- read.csv("../data/Pseud_S5_stat_12d.txt", sep="\t")
gbkData <- read.csv("../data/S5_gbk_short.csv", sep=",")
abd_fld <- "../data/abundances/"
dt <- getDGE(abd_fld)
rawdt <- getRawData(abd_fld, threshold=T)

##### Manual curation motility genes
a <- as.character(gbkData$Locus_tag[which(
  regexpr("pilus|motility|mobility|flagella|swarming|flagellum|pili", gbkData$Function)>0)])
all(a %in% annot$Gene_position) # TRUE -> ok
b <- as.character(gbkData$Locus_tag[which(
  regexpr("pilus|motility|mobility|flagella|swarming|flagellum|pili", gbkData$Product)>0)])
all(b %in% annot$Gene_position) # F
b[which(! b %in% annot$Gene_position)]

gbkData[gbkData$Locus_tag %in% b[which(! b %in% annot$Gene_position)],]
```
```

```
# Type Strand   Start     End      Locus_tag Gene_id                                Product Function
# 445   CDS      +  477446  478882  S5_genome_522       0            pilus assembly protein
PilQ        0
# 1042  CDS      - 1145050 1145361 S5_genome_1109       0 motility quorum-sensing regulator
MqsR        0
# 2060  CDS      - 2236727 2237314 S5_genome_2116       0            pilus assembly protein
PilZ        0
# 2071  CDS      - 2246411 2246710 S5_genome_2127       0                 pilus assembly
protein         0
# 3974  CDS      - 4407774 4408310 S5_genome_4013       0            type I pilus protein CsuA/
B         0
# 4334  CDS      + 4831767 4832201 S5_genome_4365       0            pilus assembly protein
PilZ        0
# 4759  CDS      - 5275681 5276040 S5_genome_4781       0            pilus assembly protein
PilZ        0

##### Manual curation chemotaxis
c <- grep("che", gbkData$Gene_id) # 5
c[which(! gbkData$Locus_tag[c] %in% annot$Gene_position)]   #5
gbkData[c,]
# Type Strand   Start     End      Locus_tag Gene_id
Product                                    Function
# 1123  CDS      + 1242569 1243579 S5_genome_1190   cheB2 Chemotaxis response regulator protein-
glutamate Involved in the modulation of the chemotaxis
# 1761  CDS      + 1915176 1915547 S5_genome_1824    cheY                          Chemotaxis protein
CheY      Involved in the transmission of sensory
# 1762  CDS      + 1915578 1916366 S5_genome_1825    cheZ                         Protein phosphatase
CheZ         Plays an important role in bacterial
# 1764  CDS      + 1918694 1919809 S5_genome_1827   cheB1 Chemotaxis response regulator protein-
glutamate Involved in the modulation of the chemotaxis
# 4546  CDS      - 5056065 5056892 S5_genome_4573    cheR            Chemotaxis protein
methyltransferase            Methylation of the membrane-bound

## Done manually
# colnames(annot): Gene_position Gene_name Motility_type
# we do not add the S5_genome_1109 and S5_genome_4013
addAnnot <- read.table(textConnection("
S5_genome_522 pilQ  pili
S5_genome_2116  pilZ  pili
S5_genome_2127  no_name2127 pili
S5_genome_4365  pilZ  pili
S5_genome_4781  pilZ  pili"), header=F)
colnames(addAnnot) <- c("Gene_position", "Gene_name", "Motility_type")
annot <- read.csv("../data/annot_mot.csv", sep=",")
annot <- rbind(annot, addAnnot)

addAnnot_c <- read.table(textConnection("
S5_genome_1190  cheB2 chemotaxis
S5_genome_1824  cheY chemotaxis
S5_genome_1825  cheZ chemotaxis
S5_genome_1827  cheB1 chemotaxis
S5_genome_4573  cheR chemotaxis"), header=F)
colnames(addAnnot_c) <- c("Gene_position", "Gene_name", "Motility_type")
annot_chemo <- rbind(annot, addAnnot_c)
annot <- annot[-which(annot$Gene_position=="S5_genome_4011"),]
annot <- annot[-which(annot$Gene_position=="S5_genome_1619"),]
annot_chemo <- annot_chemo[-which(annot_chemo$Gene_position=="S5_genome_4011"),]
annot_chemo <- annot_chemo[-which(annot_chemo$Gene_position=="S5_genome_1619"),]
figNb <- 1
```

As a supplement to the main text, we present in this document further investigations of the *Pseudomonas* S5 RNA-seq data.
All analyses were conducted in R (`r version$version.string`). We used the following packages: edgeR `r Citep(bib, "Robinson2009")`, phia `r Citep(bib, "Helios2015")` and vegan `r Citep(bib, "Oksanen2016")` for the statistical analyses,
genoPlotR `r Citep(bib, "Lionel2010")`, ggplot2 `r Citep(bib, "Wickham2009")`, pheatmap `r Citep(bib, "Kolde2015")` and VennDiagram `r Citep(bib, "Chen2016")` for the graphics,
dplyr `r Citep(bib, "Wickham2015")`, knitr `r Citep(bib, "Xie2013")`, magrittr `r Citep(bib, "Bache2014")` and reshape2 `r Citep(bib, "Wickham2007")`for data manipulation.
The script from which this document is generated as well as additional Perl scripts used during the analysis are given at the end of this document.

Quality assessment and data exploration

Histograms count data (after log-normalization)

We checked first the distribution of the counts. We presented here the histograms after log-normalization of count data (histograms of RPKM values not shown, but available in the script). After log-normalization, the counts data seem approximately normally distributed.

```{r dataExp, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=5, fig.align='center', warning=F}
# In edgeR, you should run calcNormFactors() before running rpkm(), for example:

counts <- getRawCounts(abd_fld) %>% as.data.frame

counts$Seq_tag <- rownames(counts)

len <- S5_stat[,c("Seq_tag", "Length")]

counts_len <- left_join(counts, len, by="Seq_tag")
counts_len2 <- counts_len
counts_len[,-c(ncol(counts_len), ncol(counts_len)-1)] %<>% calcNormFactors
counts_len[,-c(ncol(counts_len), ncol(counts_len)-1)] %<>% rpkm(., counts_len$Length) %>%log

counts_len2[,-c(ncol(counts_len2), ncol(counts_len2)-1)] %<>% myrpkm(., counts_len$Length)

rawcounts <- getRawCounts(abd_fld)

cpms <- cpm(rawcounts)

keep <- rowSums(cpms > 1) >= 4
countsFilter <- rawcounts[keep,]
pseudocountsFilter <- log2(countsFilter+1) %>% as.data.frame

# par(mfrow = c(1,1))
# dev.off()
for(i in colnames(counts)[1:16]){
#
p <- ggplot(counts, aes_string(x =as.name(i))) +
  geom_histogram(binwidth=2000, fill = "#525252")+
  scale_y_continuous(name="Log of raw counts")+
  theme(  panel.grid.minor.y=element_blank(),
          panel.grid.major.y=element_blank(),
          panel.grid.minor.x=element_blank(),
          panel.grid.major.x=element_blank())

q <- ggplot(counts_len, aes_string(x =as.name(i)))+
  geom_histogram(binwidth=2000, fill = "#525252")+
  scale_y_continuous(name="Log of RPKM")+
    theme(  panel.grid.minor.y=element_blank(),
          panel.grid.major.y=element_blank(),
          panel.grid.minor.x=element_blank(),
          panel.grid.major.x=element_blank())

r <- ggplot(pseudocountsFilter, aes_string(x =as.name(i)))+
  geom_histogram(binwidth=2000, fill = "#525252")+
  scale_y_continuous(name="Log2(CPM+1)/")+
    theme(  panel.grid.minor.y=element_blank(),
          panel.grid.major.y=element_blank(),
          panel.grid.minor.x=element_blank(),
          panel.grid.major.x=element_blank())


# multiplot(p,q,r,cols=3)

}

```

```{r dataExp2, echo=FALSE, eval=T, include=F, fig.height=4,fig.width=5, fig.align='center', warning=F}
df <- melt(pseudocountsFilter)
df <- data.frame(df, Condition = substr(df$variable,1,2))
```

```{r dataExp3, echo=FALSE, eval=T, include=T, fig.height=5,fig.width=6, fig.align='center', warning=F}

par(mfrow = c(1,1))
ggplot(df, aes(x = value, colour = variable, fill = variable)) +
  ylim(c(0, 0.25)) +
  geom_density(alpha = 0.2, size = 1.25) +
  facet_wrap(~ Condition) +
  theme(legend.position = "top") +
  xlab(expression(log[2](count + 1)))
k<-dev.off()

```
######*Figure S`r figNb `. Density plot of log-normalized count data for the four experimental conditions.*

\newpage

Biological coefficient of variation

We use the plotBCV function "which shows the root-estimate, i.e., the biological coefficient of variation for each gene" (Chen et al. 2015) to plot the genewise biological coefficient of variation (BCV) against gene abundance (in log2 counts per million).

