RNA-seq analysis of *Pseudomonas* S5 genes associated with motility - Supplementary materials

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 3.3.0 (2016-05-03)). We used the following packages: edgeR (Robinson et al., 2009), phia (De Rosario-Martinez, 2015) and vegan (Oksanen et al., 2016) for the statistical analyses, genoPlotR (Lionel et al., Kultima, and Andersson, 2010), ggplot2 (Wickhamm, 2009), pheatmap (Kolde, 2015) and VennDiagram (Chen, 2016) for the graphics, dplyr (Wickham and Francois, 2015), knitr (Xie, 2013), magrittr (Bache and Wickham, 2014) and reshape2 (Wickham, 2007)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.

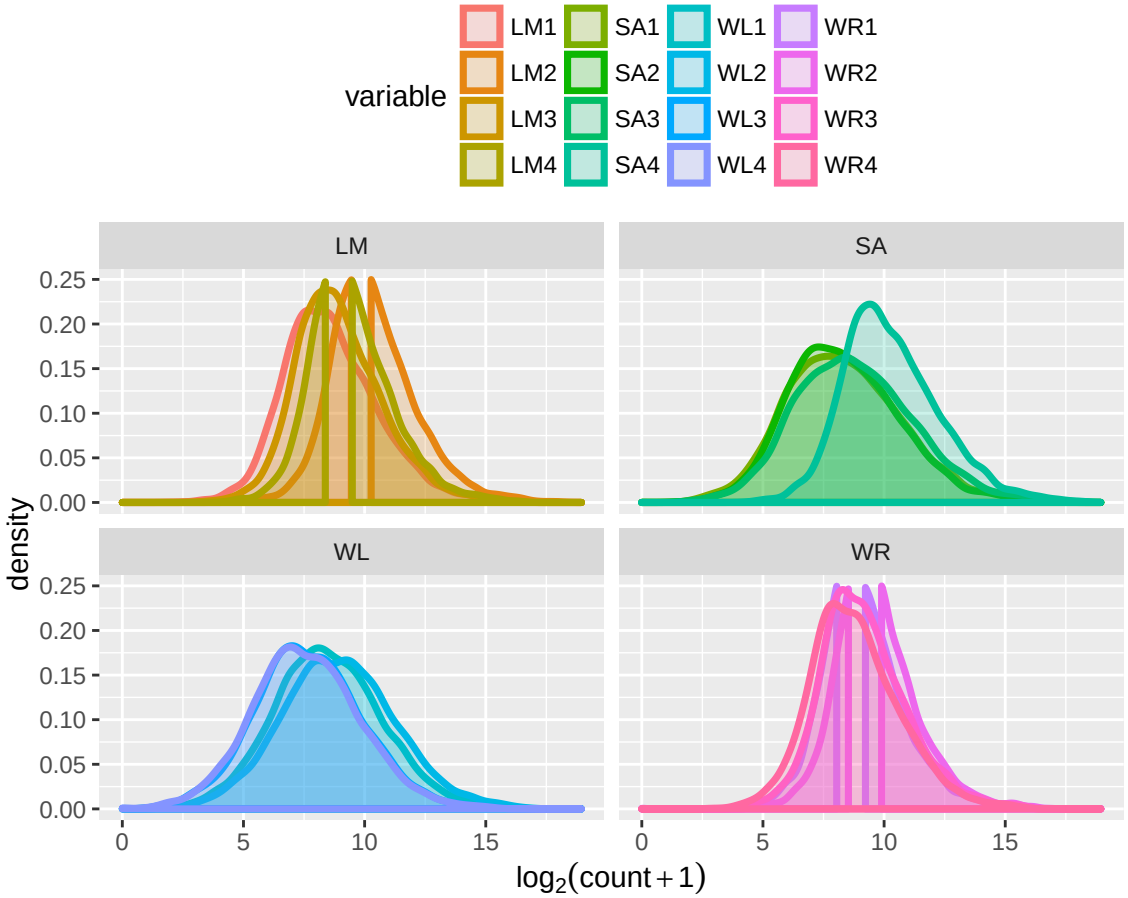


Figure S1. Density plot of log-normalized count data for the four experimental conditions.

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." (Chen et al., 2015). 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" (Diray-Arce et al., 2015).

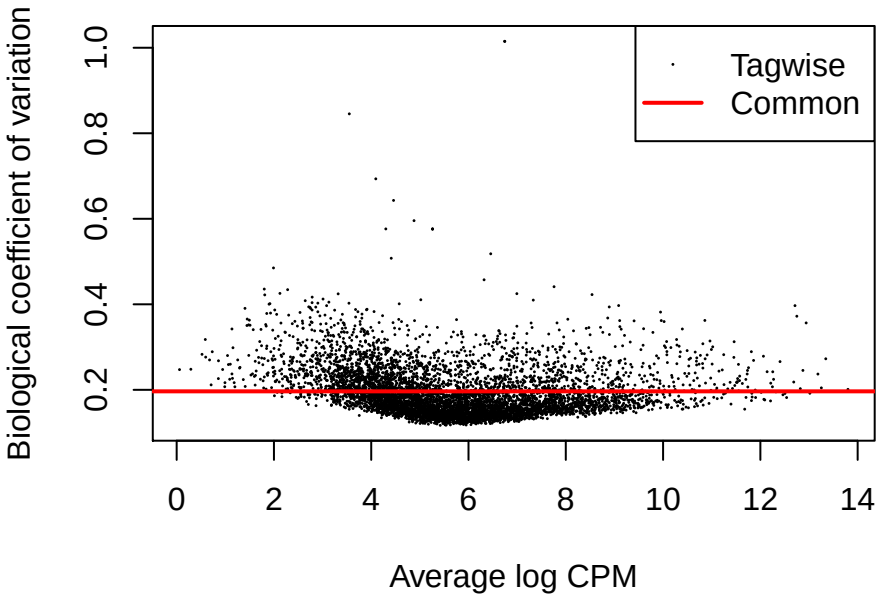


Figure S2. Plot of biological coefficient of variation.

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.

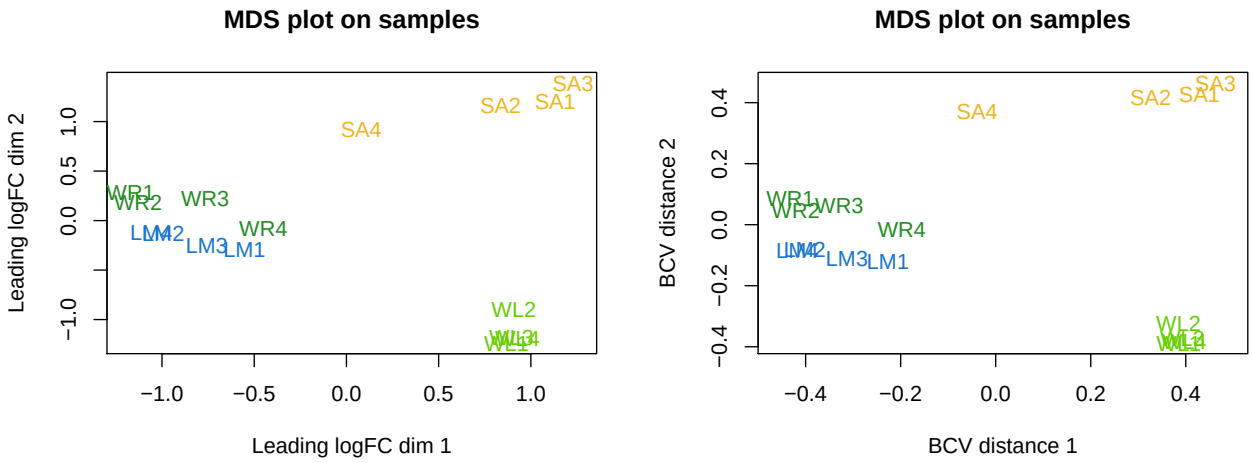


Figure S3. MDS plots for logFC (left) and BCV (right).

Multivariate analyses

PCA

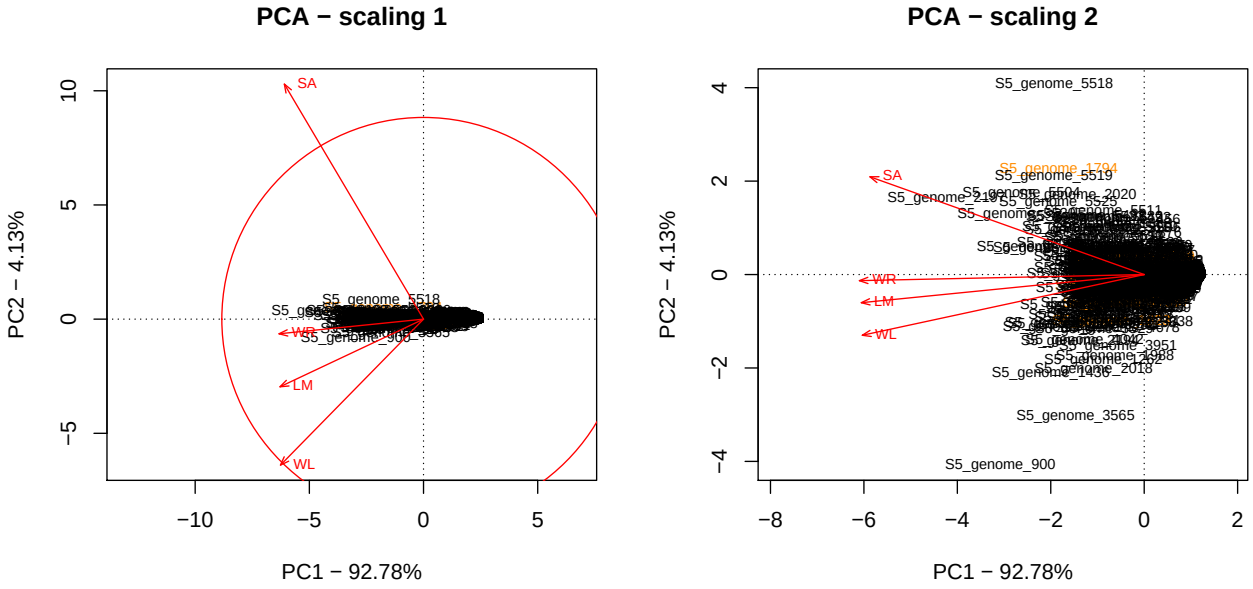


Figure S4. PCA plots for all genes and all conditions (mean data). Left: scaling 1 (angles are meaningless), right: scaling 2 (distances are meaningless).