The y-axis represents the BCV. This latter is "the coefficient of variation with which the (unknown) true abun-dance of the gene varies between replicate RNA samples. It represents the CV that would remain between biological replicates if sequencing depth could be increased indefinitely. [...] [It] is
reasonable to suppose that BCV is approximately constant across genes." `r Citep(bib, "Chen2015")`.
The black dots allow to appreciate the dispersion across reads (tags).
Witch BCV plots, "estimation of genewise BCV allows observation of changes for genes that are consistent between biological replicates and giving less priority to those with inconsistent results" `r Citep(bib, "Diray2015")`.


```{r edgeR_BCV, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=5, fig.align='center', warning=F}
figNb <- figNb+1
par(mfrow=c(1,1))
plotBCV(dt)
k<-dev.off()
```

######*Figure S`r figNb `. Plot of biological coefficient of variation.*

Multidimensional scaling plot of distance between expression profiles

We used here the plotMDS function. This latter plots samples on a two-dimensional scatterplot so that distances on the plot approximate the expression differences between the samples.
It "produces a plot in which distances between samples
correspond to leading biological coefficient of variation (BCV) between those samples" (Chen et al. 2015).

Here, we could also check that the replicates for a given condition cluster well together. This is mostly the case, except for the replicate "SA4" that seems more distinct that the three other SA replicates.


```{r ex_mds, echo=FALSE, eval=F, include=F,  warning=F}
# example of MDS plot interpretation
# In the plot, dimension 1 separates the tumor from the normal samples, while dimension 2
# roughly corresponds to patient number. This confirms the paired nature of the samples. The
# tumor samples appear more heterogeneous than the normal samples.
```

```{r edgeR_mds, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=10, fig.align='center', warning=F}
figNb <- figNb +1
calcdt <- calcNormFactors(rawdt)
mycol <- c(rep("dodgerblue3", 4), rep("goldenrod2", 4), rep("chartreuse3", 4), rep("forestgreen",4), rep("blue", 4))

par(mfrow=c(1,2))
```

```
plotMDS(calcdt, main = "MDS plot on samples", col=mycol, method="logFC")
# => convert the counts to log-counts-per-million using cpm and pass these to the limma plotMDS
function.
# This method calculates distances between samples based on log2 fold changes

plotMDS(calcdt, main = "MDS plot on samples", col=mycol, method="bcv")
# calculates distances based on biological coefficient of variation. A set of top genes are chosen
that have largest
# biological variation between the libraries (those with largest genewise dispersion treating all
libraries as one group).
# Then the distance between each pair of libraries (columns) is the biological coefficient of
variation (square root of
# the common dispersion) between those two libraries alone, using the top genes
k<-dev.off()
```

######*Figure S`r figNb `. MDS plots for logFC (left) and BCV (right).*

Multivariate analyses

PCA

```{r pca, echo=FALSE, eval=T, include=T, fig.height=5,fig.width=10, fig.alig='center', warning=F}
figNb <- figNb+1
dt <- getDGE(abd_fld)
meanPcF <- getMeanData(dt$counts)

gene1.pca <- rda(meanPcF, scale=T)
# gene1.pca
# summary(gene1.pca)  # default scaling=2
# summary(gene1.pca, scaling=1)
# biplot(gene1.pca, scaling=1, main="PCA - scaling 1") # distance; angles meaningless
# biplot(gene1.pca, scaling=2, main="PCA - scaling 2") # angles; distance meaningless

mycol <- sapply(rownames(dt$counts), function(x){
  if(x %in% annot$Gene_position){
    "darkorange"
    }else{"black"}
  })

par(mfrow=c(1,1))
cleanplot.pca(gene1.pca, mycol=mycol)
foo <- dev.off()

meanPcF$Seq_tag <- rownames(meanPcF)
rownames(meanPcF) <- NULL

statsCDS <- read.csv("../data/Pseud_S5_stat.txt", sep="\t")

meanAndStat <- left_join(meanPcF, statsCDS, by="Seq_tag")

rownames(meanAndStat) <- meanAndStat$Seq_tag
meanAndStat$Seq_tag <- NULL
gene.pca2 <- rda(meanAndStat, scale=T)
```

######*Figure S`r figNb `. PCA plots for all genes and all conditions (mean data). Left: scaling 1
(angles are meaningless), right: scaling 2 (distances are meaningless).*

PCA by condition (with coloured motility-associated genes)

```{r pca2b, echo=FALSE, eval=T, include=T, fig.height=5,fig.width=10, fig.alig='center', warning=F}
figNb <- figNb+1
dt <- getDGE(abd_fld)
motD <- dt$counts[which(rownames(dt$counts) %in% annot$Gene_position),]
motD %<>% as.data.frame
tx <- motD
tx$Tr <- rownames(tx)
tx <- left_join(tx, S5_stat, by=c("Tr"="Seq_tag"))

motD %<>% myrpkm(., tx$Length)
tx <- left_join(tx, annot, by=c("Tr"="Gene_position"))
```

```
### DOTS ARE THE GENES COLOURED BY MOTILITY TYPE


plotPCA2 <- function(i){

  pca1 <- prcomp((motD[,grep(i, colnames(motD))]))
  x <- pca1$x[,1]
  y <- pca1$x[,2]


  motCol <- sapply(tx$Motility_type, function(x) {setMyCol_PCA(x)})
  motPch <- sapply(tx$Motility_type, function(x) {setMyPch_PCA(x)})

  if(i == "LM") legpos <- "topleft"
  if(i == "SA") legpos <- "topright"
  if(i == "WL") legpos <- "bottomleft"
  if(i == "WR") legpos <- "topleft"

  labx <- paste0(colnames(summary(pca1)$importance)[1],
                 " (", round(summary(pca1)$importance[2,1],4)*100, "%)")
  laby <- paste0(colnames(summary(pca1)$importance)[2],
                 " (", round(summary(pca1)$importance[2,2],4)*100, "%)")


  plot(x,y, col=motCol, pch=motPch,cex=1.5,lwd=3, xlab=labx,ylab=laby, main=i)
  legend(legpos,  legend=c("flagella", "pili", "swarming"),col=unique(motCol),
         pch=unique(motPch),cex=1, bty="o")

}
```

```{r pcaplot, echo=FALSE, eval=T, include=T, fig.height=10,fig.width=10, fig.alig='center', warning=F}
par(mfrow=c(2,2))
plotPCA2("LM")
plotPCA2("SA")
plotPCA2("WL")
plotPCA2("WR")
foo <- dev.off()
```

######*Figure S`r figNb `. PCA plots for motility-associated genes only for all conditions separately.*

\newpage

Heatmap

Here we looked at the gobal level of expression across all replicates conditions of motility-associated genes. We observed that the replicates of a given experimental condition do not necessarily cluster together.


```{r heatmap, echo=FALSE, eval=T, include=T, fig.height=8,fig.width=6, fig.align='center', warning=F}
figNb <- figNb +1
rc1 <- rawcounts %>% as.data.frame
rc1$Tr <- rownames(rc1)
rownames(rc1) <- NULL
rc1 %<>% as.data.frame

c1 <- left_join(annot, rc1, by=c("Gene_position"="Tr"))
c2 <- left_join(c1, S5_stat,by=c("Gene_position"="Seq_tag"))

temp <- c2[,c(4:22)]

temp <- apply(temp, c(1,2), function(x){as.numeric(as.character(x))})

c2[,c(4:22)] <- temp

RP <- myrpkm(c2[,c(4:19)], c2$Length) %>% log2

rowlab <-  as.character(c2$Gene_name)

dup <- rowlab[which(duplicated(rowlab))]
dup2 <- paste0(dup, "b")
dup3 <- dup2[which(duplicated(dup2))]
```

```
dup4 <- paste0(dup3, "2")

dup2[which(duplicated(dup2))] <- dup4

rowlab[which(duplicated(rowlab))] <- dup2

rownames(RP) <- rowlab

pheatmap(RP, main = 'Heatmap motility genes',label_col="red")

k<-dev.off()
```
######*Figure S`r figNb `. Heatmap for counts data for all replicates (motility-associated genes only).*

Boxplot RPKM (without and with chemotaxis-associated genes)

Here, we compared the RPKM between all conditions (and also between all replicates).
We also included chemotaxis genes (5 *che* family genes found by searching in the data frame exported from GenDB), to see if they exhibit the same pattern of expression level as one kind of motility (plots on the right here below).

We noticed that the level of expression (log of RPKM values) is quite homogeneous for the replicates of a given condition.
Interestingly, the genes involved in chemotaxis seem more expressed in conditions where plant material is present, consistent with plant-oriented motility (as discussed in the main text). Further investigations would be needed (e.g. statistical tests, include more chemotaxis genes).

```{r prepboxplot, echo=FALSE, eval=T, include=F, warning=F}
dt_raw <- getRawData(abd_fld)
all_data <- dt_raw$counts %>% as.data.frame  #6087
all_data$Tra <- rownames(all_data)

mot_data <-left_join(annot_chemo, all_data, by=c("Gene_position"="Tra")) %>%
  left_join(., S5_stat, by=c("Gene_position"="Seq_tag"))

# we want to compare across genes and across conditions -> RPKM
mot_data[,grep("1|2|3|4", colnames(mot_data))] %<>% rpkm(., mot_data$Length)

# take the mean of the replicates for all conditions
#mot_data2 <- cbind(mot_data[,1:3], getMeanData(mot_data))
mot_data2 <- mot_data[,1:(ncol(mot_data)-3)]

# select only motility genes (without chemotaxis)
data_mot <- mot_data2[which(mot_data2$Motility_type!="chemotaxis"),]
```

```{r boxplotRPKM, echo=FALSE, eval=T, include=T, fig.height=16,fig.width=20,fig.align='center', warning=F}
p <- boxplotMotGenes(data_mot, annot, "Global expression mot. genes", chemo=F, allRep=T)
q <- boxplotMotGenes(data_mot, annot, "Global expression mot. genes", chemo=F)

r <- boxplotMotGenes(mot_data2, annot, "Global exp. mot. genes (with chemo.)", chemo=T, allRep=T)
s <- boxplotMotGenes(mot_data2, annot, "Global exp. mot. genes (with chemo.)", chemo=T)

multiplot(p,q,r,s,cols=2)
figNb <- figNb+1
```

######*Figure S`r figNb `. Boxplots of the log of RPKM values by motility type, for all replicates (top) or for the four conditions (bottom), without (left) and with (right) chemotaxis-associated genes.*

\newpage

Differential expression

In the same way as for the main text, we considered for these analyses only the genes exhibiting a statistically significant change in differential expression (adjusted p-values < 0.05).

Histogram of p-values and plots logFC vs. logCPM (M vs. A)

MA plot: plot the log-fold change (i.e. the log of the ratio of expression levels for each gene

between two experimential groups) against the log-concentration (i.e. the overall average expression level for each gene across the two groups).

Here, we drew "smear plots" (average logCPM in x-axis, logFC in y-axis) for all pairs of comparisons. We added to the plots the label of the motility-associated genes that we annotated (see figure legend). Please notice that the y-axis is not always on the same scale.

As they are neither particularly informative nor conclusive, histograms of adjusted p-values are not shown here (but the code is available in the R script).

On the smear plots here below, we noticed global variations of the change in gene expression. For example, it seems that gene expression varies slightly in LM vs. WR (less red dots).
When root material is present (SA vs. WR, WL vs. WR), a global trend of upregulation is visible (more red dots in the upper part of the plot). It seems to be the opposite ("global" downregulation) in LM vs. WL.
Broadly, we observed that the genes we annotated are not the ones that exhibit the most important changes in gene expression (not the highest on the y-axis) and have a broad-ranged level of expression (from middle to right part of the cloud of points). For the motility-associated genes, no clear trends emerge from these smear plots.

```{r tutoKA_smear, echo=FALSE, eval=T, include=F,warning=F}

conditions <- c("LM", "SA", "WL", "WR")
combCond <- combn(conditions,2)

plotSmearAll <- function(i){
#for(i in ncol(combCond)){ # not working
  cond1 = combCond[1,i]
  cond2 = combCond[2,i]

  de <- exactTest(dt, pair = c(cond1, cond2))

  #hist(de$table$PValue, breaks = 50, xlab = 'p-value (without correction)',
   #    main = paste("Histogram non adjusted p-values", cond2, "vs.", cond1))

  # gathering differential expressed genes
  tT <- topTags(de, n = nrow(dt))
  # tabular form of differentially expressed genes
  deg.list <- tT$table


  ## take the row names of the differentially expressed genes that have locus ID
  locus.ids <- rownames(deg.list)
  # select genes that have 1% false discovery
  top.deg <- locus.ids[deg.list$FDR < .01 & abs(deg.list$logFC) > 1]

  # plotSmear is a more sophisticated and superior way to produce an 'MA plot'. plotSmear resolves the problem of
  # plotting genes that have a total count of zero for one of the groups by adding the 'smear' of points at low A value.
  ourGenes <- annot$Gene_position
  ourGenesID <- annot$Gene_name


  plotSmear(dt, pair=c(cond1, cond2), main = paste("Smear plot", cond1, "vs.", cond2), lowess=F, de.tags = top.deg)

  text(x = deg.list$logCPM[which(rownames(deg.list)%in%ourGenes)],
       y = deg.list$logFC[which(rownames(deg.list)%in%ourGenes)],
       labels = ourGenesID, cex=0.8, pos=1, col=sapply(annot$Motility_type, setMyCol_smear))

}

```

```{r tutoKA_smear1a, echo=FALSE, eval=T, include=T,fig.height=5,fig.width=11, fig.align='center', warning=F}
par(mfrow=c(1,2))
plotSmearAll(1)
plotSmearAll(2)
k<-dev.off()
```

```{r tutoKA_smear1b, echo=FALSE, eval=T, include=T,fig.height=10,fig.width=11, fig.align='center', warning=F}
par(mfrow=c(2,2))
plotSmearAll(3)
plotSmearAll(4)
plotSmearAll(5)
plotSmearAll(6)
k<-dev.off()
figNb <- figNb+1
```

######*Figure S`r figNb `. Smear plots for differential expression between all pairs. Genes with more than twofold change of expression shown in red. Motility-associated genes (labelled) shown in orange (flagellum-related), blue (pilus-related) and green (swarming-related). Please notice the different scales of the y-axis.*

\newpage

Scatterplot matrix: correlation between differential expression pairs

We drew scatterplot matrix to compare the differential expression between pairs of pairwise comparisons (motility-associated genes only).
We noticed that change in differential expression is sometimes highly correlated (e.g. WR vs. SA and WL vs. SA or SA vs. LM and WL vs. SA), and sometimes not (e.g. SA vs. LM and WL vs. LM or WR vs. LM and WR vs. WL). As explained in the main text, we used this plot to decide which comparison to examine more in detail.

```{r scattMat, echo=FALSE, eval=T, include=T,fig.height=14,fig.width=14, fig.align='center', warning=F}
# First we do the matrix with all pairs of conditions
# it will allow us to justify which pairs we choose
# before merging, select only needed data
# (not mandatory)

dt <- getDGE(abd_fld)  # normalize
# Pairwise comparisons
# exact test for the 2 conditions passed in argument (last 2 arguments)
# for a given set of genes (2nd argument)
dataLMSA <- pairTestGenes(dt, annot$Gene_position , "LM", "SA")  #1
dataLMWL <- pairTestGenes(dt, annot$Gene_position , "LM", "WL")  #2
dataLMWR <- pairTestGenes(dt, annot$Gene_position , "LM", "WR")  #3
dataSAWL <- pairTestGenes(dt, annot$Gene_position , "SA", "WL")  #4
dataSAWR <- pairTestGenes(dt, annot$Gene_position , "SA", "WR")  #5
dataWLWR <- pairTestGenes(dt, annot$Gene_position , "WL", "WR")  #6
allPairs <- rbind(dataLMSA, dataLMWL, dataLMWR, dataSAWL, dataSAWR, dataWLWR)


subLMSA <- dataLMSA[,c("logFC", "FDR", "Transcript")]      #1
colnames(subLMSA)[1:2] %<>% paste0(., ".LMSA")
subLMWL <- dataLMWL[,c("logFC", "FDR", "Transcript")]      #2
colnames(subLMWL)[1:2] %<>% paste0(., ".LMWL")
subLMWR <- dataLMWR[,c("logFC", "FDR", "Transcript")]      #3
colnames(subLMWR)[1:2] %<>% paste0(., ".LMWR")
subSAWL <- dataSAWL[,c("logFC", "FDR", "Transcript")]      #4
colnames(subSAWL)[1:2] %<>% paste0(., ".SAWL")
subSAWR <- dataSAWR[,c("logFC", "FDR", "Transcript")]      #5
colnames(subSAWR)[1:2] %<>% paste0(., ".SAWR")
subWLWR <- dataWLWR[,c("logFC", "FDR", "Transcript")]      #6
colnames(subWLWR)[1:2] %<>% paste0(., ".WLWR")

# merge all in a single DF
allJoins <- full_join(subLMSA, subLMWL, by="Transcript") %>%   #1,2
  full_join(., subLMWR, by="Transcript")  %>% #3
  full_join(., subSAWL, by="Transcript")  %>% #4
  full_join(., subSAWR, by="Transcript")  %>% #5
  full_join(., subWLWR, by="Transcript")  #6

# convert into a matrix with only logFC values
matAllJoins <- allJoins
rownames(matAllJoins) <- matAllJoins$Transcript
matAllJoins <- matAllJoins[,grep("log", colnames(matAllJoins))]
# change the colnames for nicer titles in the matrix plot
colnames(matAllJoins) %<>% gsub("logFC.", "",.) %<>%
```

```
  gsub('(^.{2})(.{2}$)', '\\2 vs. \\1', .)

pairs(matAllJoins,panel=panel.smooth, upper.panel=panel.cor,
      diag.panel=panel.hist)  # panel.hist defined in functions_4.R
title("Log2FC for motility associated genes - all pairs", line=3)
figNb <- figNb+1
```