PCA by condition (with coloured motility-associated genes)

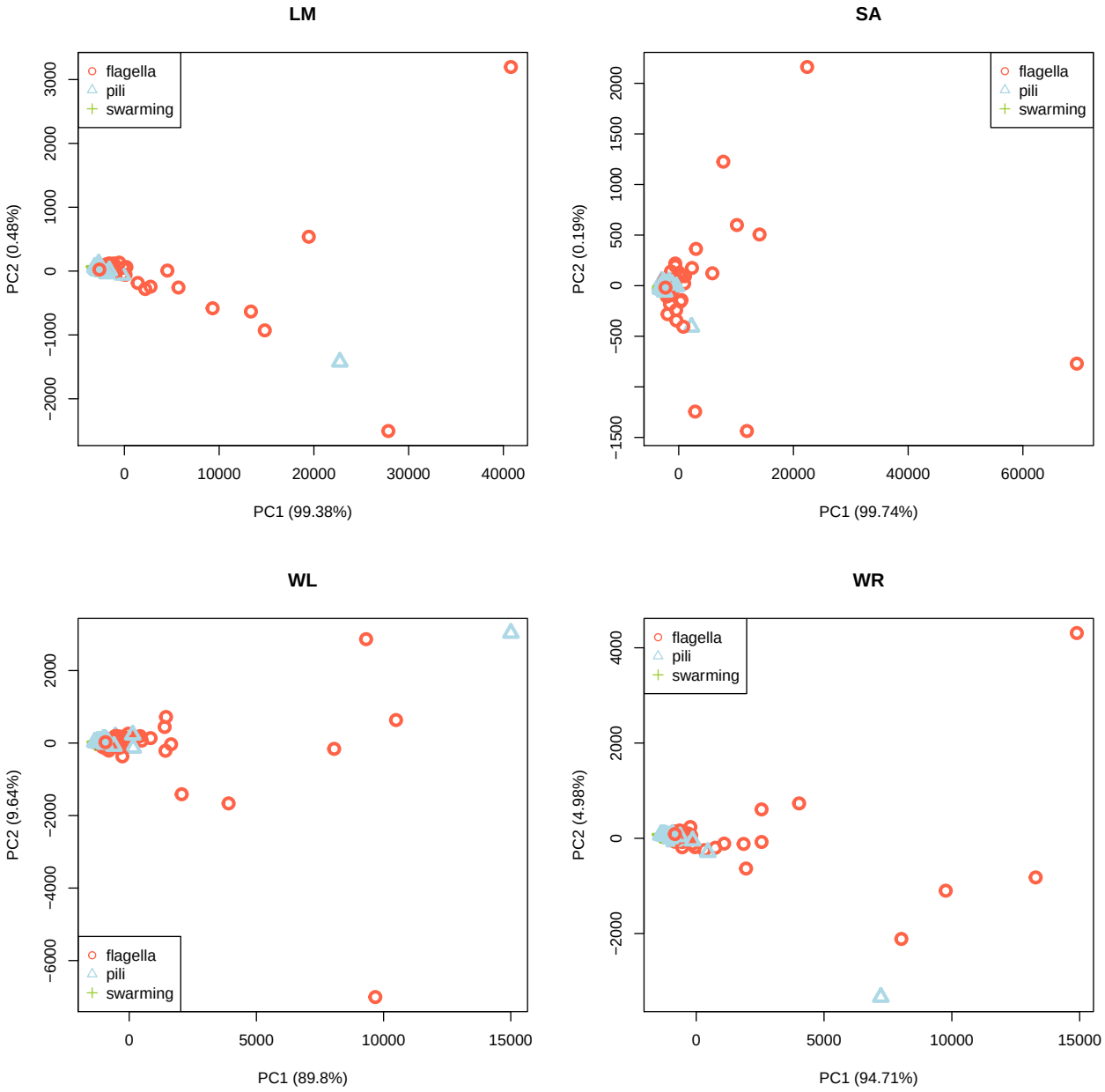


Figure S5. PCA plots for motility-associated genes only for all conditions separately.

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.

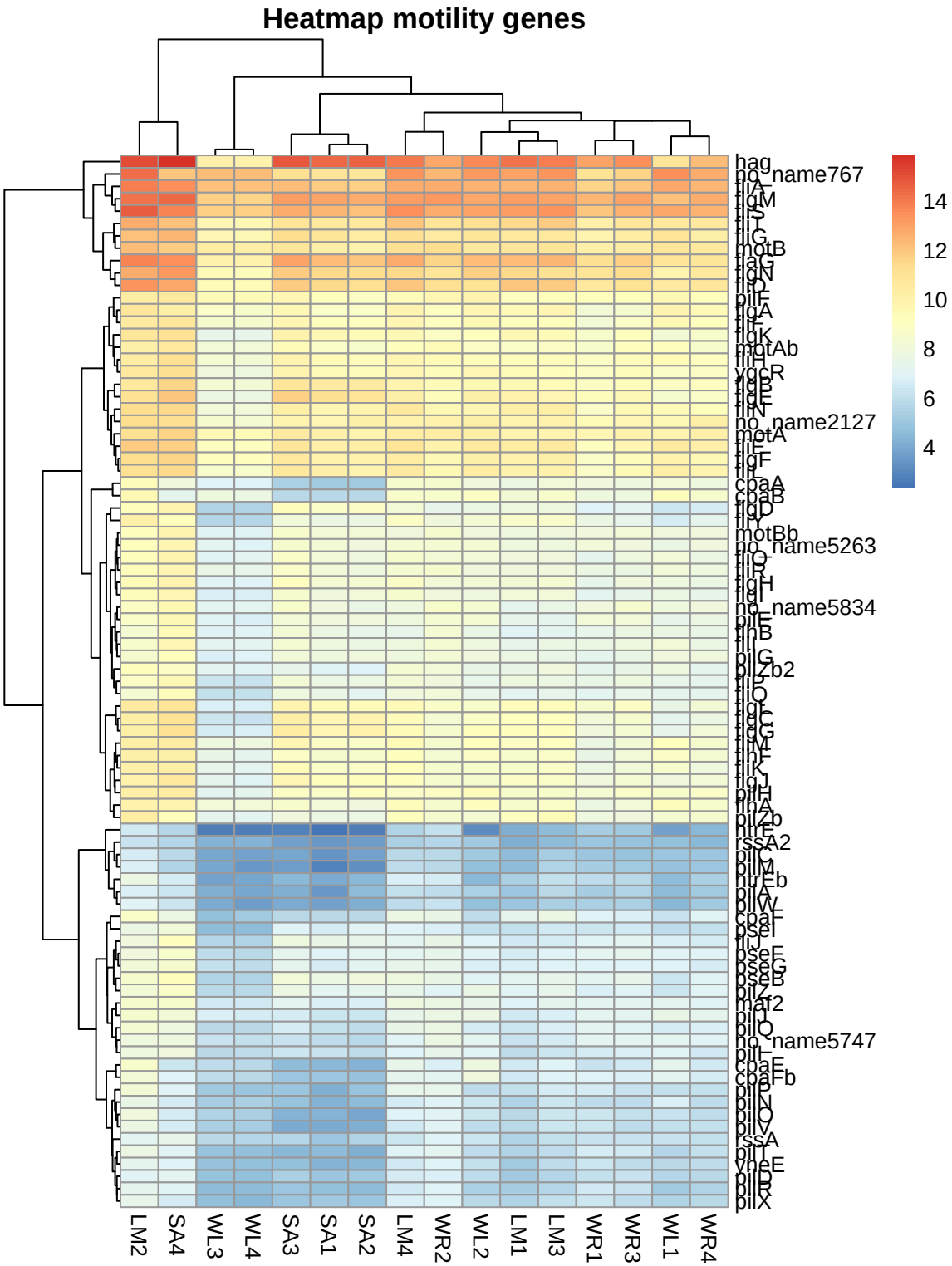


Figure S6. Heatmap for counts data for all replicates (motility-associated genes only).

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).