######*Figure S`r figNb `. Scatterplot matrix: correlation between differential expression between pairs of conditions (motility-associated genes only.*

\newpage

Heatmap for all pairs of comparisons

Here, we used a heatmap for visualization of differential expression in all pairs of comparisons.
We noticed that the profile of differential expression is sometimes very similar (e.g. SA vs. wR or SA vs. WL).
For all tests of differential expression, we only retained the genes for which the adjusted p-value was below 0.05 (hence the grey cases).

```{r heatmapDE, echo=FALSE, eval=T, include=T, fig.height=7,fig.width=7, fig.align='center', warning=F}

heatmapPairs(allPairs, gbkData[,c("Start", "Locus_tag")], annot, "Heatmap all conditions pairwise") %>
% plot

figNb <- figNb+1
```

######*Figure S`r figNb `. Heatmap of differential expression for all pairs of conditions (motility-associated genes only; only statistically significant genes with adjusted p-values < 0.05 are shown). Y-axis: genes are ordred according to their position on the chromosome.*

Volcano plots: all genes and motility-associated genes

Next, we drew the volcano plots for all pairs of conditions.
They provide more precise information than the heatmap.
Because of time limitation, we could not discuss all pairs of conditions.
We observed nonetheless that motility-associated genes are not the genes that exhibit the most important changes in expression (left column).
The three genes with the most changing expression are labelled (e.g. 3353 corresponds to the CDS S5_genome_3353).
We used an easy-to-use custom Perl script (provided at the end of this document; blast ouptuts available in the data folder) to investigate quickly the function of these genes (920: siderophore receptor; 3338: cytochrome oxidase; 3353, 5852: transport proteins; 4845: bacterioferritin-associated ferredoxin, 5853: import protein; all others: uncharacterized proteins).

We noticed also that the differential expression of flagella and pili in some cases shows a clear opposite pattern (e.g. SA vs. LM, WR vs. SA; discussed in the main text), although this tendency is not obvious in all pairwise comparisons (e.g. WL vs. LM).
In particular, we noticed that the profile of WL vs. SA is very similar to the one of WR vs. SA discussed in detail in the main text.
As discussed in the main text, genes specifically associated with swarming more often exhibit the same pattern of differential expression than the one of pilus-associated genes.

```{r volcan1, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=10, fig.align='center', warning=F}
# not working in for loop
# for(i in ncol(combCond)){} [1,i][2,i]
figNb <- figNb+1
cond1 = combCond[1,1];cond2 = combCond[2,1]
p <- volcanoAllPoints(dt, annot, cond1, cond2, plotAnnot=T, myT=3)
q <- volcanoMotilityPointsAnnot(dt, annot, cond1, cond2)
multiplot(p,q, cols=2)
```

```{r volcan2, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=10, fig.align='center', warning=F}
cond1 = combCond[1,2];cond2 = combCond[2,2]
p <- volcanoAllPoints(dt, annot, cond1, cond2, plotAnnot=T, myT=3)
q <- volcanoMotilityPointsAnnot(dt, annot, cond1, cond2)
multiplot(p,q, cols=2)
```

```

```{r volcan3, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=10, fig.align='center', warning=F}
cond1 = combCond[1,3];cond2 = combCond[2,3]
p <- volcanoAllPoints(dt, annot, cond1, cond2, plotAnnot=T, myT=3)
q <- volcanoMotilityPointsAnnot(dt, annot, cond1, cond2)
multiplot(p,q, cols=2)
```

```{r volcan4, echo=FALSE, eval=T, include=T, fig.height=5,fig.width=10, fig.align='center', warning=F}
cond1 = combCond[1,4];cond2 = combCond[2,4]
p <- volcanoAllPoints(dt, annot, cond1, cond2, plotAnnot=T, myT=3)
q <- volcanoMotilityPointsAnnot(dt, annot, cond1, cond2)
multiplot(p,q, cols=2)
```

```{r volcan5, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=10, fig.align='center', warning=F}
cond1 = combCond[1,5];cond2 = combCond[2,5]
p <- volcanoAllPoints(dt, annot, cond1, cond2, plotAnnot=T, myT=3)
q <- volcanoMotilityPointsAnnot(dt, annot, cond1, cond2)
multiplot(p,q, cols=2)
# seems to have only 2 points but 803 and 208 => 1 point

```

```{r volcan6, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=10, fig.align='center', warning=F}
cond1 = combCond[1,6];cond2 = combCond[2,6]
p <- volcanoAllPoints(dt, annot, cond1, cond2, plotAnnot=T, myT=3)
q <- volcanoMotilityPointsAnnot(dt, annot, cond1, cond2)
multiplot(p,q, cols=2)
```

######*Figure S`r figNb `. Volcano plots for all pairs of comparisons (left column: all genes differentially expressed in a statistically significant manner (FDR < 0.05); right column: only motility-associated genes).*

\newpage

Up- and downregulation for all pairs

Again, we focused at the up- and downexpression for all pairs of conditions.
In fact, these plots show the same information as the volcano plots.
Here, it is particularly apparent that the fold change of expression of the motility-associated genes is rarely more than twofold (dashed line).

```{r barplot2a, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=3.5, fig.alig='center', warning=F}
par(mfrow = c(1,1))
tit <- "log 2 FC (SA vs. LM)"
fc_barAndCpm_line(dataLMSA, annot, gbkData, tit,pt=F) %>% grid.draw
tit <- "log 2 FC (WR vs. SA)"
fc_barAndCpm_line(dataSAWR, annot, gbkData, tit,pt=F) %>% grid.draw
foo <- dev.off()
```

```{r barplot2, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=3.5, fig.alig='center', warning=F}
figNb <- figNb+1
par(mfrow = c(1,1))
tit <- "log 2 FC (WL vs. LM)"
fc_barAndCpm_line(dataLMWL, annot, gbkData, tit,pt=F) %>% grid.draw
tit <- "log 2 FC (WR vs. LM)"
fc_barAndCpm_line(dataLMWR, annot, gbkData, tit,pt=F) %>% grid.draw
foo <- dev.off()
```

```{r barplot2b, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=3.5, fig.alig='center', warning=F}
grid.newpage()
par(mfrow = c(1,1))
tit <- "log 2 FC (WL vs. SA)"
fc_barAndCpm_line(dataSAWL, annot, gbkData, tit,pt=F) %>% grid.draw
tit <- "log 2 FC (WR vs. WL)"
fc_barAndCpm_line(dataWLWR, annot, gbkData, tit,pt=F) %>% grid.draw
```
```

```

######*Figure S`r figNb `. Barplot for all pairs of comparisons. Only motility-associated genes are shown. Dashed line indicating |logFC| = 1  (twofold change of expression). X-axis: genes ordered according to their chromosomal position.*

\newpage

# Association between gene expression and other gene characteristics

### GC content, purine content and gene length

Here we tried to see if some characteristics (GC content, purine content and length of the genes; computed with a short Perl script provided at the end of this document) of the genes could explain their level of expression (expressed in log of RPKM).
We noted a clear inverse correlation between the GC content and the expression level as well as between the length of the gene and the expression level (assessed using Spearman's correlation coefficient).
This correlation is stronger for the third codon position than for the first two codon positions (see plots and table below).
GC content has already been reported to be associated with gene expression in other species and phyla, e.g. neem `r Citep(bib, "Krishnan2011")`, chicken `r Citep(bib, "Rao2013")` or human `r Citep(bib, "Vinogradov2005")`. But technological biases should not be overlooked. In our case, we do not exactly know which biases could skew our data, but for example it has been reported that "GC-rich and GC-poor fragments tend to be under-represented in RNA-Seq" `r Citep(bib, "Risso2011")`.


```{r barplot3, echo=FALSE, eval=T, include=T,warning=F}
figNb <- figNb+1
S5_stat <- read.csv("../data/Pseud_S5_stat.txt", sep="\t")
S5_stat3d <- read.csv("../data/Pseud_S5_stat_3d.txt", sep="\t")
S5_stat12d <- read.csv("../data/Pseud_S5_stat_12d.txt", sep="\t")
S5_stat$ratioGC_3pos <- S5_stat3d$ratioGC
S5_stat$ratioGC_12pos <- S5_stat12d$ratioGC

counts <- getRawCounts(abd_fld) %>% as.data.frame
counts$Tr <- rownames(counts)
counts_Str <- left_join(counts, S5_stat, by=c("Tr" = "Seq_tag"))
counts_Str[,1:16] <- myrpkm(counts_Str[,1:16], counts_Str$Length)

meanD <- getMeanData(counts_Str[,1:16])

meanStr <- cbind(counts_Str, meanD)
meanStr$logLM <- log(meanStr$LM)
meanStr$logSA <- log(meanStr$SA)
meanStr$logWR <- log(meanStr$WR)
meanStr$logWL <- log(meanStr$WL)
meanStr$logLen <- log(meanStr$Length)


mycol <- sapply(meanStr$Tr, function(x){
  if(x %in% annot$Gene_position){
    y <- "violetred1"
  }else{
    y <- "black"
  }
  y
})

mycond <- c("logLM", "logSA", "logWR", "logWL")
plotGCcond <- function(i){
# for(i in mycond){
  p <- ggplot(meanStr, aes_string(x=i, y="ratioGC",group=1))+
          geom_point(size=2, colour=mycol)+
    geom_smooth(method = "lm", se = FALSE, colour="slateblue4")+
          #scale_y_continuous("Number of altered cases",
           #                  breaks=seq(0, max(my.genes$alterations),5))+
        scale_y_continuous("GC content")+
        scale_x_continuous("log(RPKM)")+
        ggtitle(paste0("Expression~GC (", substr(i, 4,5),")"))+
          theme(axis.title.y = element_text(face="bold", colour="#990000", size=15),
                axis.title.x = element_text(face="bold", colour="#990000", size=15),
                axis.text.y = element_text(colour="black", size=12),
                axis.text.x  = element_text(angle=90, vjust=0.5, size=12,
```
```

```
                                        lineheight=5,hjust=1),
              plot.title =  element_text(colour="darkslateblue", size=15),
              panel.grid.minor.y=element_blank(),
              panel.grid.major.y=element_blank(),
              panel.grid.minor.x=element_blank(),
              panel.grid.major.x=element_blank())

  q <- ggplot(meanStr, aes_string(x=i, y="ratioPu",group=1))+
          geom_point(size=2, colour=mycol)+
    geom_smooth(method = "lm", se = FALSE, colour="slateblue4")+
          #scale_y_continuous("Number of altered cases",
           #                  breaks=seq(0, max(my.genes$alterations),5))+
        scale_y_continuous("Purine content")+
        scale_x_continuous("log(RPKM)")+
        ggtitle(paste0("Expression~AG (", substr(i, 4,5),")"))+
          theme(axis.title.y = element_text(face="bold", colour="#990000", size=15),
              axis.title.x = element_text(face="bold", colour="#990000", size=15),
              axis.text.y = element_text(colour="black", size=12),
              axis.text.x  = element_text(angle=90, vjust=0.5, size=12,
                                        lineheight=5,hjust=1),
              plot.title =  element_text(colour="darkslateblue", size=15),
              panel.grid.minor.y=element_blank(),
              panel.grid.major.y=element_blank(),
              panel.grid.minor.x=element_blank(),
              panel.grid.major.x=element_blank())

  r <- ggplot(meanStr, aes_string(x=i, y="logLen",group=1))+
          geom_point(size=2, colour=mycol)+
    geom_smooth(method = "lm", se = FALSE, colour="slateblue4")+
          #scale_y_continuous("Number of altered cases",
           #                  breaks=seq(0, max(my.genes$alterations),5))+
        scale_y_continuous("log(Length)")+
        scale_x_continuous("log(RPKM)")+
        ggtitle(paste0("Expression~length (", substr(i, 4,5),")"))+
          theme(axis.title.y = element_text(face="bold", colour="#990000", size=15),
              axis.title.x = element_text(face="bold", colour="#990000", size=15),
              axis.text.y = element_text(colour="black", size=12),
              axis.text.x  = element_text(angle=90, vjust=0.5, size=12,
                                        lineheight=5,hjust=1),
              plot.title =  element_text(colour="darkslateblue", size=15),
              panel.grid.minor.y=element_blank(),
              panel.grid.major.y=element_blank(),
              panel.grid.minor.x=element_blank(),
              panel.grid.major.x=element_blank())


multiplot(p,q,r,cols=3)
}
```

```{r b1, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=18, fig.alig='center', warning=F}
mycond <- c("logLM", "logSA", "logWR", "logWL")
plotGCcond(mycond[1])
```

```{r b2, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=18, fig.alig='center', warning=F}
plotGCcond(mycond[2])
```

```{r b3, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=18, fig.alig='center', warning=F}
plotGCcond(mycond[3])
```

```{r b4, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=18, fig.alig='center', warning=F}
plotGCcond(mycond[4])
```

######*Figure S`r figNb `. For each condition, plot showing log of RPKM values for all genes against i) GC content of the gene (left column), ii) purine content of the gene (mid column), iii) length of the gene (right column). Motility-associated genes are shown with pink dots.*


```{r barplot4, echo=FALSE, eval=T, include=T, fig.height=6,fig.width=12, fig.alig='center', warning=F}
figNb <- figNb+1
```

```
# NOW PLOT GC-CONT 3D POS & GC-CONT 12D POS
plotGCposcond <- function(i){
# for(i in mycond){
  p <- ggplot(meanStr, aes_string(x=i, y="ratioGC_3pos",group=1))+
          geom_point(size=2, colour=mycol)+
    geom_smooth(method = "lm", se = FALSE, colour="slateblue4")+
        scale_y_continuous("GC content - 3d pos")+
        scale_x_continuous("log(RPKM)")+
        ggtitle(paste0("Expression~GC 3d pos. (", substr(i, 4,5),")"))+
          theme(axis.title.y = element_text(face="bold", colour="#990000", size=15),
                axis.title.x = element_text(face="bold", colour="#990000", size=15),
                axis.text.y = element_text(colour="black", size=12),
                axis.text.x  = element_text(angle=90, vjust=0.5, size=12,
                                            lineheight=5,hjust=1),
                plot.title =  element_text(colour="darkslateblue", size=15),
                panel.grid.minor.y=element_blank(),
                panel.grid.major.y=element_blank(),
                panel.grid.minor.x=element_blank(),
                panel.grid.major.x=element_blank())

   q <- ggplot(meanStr, aes_string(x=i, y="ratioGC_12pos",group=1))+
          geom_point(size=2, colour=mycol)+
    geom_smooth(method = "lm", se = FALSE, colour="slateblue4")+
        scale_y_continuous("GC content - 1st&2d pos")+
        scale_x_continuous("log(RPKM)")+
        ggtitle(paste0("Expression~GC 1-2nd pos. (", substr(i, 4,5),")"))+
          theme(axis.title.y = element_text(face="bold", colour="#990000", size=15),
                axis.title.x = element_text(face="bold", colour="#990000", size=15),
                axis.text.y = element_text(colour="black", size=12),
                axis.text.x  = element_text(angle=90, vjust=0.5, size=12,
                                            lineheight=5,hjust=1),
                plot.title =  element_text(colour="darkslateblue", size=15),
                panel.grid.minor.y=element_blank(),
                panel.grid.major.y=element_blank(),
                panel.grid.minor.x=element_blank(),
                panel.grid.major.x=element_blank())

multiplot(p,q,cols=2)
}


```

```{r b7, echo=FALSE, eval=T, include=T, fig.height=4, fig.width=15, fig.alig='center', warning=F}
plotGCposcond(mycond[1])
```

```{r b8, echo=FALSE, eval=T, include=T, fig.height=4, fig.width=15, fig.alig='center', warning=F}
plotGCposcond(mycond[2])
```

```{r b9, echo=FALSE, eval=T, include=T, fig.height=4, fig.width=15, fig.alig='center', warning=F}
plotGCposcond(mycond[3])
```

```{r b10, echo=FALSE, eval=T, include=T, fig.height=4, fig.width=15, fig.alig='center', warning=F}
plotGCposcond(mycond[4])
```