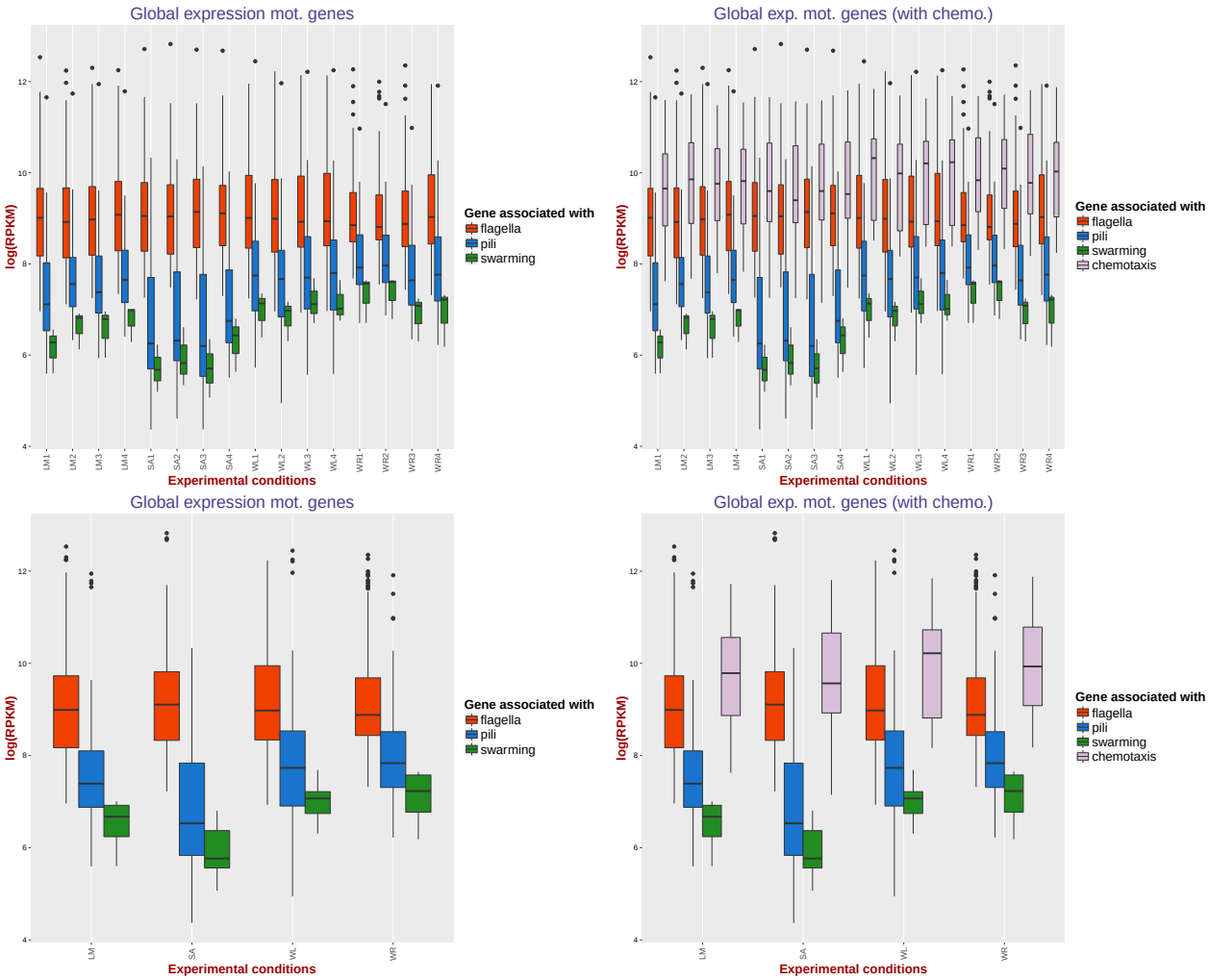


Figure S7. Boxplots of the log of RPKM values by motility type, for all replicates (top) or for the four conditions (bottom), without (left) and with (right) chemotaxis-associated genes.

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).

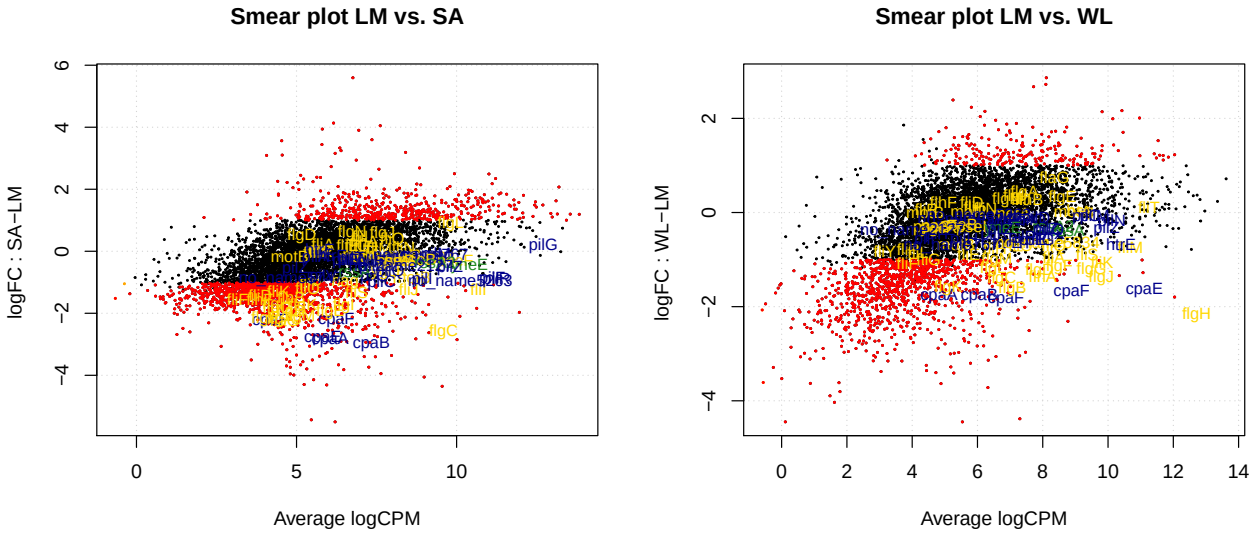
Histogram of p-values and plots logFC vs. logCPM (M vs. A)

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



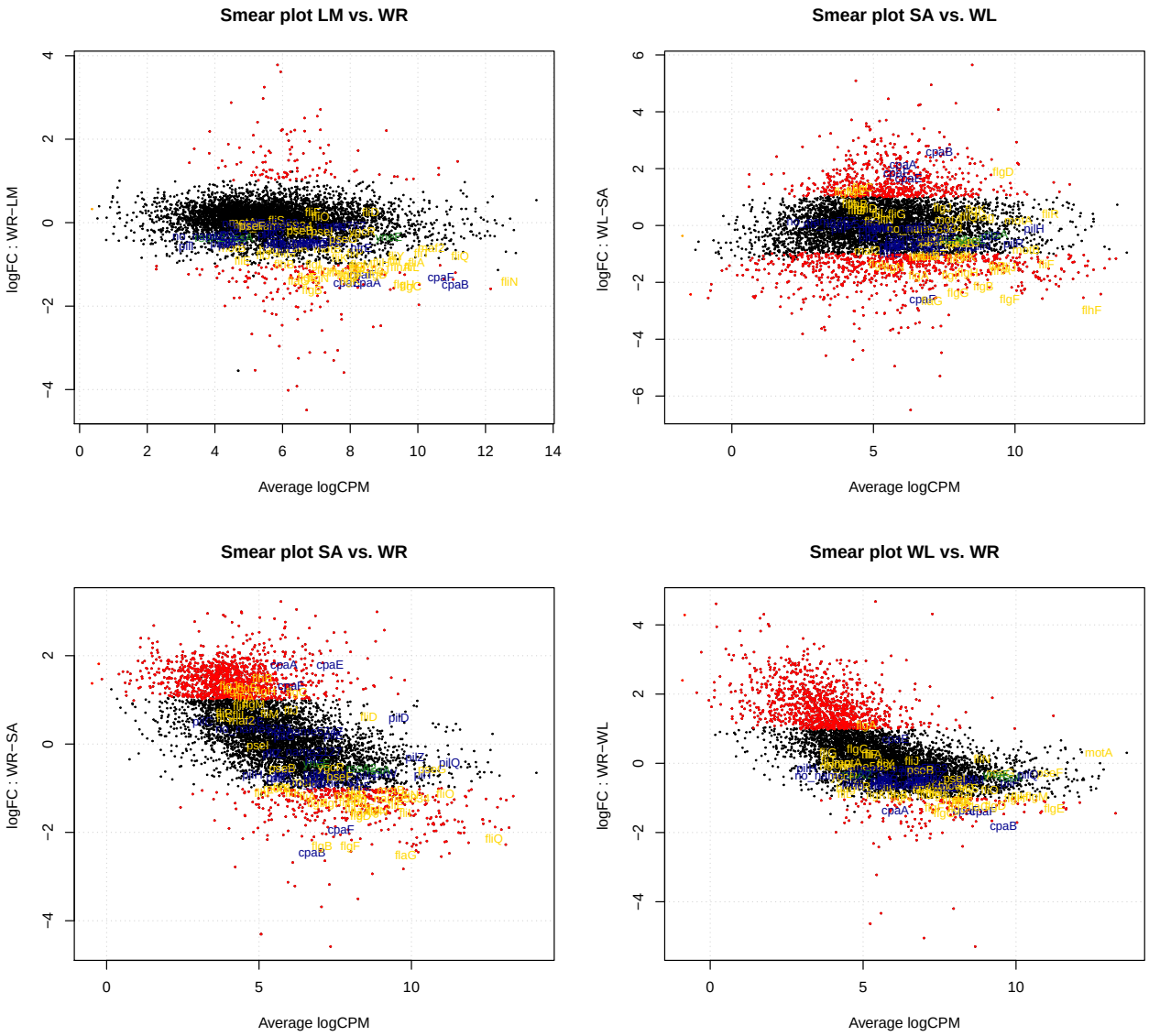


Figure S8. Smear plots for differential expression between all pairs. Genes with more than twofold change of expression shown in red. Motility-associated genes (labelled) shown in orange (flagellum-related), blue (pilus-related) and green (swarming-related). Please notice the different scales of the y-axis.