######*Figure S`r figNb `. For each condition, plot showing log of RPKM values for all genes against i) GC content at the third codon position (left column), ii) GC content at the first two positions (right column). Motility-associated genes are shown with pink dots.*

```{r barplot4b, echo=FALSE, eval=T, include=F, warning=F}
gbk2 <- gbkData[,c("Strand", "Locus_tag")]
meanStr2 <- left_join(meanStr,gbk2,by=c("Tr"="Locus_tag"))
meanStr2 <- meanStr2[,c(colnames(meanStr2)[1:16],"Tr", "Strand")]
meanStr3 <- melt(meanStr2)

LMgc <- cor.test(meanStr$LM, meanStr$ratioGC, method="spearman")$estimate
LMgcp <- cor.test(meanStr$LM, meanStr$ratioGC, method="spearman")$p.value
LMgcp <- ifelse(LMgcp==0, "< 2.2e-16", as.character(format(LMgcp,digit=2)))
```

```
LMgc12 <- cor.test(meanStr$LM, meanStr$ratioGC_12pos, method="spearman")$estimate
LMgc12p <- cor.test(meanStr$LM, meanStr$ratioGC_12pos, method="spearman")$p.value
LMgc12p <- ifelse(LMgc12p==0, "< 2.2e-16", as.character(format(LMgc12p,digit=2)))
LMgc3 <- cor.test(meanStr$LM, meanStr$ratioGC_3pos, method="spearman")$estimate
LMgc3p <- cor.test(meanStr$LM, meanStr$ratioGC_3pos, method="spearman")$p.value
LMgc3p <- ifelse(LMgc3p==0, "< 2.2e-16", as.character(format(LMgc3p,digit=2)))

SAgc <- cor.test(meanStr$SA, meanStr$ratioGC, method="spearman")$estimate
SAgcp <- cor.test(meanStr$SA, meanStr$ratioGC, method="spearman")$p.value
SAgcp <- ifelse(SAgcp==0, "< 2.2e-16", as.character(format(SAgcp,digit=2)))
SAgc12 <- cor.test(meanStr$SA, meanStr$ratioGC_12pos, method="spearman")$estimate
SAgc12p <- cor.test(meanStr$SA, meanStr$ratioGC_12pos, method="spearman")$p.value
SAgc12p <- ifelse(SAgc12p==0, "< 2.2e-16", as.character(format(SAgc12p,digit=2)))
SAgc3 <- cor.test(meanStr$SA, meanStr$ratioGC_3pos, method="spearman")$estimate
SAgc3p <- cor.test(meanStr$SA, meanStr$ratioGC_3pos, method="spearman")$p.value
SAgc3p <- ifelse(SAgc3p==0, "< 2.2e-16", as.character(format(SAgc3p,digit=2)))

WLgc <- cor.test(meanStr$WL, meanStr$ratioGC, method="spearman")$estimate
WLgcp <- cor.test(meanStr$WL, meanStr$ratioGC, method="spearman")$p.value
WLgcp <- ifelse(WLgcp==0, "< 2.2e-16", as.character(format(WLgcp,digit=2)))
WLgc12 <- cor.test(meanStr$WL, meanStr$ratioGC_12pos, method="spearman")$estimate
WLgc12p <- cor.test(meanStr$WL, meanStr$ratioGC_12pos, method="spearman")$p.value
WLgc12p <- ifelse(WLgc12p==0, "< 2.2e-16", as.character(format(WLgc12p,digit=2)))
WLgc3 <- cor.test(meanStr$WL, meanStr$ratioGC_3pos, method="spearman")$estimate
WLgc3p <- cor.test(meanStr$WL, meanStr$ratioGC_3pos, method="spearman")$p.value
WLgc3p <- ifelse(WLgc3p==0, "< 2.2e-16", as.character(format(WLgc3p,digit=2)))

WRgc <- cor.test(meanStr$WR, meanStr$ratioGC, method="spearman")$estimate
WRgcp <- cor.test(meanStr$WR, meanStr$ratioGC, method="spearman")$p.value
WRgcp <- ifelse(WRgcp==0, "< 2.2e-16", as.character(format(WRgcp,digit=2)))
WRgc12 <- cor.test(meanStr$WR, meanStr$ratioGC_12pos, method="spearman")$estimate
WRgc12p <- cor.test(meanStr$WR, meanStr$ratioGC_12pos, method="spearman")$p.value
WRgc12p <- ifelse(WRgc12p==0, "< 2.2e-16", as.character(format(WRgc12p,digit=2)))
WRgc3 <- cor.test(meanStr$WR, meanStr$ratioGC_3pos, method="spearman")$estimate
WRgc3p <- cor.test(meanStr$WR, meanStr$ratioGC_3pos, method="spearman")$p.value
WRgc3p <- ifelse(WRgc3p==0, "< 2.2e-16", as.character(format(WRgc3p,digit=2)))
figNb <- figNb+1
```

\newpage

Correlations between GC content (global, first two codon positions, third codon position) across all conditions:

Correlation	*Spearman's corr. coeff.*	*p-value*
LM ~ GC-content	`r round(LMgc,2)`	`r LMgcp`
LM ~ GC-content (1st&2d pos.)	`r round(LMgc12,2)`	`r LMgc12p`
LM ~ GC-content (3d pos.)	`r round(LMgc3,2)`	`r LMgc3p`
SA ~ GC-content	`r round(SAgc,2)`	`r SAgcp`
SA ~ GC-content (1st&2d pos.)	`r round(SAgc12,2)`	`r SAgc12p`
SA ~ GC-content (3d pos.)	`r round(SAgc3,2)`	`r SAgc3p`
WL ~ GC-content	`r round(WLgc,2)`	`r WLgcp`
WL ~ GC-content (1st&2d pos.)	`r round(WLgc12,2)`	`r WLgc12p`
WL ~ GC-content (3d pos.)	`r round(WLgc3,2)`	`r WLgc3p`
WR ~ GC-content	`r round(WRgc,2)`	`r WRgcp`
WR ~ GC-content (1st&2d pos.)	`r round(WRgc12,2)`	`r WRgc12p`
WR ~ GC-content (3d pos.)	`r round(WRgc3,2)`	`r WRgc3p`

######*Table S`r figNb `. Results of correlation tests (Spearman's coefficient) between GC content (global, first two positions and third position) and log of RPKM values for all genes for all experimental conditions separately.*

After that, we also tried to see if a difference between leading and lagging strand was noticeable. This does not seem to be the case (maybe a slightly higher level of expression for genes on leading ("+") strand).

```{r barplot5, echo=FALSE, eval=T, include=T, fig.height=5.5,fig.width=10, fig.alig='center', warning=F}
```

```
figNb <- figNb+1
meanStr3$Var <- substr(meanStr3$variable,1,2)
fillCol <- c( "orangered2", "dodgerblue3", "forestgreen")
meanStr3$logVal <- log(meanStr3$value)

meanStr3b <- meanStr3[which(meanStr3$Strand=="+" |meanStr3$Strand=="-"), ]

maintit = "Gene expression and strand"
p <-ggplot(meanStr3b, aes(x=Var, y=logVal, fill=Strand)) +
    geom_boxplot()+
    ggtitle(maintit)+
    scale_y_continuous("log(RPKM)")+
    #scale_colour_discrete(name  ="Experimental conditions")+
    scale_x_discrete("Experimental conditions")+
  scale_fill_manual(name="Gene associated with", values=fillCol)+
#    stat_summary(fun.y=mean, geom="line", aes(group=Motility_type, colour=fillCol))  +
    theme(axis.title.y = element_text(face="bold", colour="#990000", size=15),
          axis.text.y = element_text(colour="black"),
          axis.title.x = element_text(face="bold", colour="#990000", size=15),
          axis.text.x  = element_text(angle=90, vjust=0.5, size=10),
          plot.title =  element_text(colour="darkslateblue", size=20),
          legend.text=element_text(size=15),
          legend.title=element_text(size=15, face="bold"),
          panel.grid.minor.y=element_blank(),panel.grid.major.y=element_blank())+
    guides(colour=FALSE)

plot(p)

```
```

######*Figure S`r figNb `. For each condition, plot showing log of RPKM values conditioned by the strand on which the gene is located (this information was not available for all, but for most of the genes).*

Multivariate analyses

We also tried to use multivariate tools to visualize the contribution of "structural" parameters to variation of gene expression. We first used a symmetrical method (PCA). Then, we tried an asymmetrical method, redundancy analysis (RDA), that performs a multivariate multiple linear regression followed by PCA `r Citep(bib, "Legendre2011")`. We still doubt that this method is appropriate for RNA-seq data. Only a small fraction of expression variation seems to be explained by "structural" parameters (see percents along the axis).

```{r pca2, echo=FALSE, eval=T, include=T, fig.height=3.3,fig.width=12, fig.alig='center', warning=F}
figNb <- figNb+1
par(mfrow=c(1,1))
cleanplot.pca(gene.pca2, mycol=mycol)
foo <- dev.off()
```

######*Figure S`r figNb `. PCA plots for all genes and "structural parameters". Left: scaling 1 (angles are meaningless), right: scaling 2 (distances are meaningless).*

```{r rda, echo=FALSE, eval=T, include=T, fig.height=3.5,fig.width=4, fig.alig='center', warning=F}
## RDA
figNb <- figNb+1
rdaFct(meanAndStat[,1:4], meanAndStat[,5:7])

```

######*Figure S`r figNb `. RDA plot of expression values regressed against "structural parameters".*

KEGG pathways and GO categories

Here, we retrieved the KEGG pathways of *Pseudomonas fluorescens* Pf5 available on the KEGG database. In the first step, we "matched" the *Pseudomonas fluorescens* Pf5 genes with the ones of our *Pseudomonas* S5 (with BLAT, see Perl script at this end the document; although this is probably not the most optimal solution, it is fast and presumably convenient for explanatory purposes). This allowed us to associate most genes of *Pseudomonas* S5 with a pathway.

For the gene ontology (GO) categories, we did something "on the fly" as another group was already
```

working with the time-consuming BLAST2GO.
We retrieved the GO categories for *Pseudomonas aeruginosa* PAO1 genes, as we did not find GO data for the *Pseudomonas fluorescens* Pf5 on the Pseudomonas database (www.pseudomonas.com). We found the orthologous pairs of genes between *Pseudomonas aeruginosa* PAO1 and *Pseudomonas fluorescens* Pf5 genes. Thus we could retrieve GO of a large number of *Pseudomonas fluorescens* Pf5 genes.
Then, we could associate GO to our *Pseudomonas* S5 genes as we had already linked *Pseudomonas protegens* Pf5 and *Pseudomonas* S5 genes (as described just here above).

\newpage

GO categories

We brought together the categories associated with flagella or type IV pili under a "motility" category. We observed that motility is clearly not the most represented category.

```{r goCat, echo=FALSE, eval=T, include=T, fig.height=24,fig.width=24, fig.alig='center', warning=F}
figNb <- figNb+1
# LOAD ORTHOLOGY DATA
orthoPf5_PA <- read.csv("../data/orthoPA_PFL.csv", sep="\t", header=F)
colnames(orthoPf5_PA) <- c("PA_genes", "PFL_genes")
# nrow(orthoPf5_PA)
#3759
goPA <- read.csv("../data/GO.csv")
goPA2 <- goPA[,c("Locus.Tag", "GO.Term")]
# nrow(goPA)
# 16465
Pf5GO <- left_join(orthoPf5_PA,goPA, by=c("PA_genes"="Locus.Tag"))
S5Pf5 <- read.csv("../data/Pf5/S5vsPf5.txt", sep="\t")
S5_GO <- left_join(S5Pf5, Pf5GO, by=c("Pf5"="PFL_genes"))
# remove other columns and remove NA rows
S5_GO_short <- S5_GO[,c("S5_genome_id", "GO.Term")] %>% na.omit
    # COLNAMES: "S5_genome_id" "GO.Term"

S5_GO_mot <- S5_GO_short

S5_GO_mot$GO.Term <- as.character(S5_GO_mot$GO.Term)

# group all GO linked with motility in 1 category
S5_GO_mot$GO.Term[grep("motility|flagella|type IV pilus|swarming",S5_GO_mot$GO.Term)] <- "motility"

p <- goHisto(dt, annot, S5_GO_mot, "LM", "SA", 75, withMot=T)
q <- goHisto(dt, annot, S5_GO_mot, "LM", "WL", 75, withMot=T)
r <- goHisto(dt, annot, S5_GO_mot, "LM", "WR", 75, withMot=T)
s <- goHisto(dt, annot, S5_GO_mot, "SA", "WL", 75, withMot=T)
t <- goHisto(dt, annot, S5_GO_mot, "SA", "WR", 75, withMot=T)
u <- goHisto(dt, annot, S5_GO_mot, "WL", "WR", 75, withMot=T)

multiplot(p, q,r,s,t,u, cols=2)

# without motility
# goHisto(dt, annot, S5_GO_short, "SA", "WR", 75) %>% plot

```

######*Figure S`r figNb `. Barplots showing for each pairs of condition to which GO category the up- and downregulated genes belong. Threshold: 75 occurences of the GO category (motility added independently of the number of occurences, as explained in the text).*

KEGG pathways

```{r kegg, echo=FALSE, eval=T, include=T, fig.height=29,fig.width=24, fig.alig='center', warning=F}
figNb <- figNb+1
# match S5_genome ID with Pf5 id
S5Pf5 <- read.csv("../data/Pf5/S5vsPf5.txt", sep="\t")
Pf5genes_kegg <- read.csv("../data/Pf5/kegg_pathway_gene_pfl.txt",
                          header=F,sep="\t")
colnames(Pf5genes_kegg) <- c("path_id", "gene_id")
Pf5genes_kegg$gene_id <- gsub("^pfl:", "", Pf5genes_kegg$gene_id)
kegg_path <- read.csv("../data/Pf5/kegg_pathway_id_pfl.txt",
                      header=F, sep="\t")
colnames(kegg_path) <- c("path_id", "path_name")
kegg_path$path_name <- gsub(" - Pseudomonas protegens Pf-5$", "", kegg_path$path_name)
```

```
path_pf5 <- full_join(Pf5genes_kegg, kegg_path, by="path_id")
path_S5 <- full_join(S5Pf5, path_pf5, by=c("Pf5"="gene_id"))

annot <- read.csv("../data/annot_mot.csv", sep=",")
gbkData <- read.csv("../data/S5_gbk_short.csv", sep=",")
abd_fld <- "../data/abundances/"
dt <- getDGE(abd_fld)  # compute it once here
# cond1="LM";cond2="SA"
# threshold = 30

#***********************************************
##### PLOT 11:  KEGG histo
#***********************************************
# par(mfrow=c(3,2))
p <- keggHisto(dt, annot, path_S5, "LM", "SA", 20)
q <- keggHisto(dt, annot, path_S5, "LM", "WL", 20)
r <- keggHisto(dt, annot, path_S5, "LM", "WR", 20)
s <- keggHisto(dt, annot, path_S5, "SA", "WL", 20)
t <- keggHisto(dt, annot, path_S5, "SA", "WR", 20)
u <- keggHisto(dt, annot, path_S5, "WL", "WR", 20)
multiplot(p, q,r,s,t,u, cols=2)
# foo <- dev.off()

```

######*Figure S`r figNb `. For all pairs of comparisons, barplots showing to which KEGG pathway the down- and upregulated genes belong.*

\newpage

Further statistical tests

```{r tukey, echo=FALSE, eval=T, include=F, fig.height=4,fig.width=5, fig.alig='center', warning=F}
countsB <- getRawCounts(abd_fld) %>% as.data.frame
countsB$Tr <- rownames(counts)
counts_StrB <- left_join(counts, S5_stat, by=c("Tr" = "Seq_tag"))
counts_StrB[,1:16] <- myrpkm(counts_StrB[,1:16], counts_StrB$Length)
nF <- counts_StrB[,1:16]
meanD <- getMeanData(nF)
meanD$Tr <- counts_StrB$Tr
meltDf <- melt(meanD, by=meanD$Tr)
m1 <- lm(meltDf$value ~ meltDf$variable)
anova(m1)
m2 <- aov(m1)
posthoc <- TukeyHSD(x=m2, 'meltDf$variable', conf.level=0.95)
# phT <- posthoc$meltDf
# phT %<>% as.data.frame
# phT$Cond <- rownames(phT)
# rownames(phT) <- NULL
# phT <- phT[,c(5, 1:4)]
# colnames(phT) <- c("Conditions tested", "Diff.", "Lower", "Upper", "p-adj")
# kable(phT, digits=2)
```

```{r tukey2, echo=FALSE, eval=T, include=F, fig.height=4,fig.width=5, fig.alig='center', warning=F}
countsB <- getRawCounts(abd_fld) %>% as.data.frame
countsB$Tr <- rownames(counts)
counts_StrB <- left_join(counts, S5_stat, by=c("Tr" = "Seq_tag"))
counts_StrB[,1:16] <- myrpkm(counts_StrB[,1:16], counts_StrB$Length)
nF <- counts_StrB[,1:16]
meanD <- getMeanData(nF)
meanD$Tr <- counts_StrB$Tr

meanM <- left_join(annot[,c(1,3)], meanD, by=c("Gene_position"="Tr"))

meltM <- melt(meanM, by=meanM$Gene_position)


m3 <- lm(meltM$value ~ meltM$variable*meltM$Motility_type)
anova(m3)
m4 <- aov(m3)
```

```
posthoc <- TukeyHSD(x=m4, c('meltM$Motility_type', 'meltM$variable'),conf.level=0.95)