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.

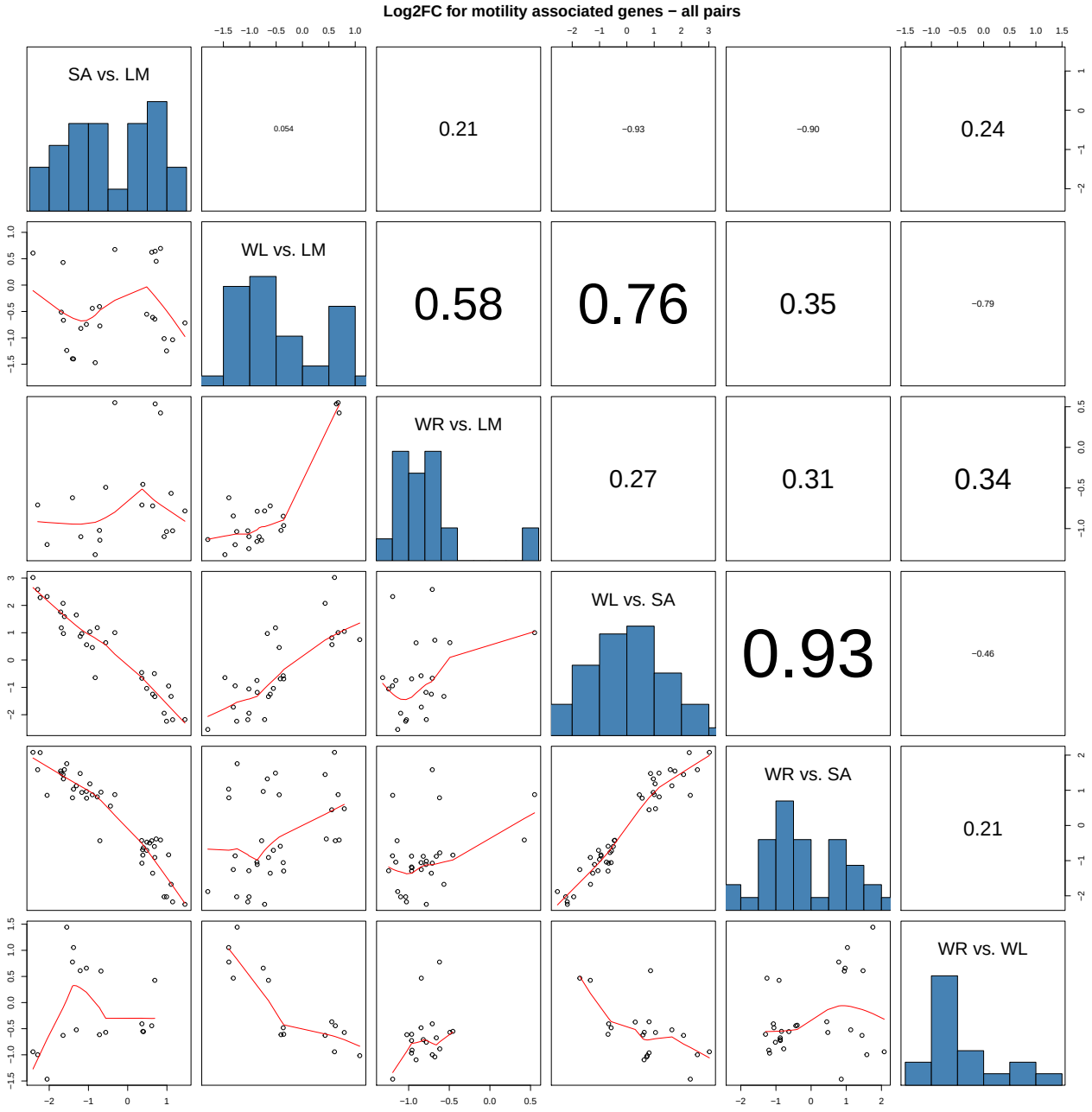


Figure S9. Scatterplot matrix: correlation between differential expression between pairs of conditions (motility-associated genes only.

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).

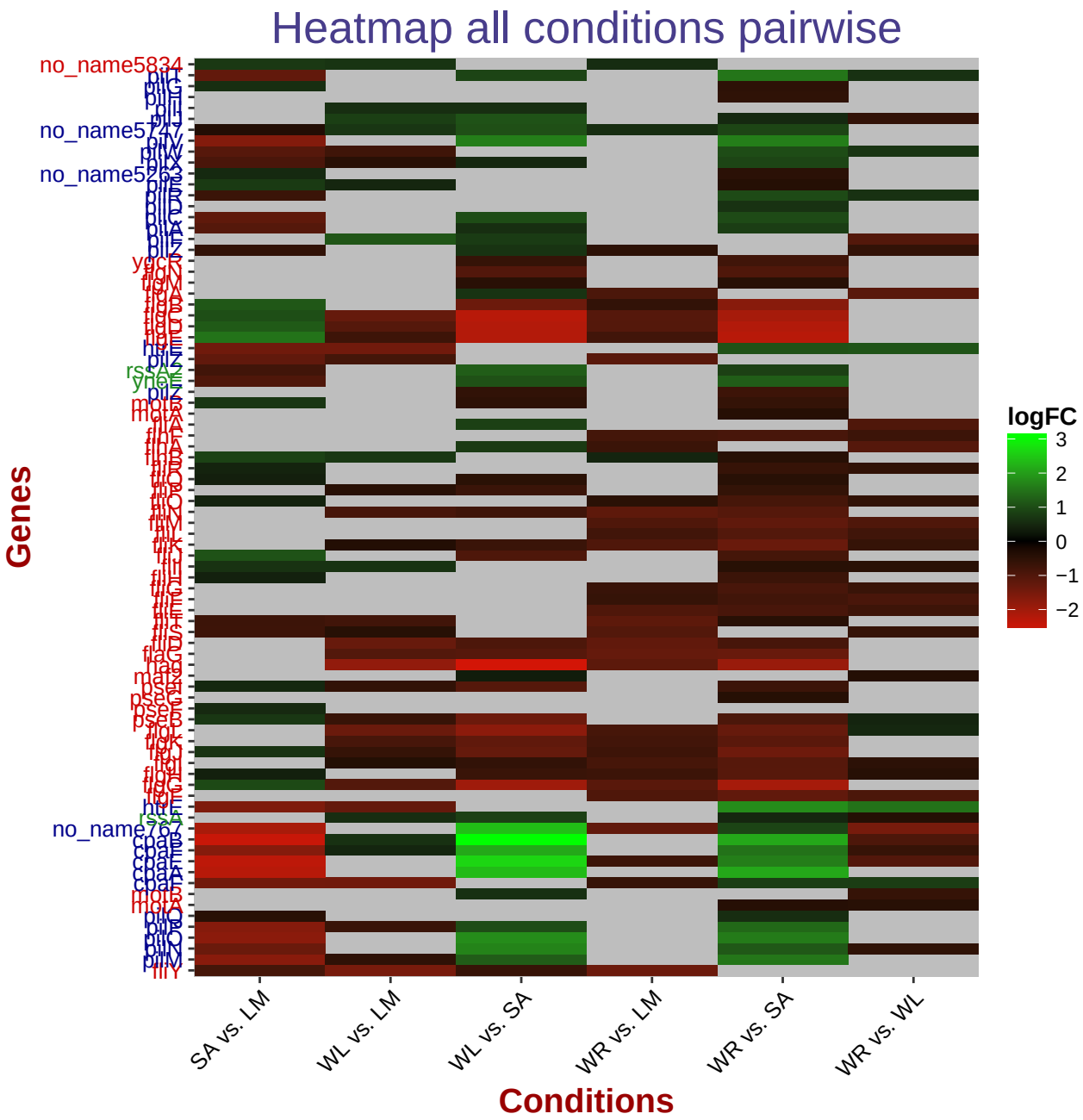
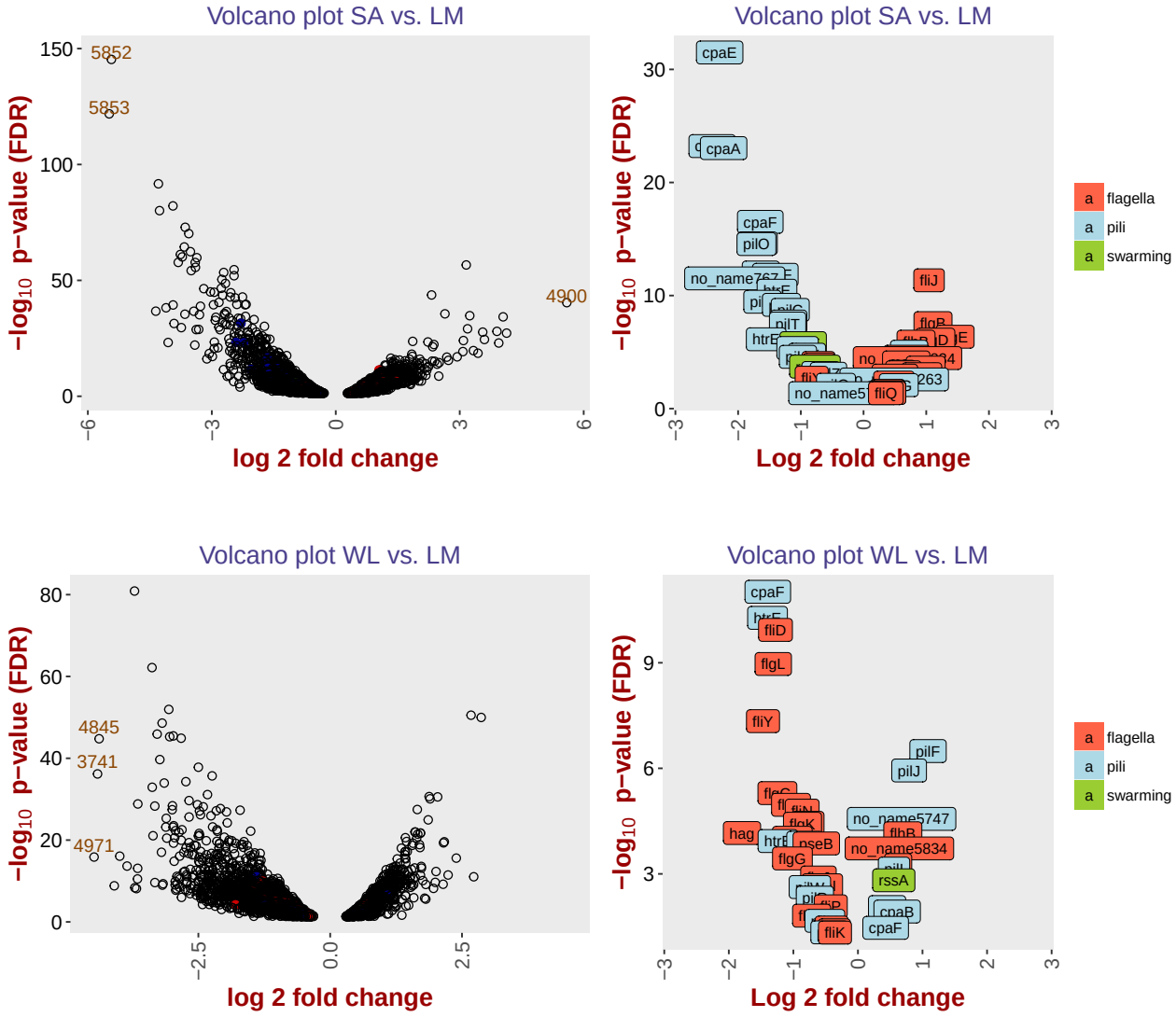