```

```{r t1, echo=FALSE, eval=T, include=T, fig.height=5,fig.width=7, fig.alig='center', warning=F}
figNb <- figNb+1
im <- interactionMeans(m3)
plot(im)
```


######*Figure S`r figNb `. Interaction plots.*


```{r t2, echo=FALSE, eval=T, include=T, fig.height=4,fig.width=5, fig.alig='center', warning=F}
figNb <- figNb+1
tint <- testInteractions(m3, adjustment="BH") %>% as.data.frame
tint <- tint[-nrow(tint),]
kable(tint, digits=2)
```


######*Table S`r figNb `. Test contrasts of factor interactions (experimental conditions and motility type).*

Venn diagram

Here, we also tried to draw Venn diagram to help us visualize differential expression.
Our trial was with liquid medium as reference. This was nonetheless not very conclusive.

```{r venn, echo=FALSE, eval=T, include=T, fig.height=3,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
lmsa <- dataLMSA[,c("logFC", "Transcript")]
colnames(lmsa)[1] %<>% paste0(., ".LMSA")
lmwl <- dataLMWL[,c("logFC", "Transcript")]
colnames(lmwl)[1] %<>% paste0(., ".LMWL")
lmwr <- dataLMWR[,c("logFC", "Transcript")]
colnames(lmwr)[1] %<>% paste0(., ".LMWR")

allLM <- full_join(lmsa, lmwl, by="Transcript") %>%
  full_join(., lmwr, by="Transcript")


allLM$upSA <- as.numeric(allLM$logFC.LMSA>0)
allLM$upWL <- as.numeric(allLM$logFC.LMWL>0)
allLM$upWR <- as.numeric(allLM$logFC.LMWR>0)

allLM$downSA <- as.numeric(allLM$logFC.LMSA<0)
allLM$downWL <- as.numeric(allLM$logFC.LMWL<0)
allLM$downWR <- as.numeric(allLM$logFC.LMWR<0)


upSAname <- allLM$Transcript[which(allLM$upSA==1)]
upWLname <- allLM$Transcript[which(allLM$upWL==1)]
upWRname <- allLM$Transcript[which(allLM$upWR==1)]

doSAname <- allLM$Transcript[which(allLM$downSA==1)]
doWLname <- allLM$Transcript[which(allLM$downWL==1)]
doWRname <- allLM$Transcript[which(allLM$downWR==1)]


allLM[is.na(allLM)] <- 0

upCol <- c("chartreuse2", "darkolivegreen1", "forestgreen")
vp <- venn.diagram(list(SA=upSAname,WL=upWLname,WR=upWRname), fill=upCol,
                   alpha = 0.3, filename = NULL, height = 3000, width = 3000,
                   main="Upregulated genes (ref: LM)", main.cex=1.5)#, main.fontfamily=1);
grid.newpage()
grid.draw(vp)
foo <- dev.off()
```

```{r venn2, echo=FALSE, eval=T, include=T, fig.height=3,fig.width=4.5, fig.alig='center', warning=F}
```

```
downCol <- c("darksalmon", "brown1", "coral")
vd <- venn.diagram(list(SA=doSAname,WL=doWLname,WR=doWRname), fill=downCol,
                   height = 3000, width = 3000,
                   alpha = 0.3, filename = NULL, main="Downregulated genes (ref: LM)",
                   main.cex=1.5)#, main.fontfamily=1);
# grid.newpage()
grid.draw(vd)
foo <- dev.off()
```

######*Figure S`r figNb `. Example of Venn diagram for up- and downregulated genes. The number indicates the number of motility-associated genes up- (top) and downregulated (bottom) in the indicated condition when compared to LM.*

\newpage

Genome plots

Finally, we tried to visualize the clusters of motility-associated genes along the *Pseudomonas S5* genome with tools of the genoPlotR package `r Citep(bib, "Lionel2010")`.
We compared their position in *Pseudomonas S5* genome and *Pseudomonas fluorescens* Pf5 genome (retrieved from http://www.pseudomonas.com and then processed in the terminal to obtain the optimal data shape; 58 motility-associated genes in common based on the gene name).
Globally, the order of these genes is conserved for a large part of the motility-associated genes.

```{r genePlot, echo=FALSE, eval=T, include=F, warning=F}
numStrand <- function(x){
  if(is.na(x)){
    y = -1
  }
  else if(x=="+"){
    y = 1
  } else{
    y = -1
  }
  y
}
grid.newpage()
# foo <- dev.off()
par(mfrow=c(1,1))

pf5_genes <- read.csv("../data/Pf5/genes_Pf5.gtf", header=F, sep="\t")
colnames(pf5_genes) <- c("start_pf", "end_pf", "strand_pf", "gene_id_pf", "fonction_pf")

pf5_genes$fonc_short_pf <- gsub('(^.+) ,.+$', '\\1', pf5_genes$fonction_pf)

pf5_mot <- pf5_genes[which(pf5_genes$fonc_short_pf %in% annot$Gene_name),]

ps5_mot <- left_join(annot, gbkData, by=c("Gene_position"="Locus_tag"))
colnames(ps5_mot) %<>% paste0(., "_ps")

P_mot <- left_join(pf5_mot, ps5_mot, by=c("fonc_short_pf"="Gene_name_ps"))
```

All anotated motility-associated genes

```{r genePlot1, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot, mt="")
```

######*Figure S`r figNb `. Genome plot for all motility-associated genes: comparison* Pseudomonas protegens*S5 and* Pseudomonas fluorescens *Pf5.*

Flagellum-associated genes

```{r genePlot2, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[which(P_mot$Motility_type_ps=="flagella"),], mt ="")
```

######*Figure S`r figNb `. Genome plot for flagellum-associated genes: comparison* Pseudomonas

protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot3, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("flg",P_mot$fonc_short_pf),], mt = "")
```

######*Figure S`r figNb `. Genome plot for* flg *family genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot4, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("flg",P_mot$fonc_short_pf)[1:7],], mt = "")
```

######*Figure S`r figNb `. Genome plot for* flg *family genes (close-up 1): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot5, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("flg",P_mot$fonc_short_pf)[8:12],],mt = "")
```

######*Figure S`r figNb `. Genome plot for* flg *family genes (close-up 2): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot6, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("fli",P_mot$fonc_short_pf),], mt = "")
```

######*Figure S`r figNb `. Genome plot for* fli *family genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot7, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("fli",P_mot$fonc_short_pf)[8:12],])
```

######*Figure S`r figNb `. Genome plot for* fli *family genes (close-up): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

Pilus-associated genes

```{r genePlot8, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[which(P_mot$Motility_type_ps=="pili"),], mt = "")
```

######*Figure S`r figNb `. Genome plot for pilus-associated genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot9, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("pil",P_mot$fonc_short_pf),], mt = "")
```

######*Figure S`r figNb `. Genome plot for* pil *family genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot10, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("pil",P_mot$fonc_short_pf)[8:10],], mt = "")
```

######*Figure S`r figNb `. Genome plot for* pil *family genes (close-up 1): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot11, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("pil",P_mot$fonc_short_pf)[11:12],], mt = "")
```

######*Figure S`r figNb `. Genome plot for* pil *family genes (close-up 2): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot12, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("pil",P_mot$fonc_short_pf)[13:15],], mt = "")
```

######*Figure S`r figNb `. Genome plot for* pil *family genes (close-up 3): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

MotA/MotB duplication ?

We also observed that the motor proteins of the flagellum (*motA* and *motB*) are duplicated in the *Pseudomonas* S5 genome that we sequenced. Interestingly, these genes have been reported to be present in two sets in other bacterial genome (*Pseudomonas aeruginosa*; Doyle et al. 2004)

```{r genePlot13, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("motA",P_mot$fonc_short_pf),], agl=45, mt="")
```

######*Figure S`r figNb `. Genome plot for* motA *genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

```{r genePlot14, echo=FALSE, eval=T, include=T, fig.height=2.5,fig.width=4.5, fig.alig='center', warning=F}
figNb <- figNb+1
geneMapMot(P_mot[grep("motB",P_mot$fonc_short_pf),], agl=270, mt = "")
```

######*Figure S`r figNb `. Genome plot for* motB *genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

References


```{r, results='asis', echo=FALSE}
PrintBibliography(bib,.opts=list(check.entries=FALSE,sorting="aynt", max.names=2))
```