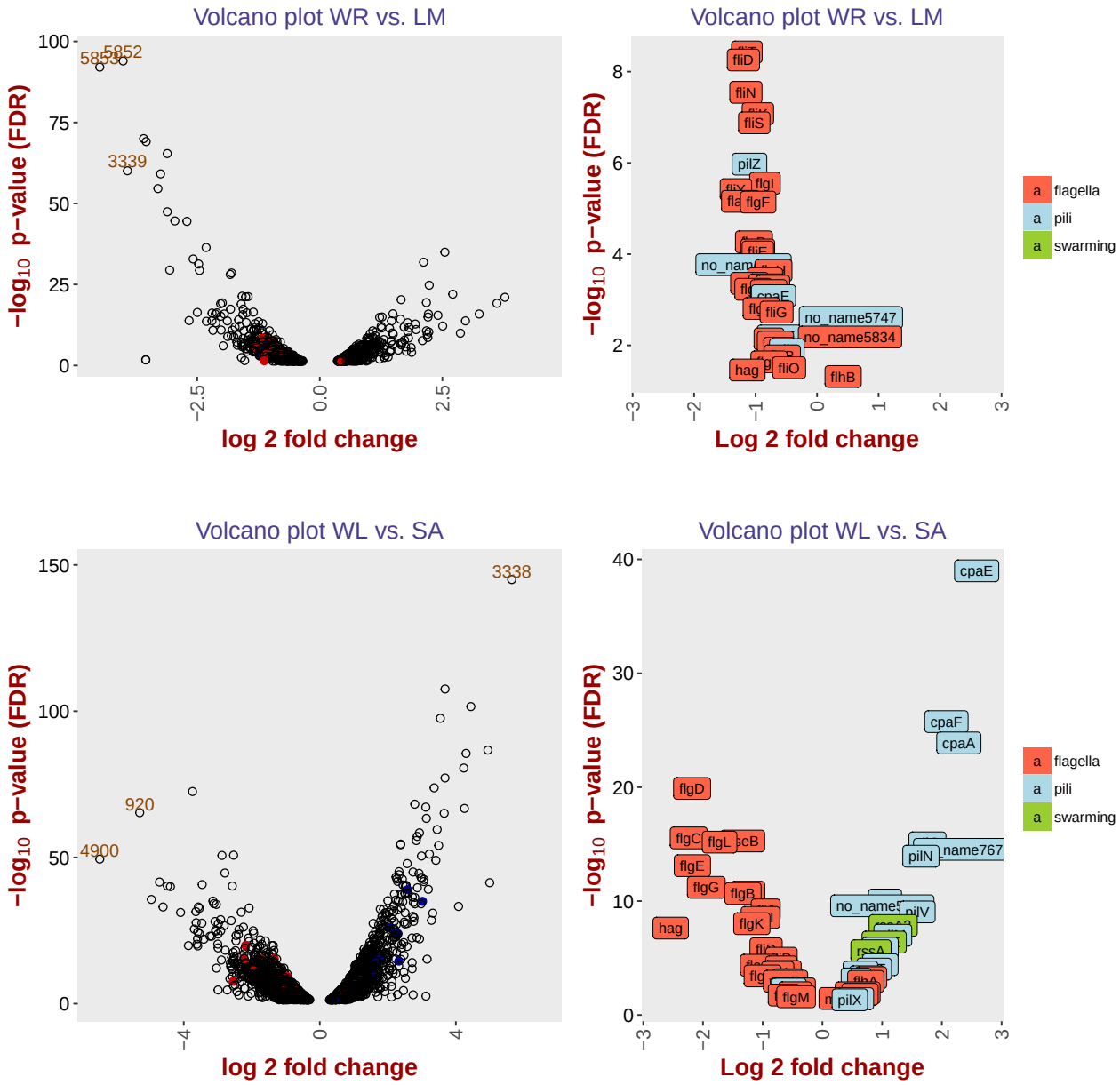
Figure S10. Heatmap of differential expression for all pairs of conditions (motility-associated genes only; only statistically significant genes with adjusted p-values < 0.05 are shown). Y-axis: genes are ordred according to their position on the chromosome.

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.




Volcano plot WR vs. LM
−log10 p−value (FDR)
log 2 fold change
5852
3339
Volcano plot WR vs. LM
−log10 p−value (FDR)
Log 2 fold change
fliD
fliN
fliS
pilZ
flgI
flgF
fliG
no_name5747
no_name5834
fliO
flhB
hag
a flagella
a pili
a swarming
Volcano plot WL vs. SA
−log10 p−value (FDR)
log 2 fold change
3338
920
4900
Volcano plot WL vs. SA
−log10 p−value (FDR)
Log 2 fold change
cpaE
cpaF
cpaA
flgD
flgE
flgG
flgK
hag
pilN
pilV
flgM
pilX
a flagella
a pili
a swarming

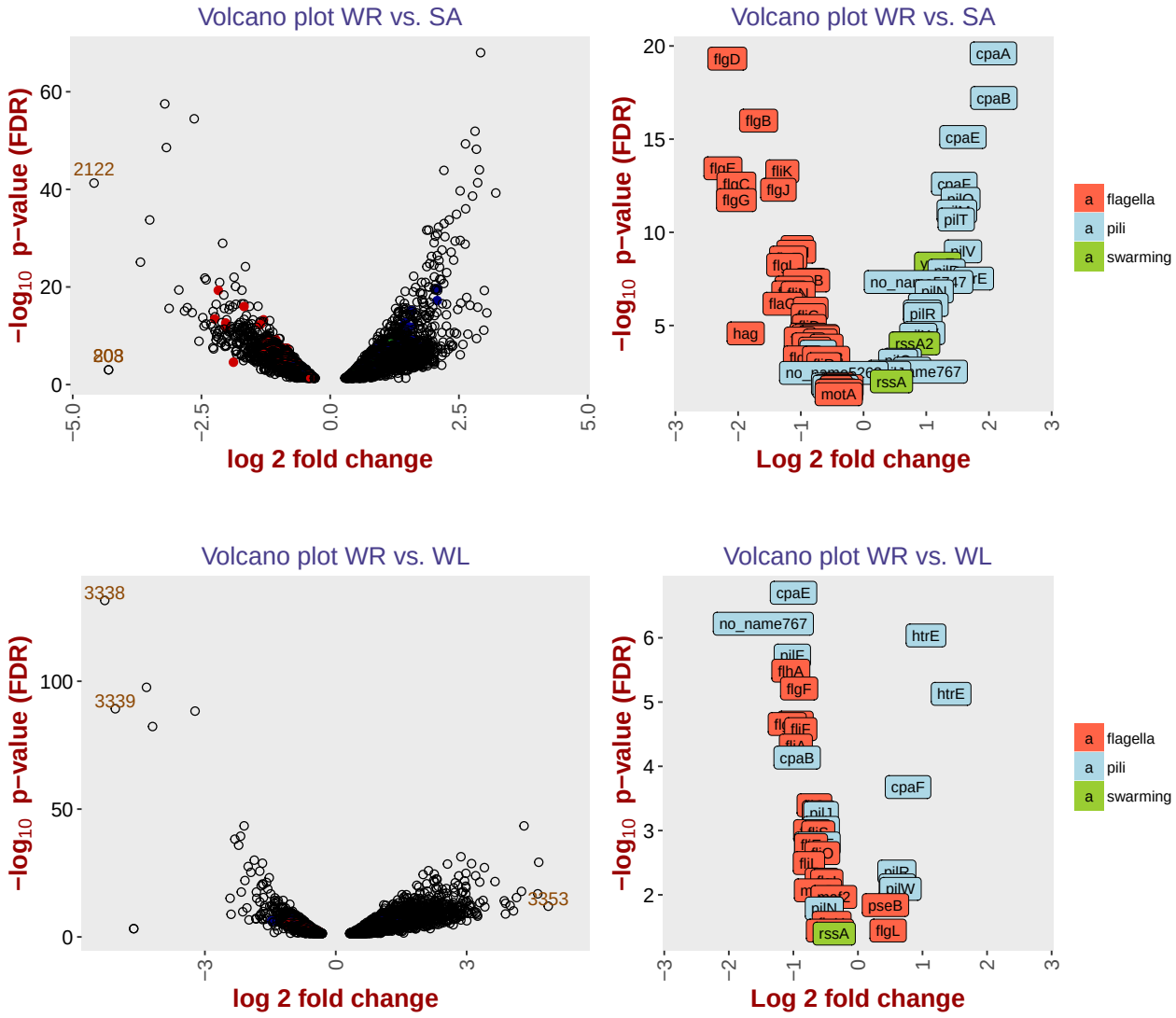


Figure S11. Volcano plots for all pairs of comparisons (left column: all genes differentially expressed in a statistically significant manner (FDR < 0.05); right column: only motility-associated genes).

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).

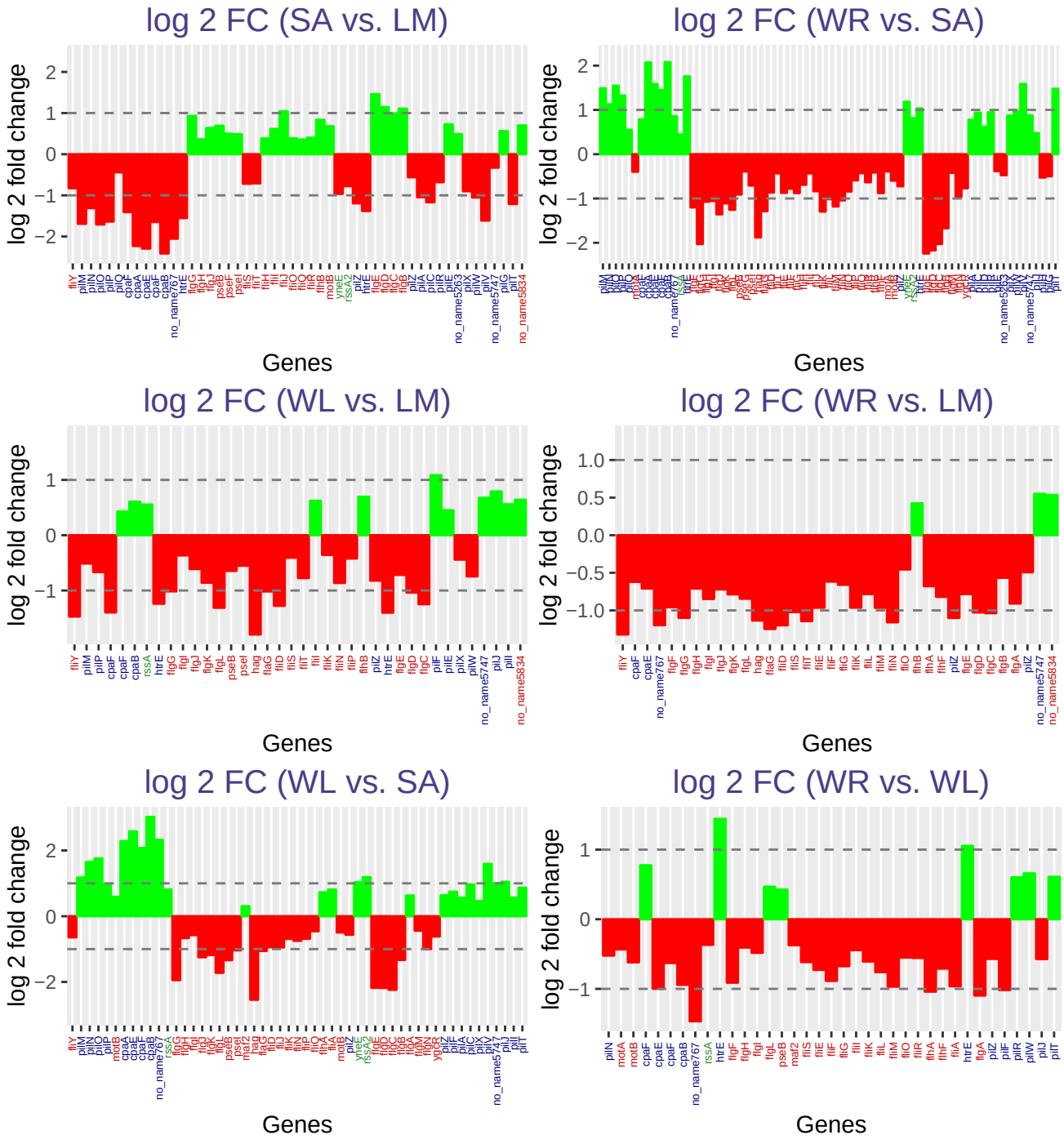


Figure S12. Barplot for all pairs of comparisons. Only motility-associated genes are shown. Dashed line indicating |logFC| = 1 (twofold change of expression). X-axis: genes ordered according to their chromosomal position.

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 (Krishnan et al., 2011), chicken (Rao et al., 2013) or human (Vinogradov, 2005). 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" (Risso et al., 2011).

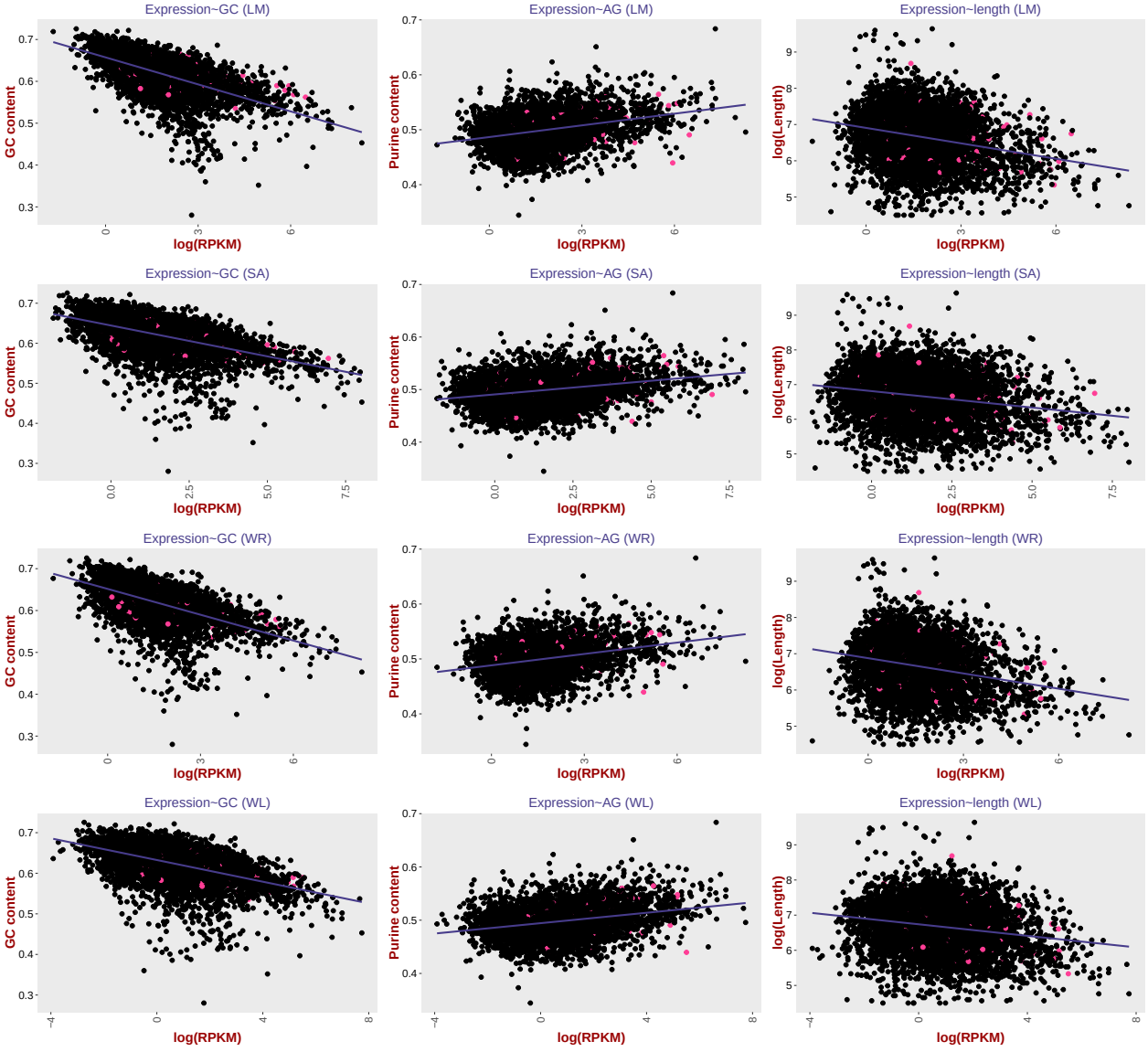


Figure S13. For each condition, plot showing log of RPKM values for all genes against i) GC content of the gene (left column), ii) purine content of the gene (mid column), iii) length of the gene (right column). Motility-associated genes are shown with pink dots.

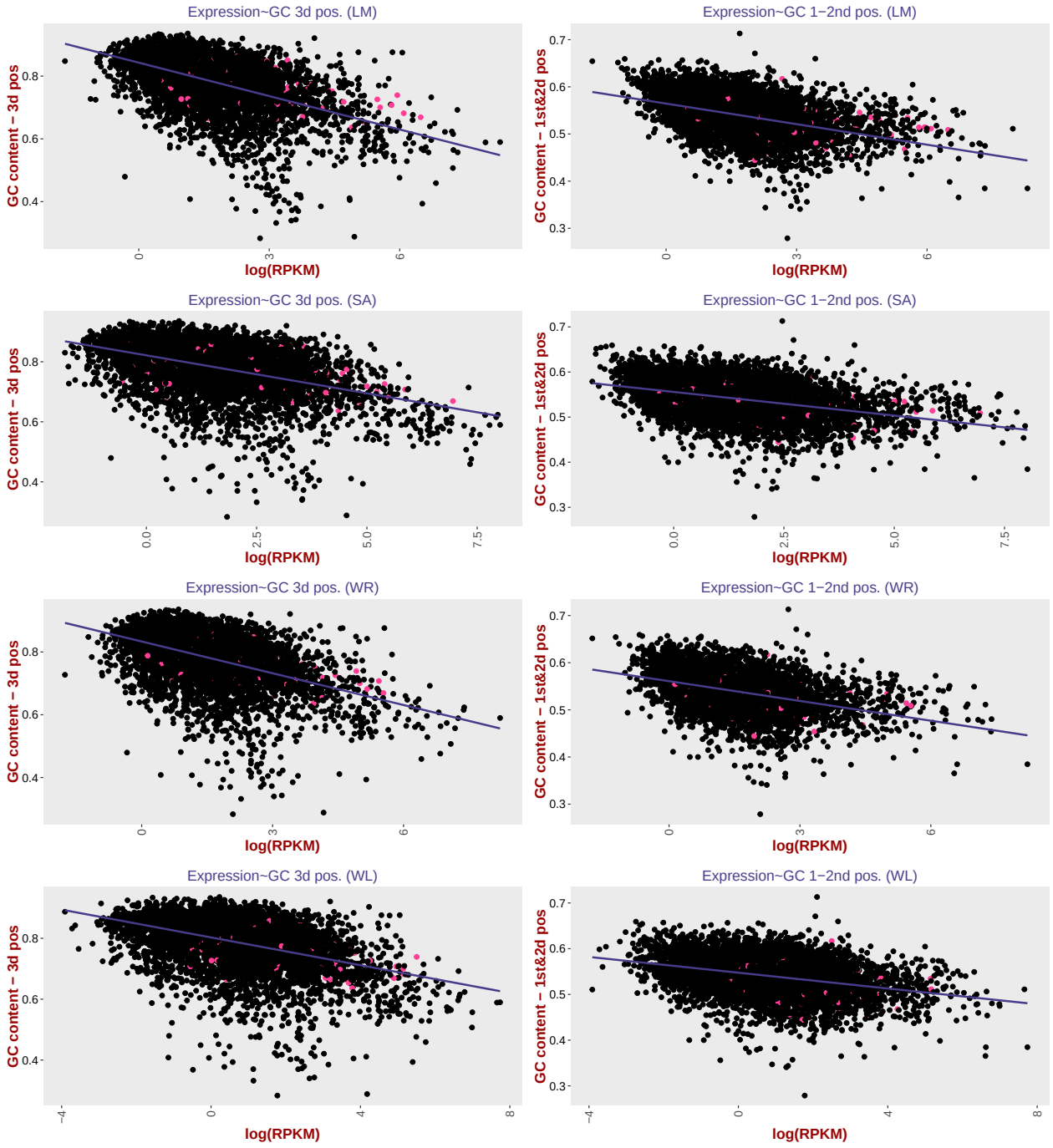


Figure S14. For each condition, plot showing log of RPKM values for all genes against i) GC content at the third codon position (left column), ii) GC content at the first two positions (right column). Motility-associated genes are shown with pink dots.

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	-0.64	< 2.2e-16
LM ~ GC-content (1st&2d pos.)	-0.44	1.1e-288
LM ~ GC-content (3d pos.)	-0.49	< 2.2e-16
SA ~ GC-content	-0.56	< 2.2e-16
SA ~ GC-content (1st&2d pos.)	-0.39	1.4e-220
SA ~ GC-content (3d pos.)	-0.43	4.2e-274
WL ~ GC-content	-0.51	< 2.2e-16
WL ~ GC-content (1st&2d pos.)	-0.32	1e-148
WL ~ GC-content (3d pos.)	-0.42	6.3e-262
WR ~ GC-content	-0.59	< 2.2e-16
WR ~ GC-content (1st&2d pos.)	-0.42	1.9e-252
WR ~ GC-content (3d pos.)	-0.45	4e-294

Table S15. 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).

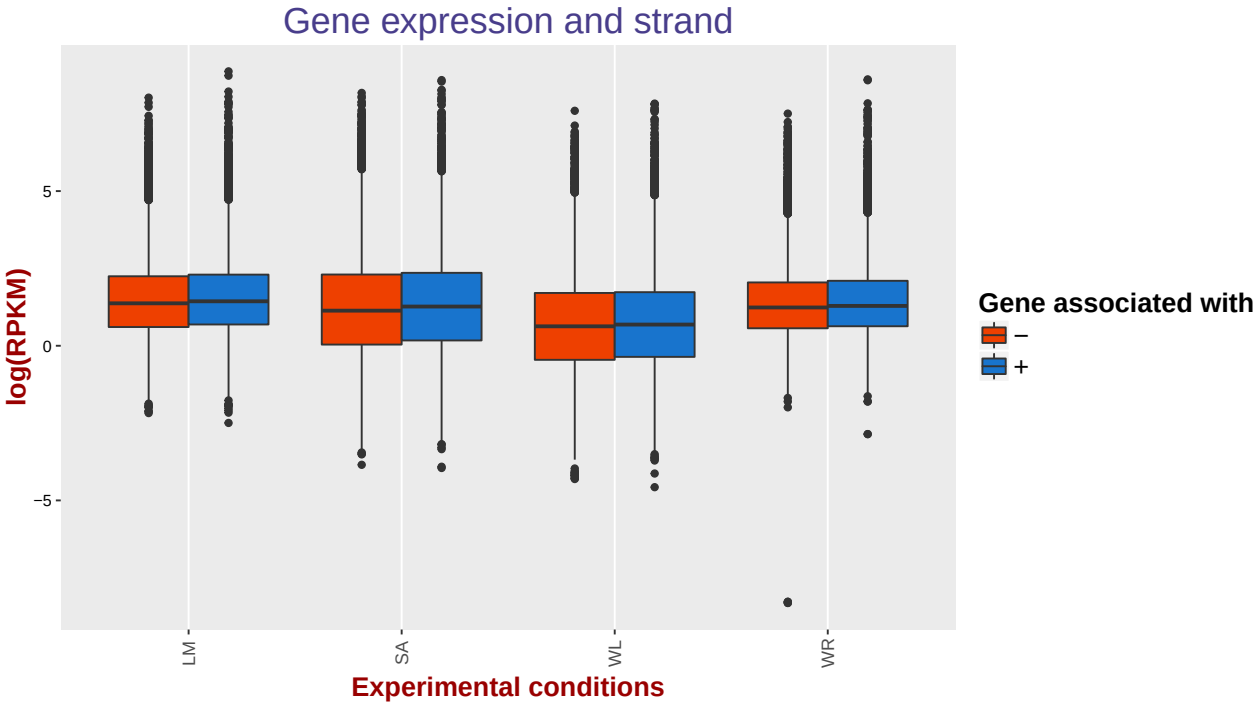


Figure S16. For each condition, plot showing log of RPKM values conditioned by the strand on which the gene is located (this information was not available for all, but for most of the genes).

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 (Borcard et al., 2011). 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).

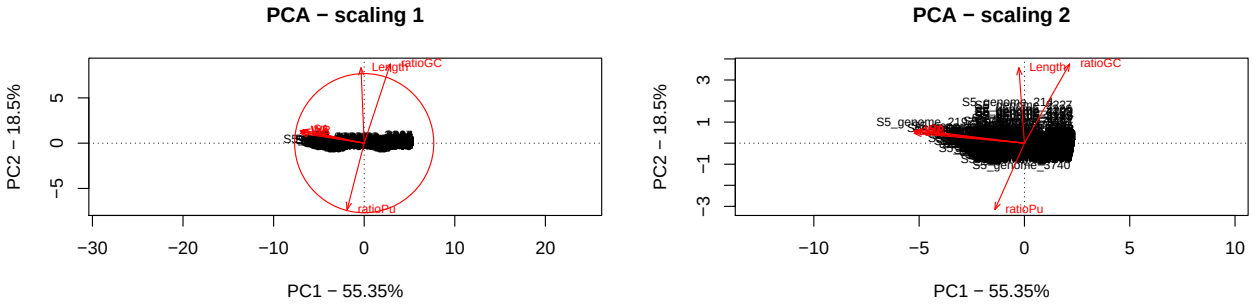


Figure S17. PCA plots for all genes and "structural parameters". Left: scaling 1 (angles are meaningless), right: scaling 2 (distances are meaningless).

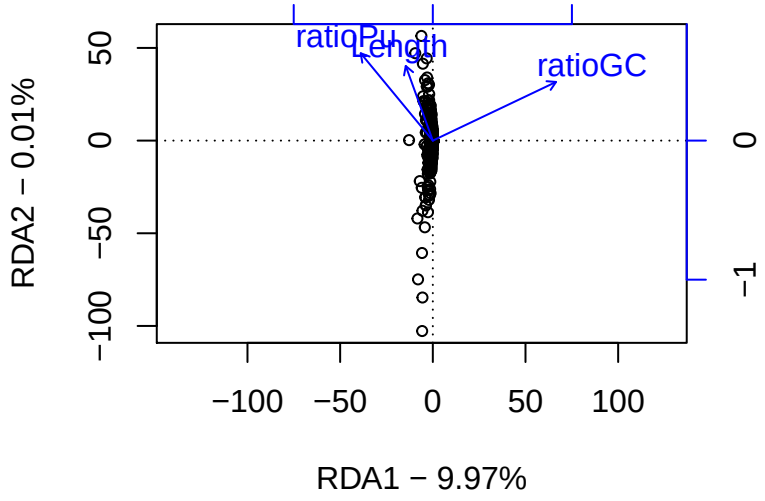


Figure S18. RDA plot of expression values regressed against "structural parameters".

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).

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.

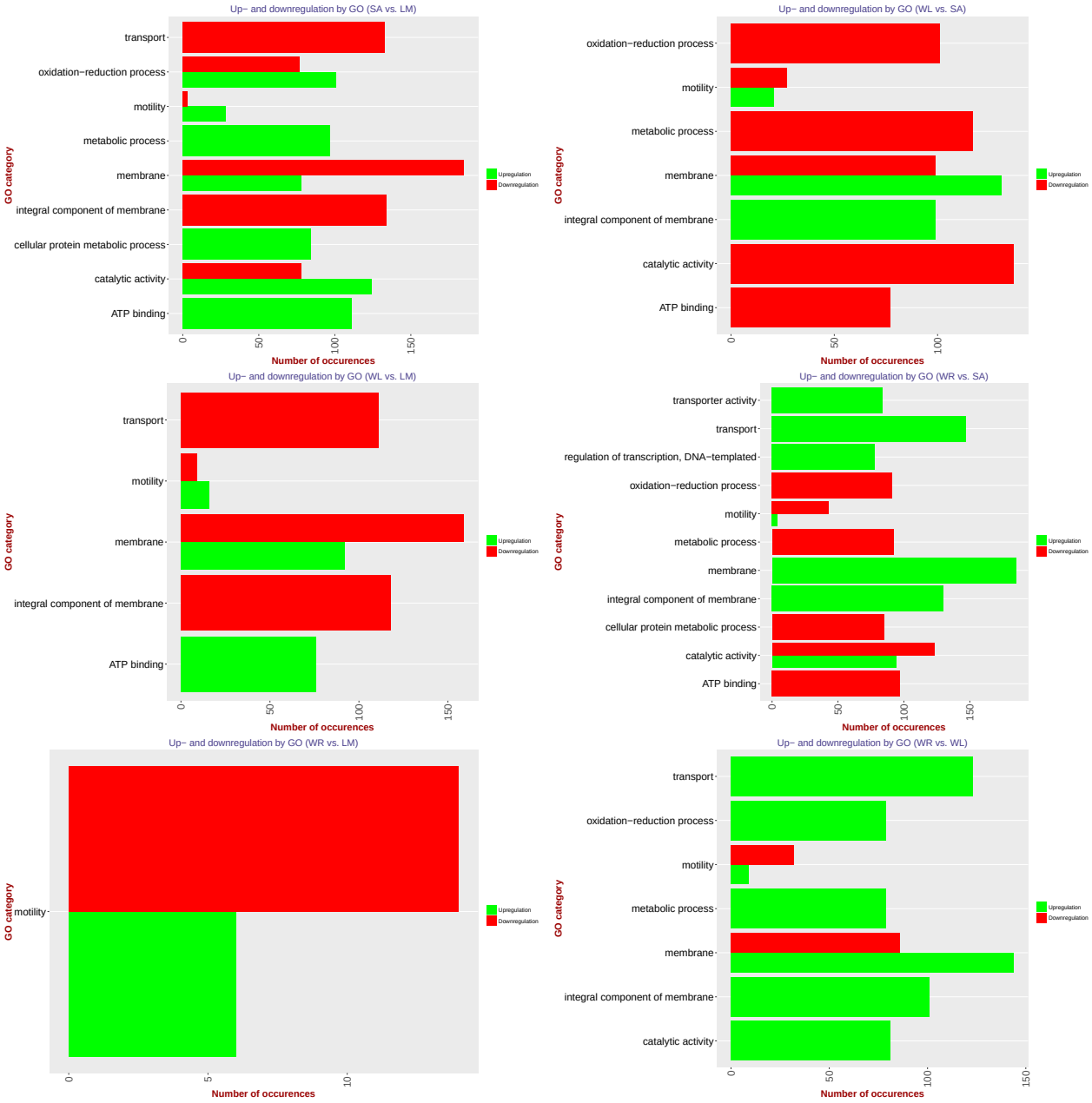


Figure S19. Barplots showing for each pairs of condition to which GO category the up- and downregulated genes belong. Threshold: 75 occurences of the GO category (motility added independently of the number of occurences, as explained in the text).

KEGG pathways

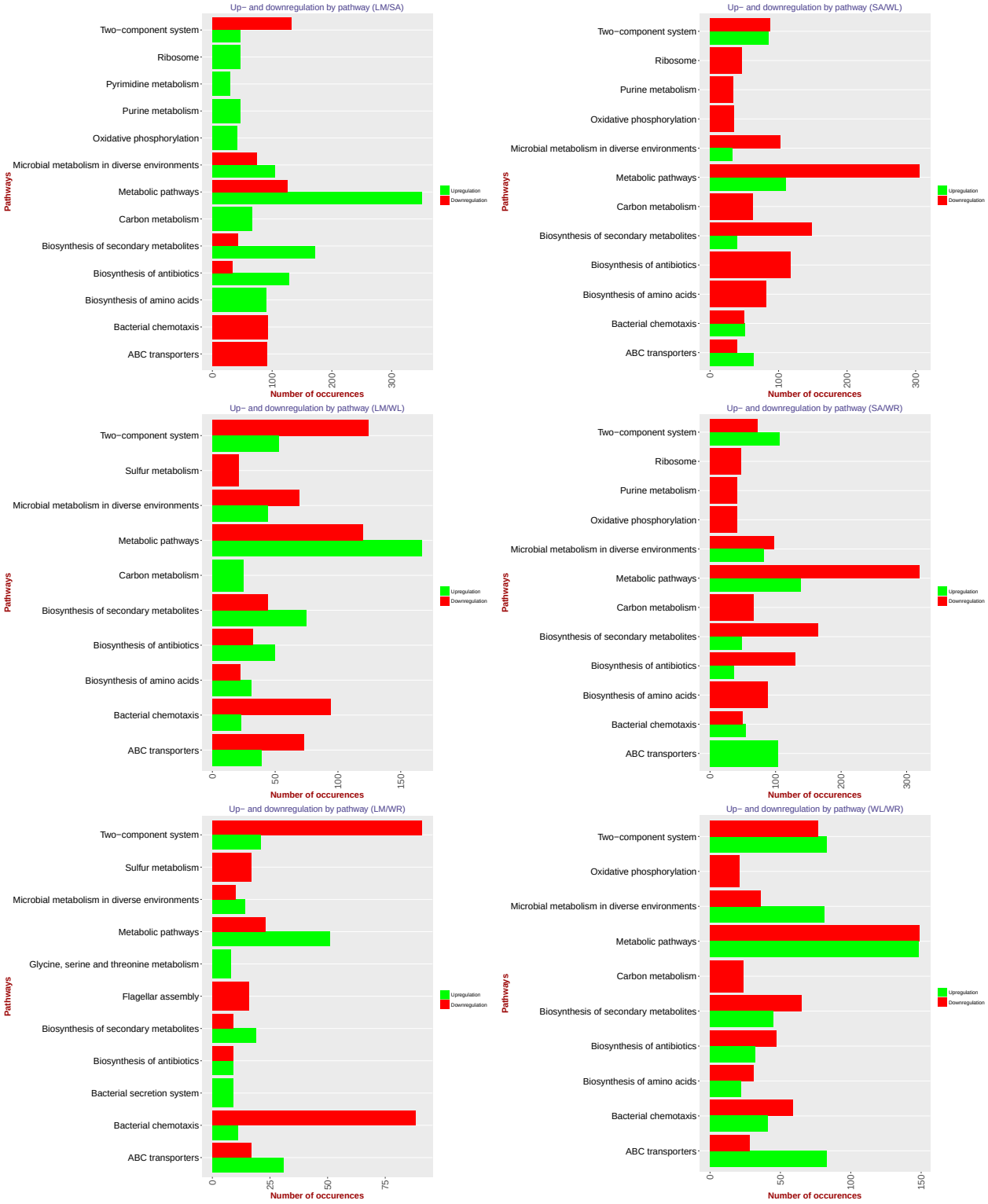


Figure S20. For all pairs of comparisons, barplots showing to which KEGG pathway the down- and upregulated genes belong.

Further statistical tests

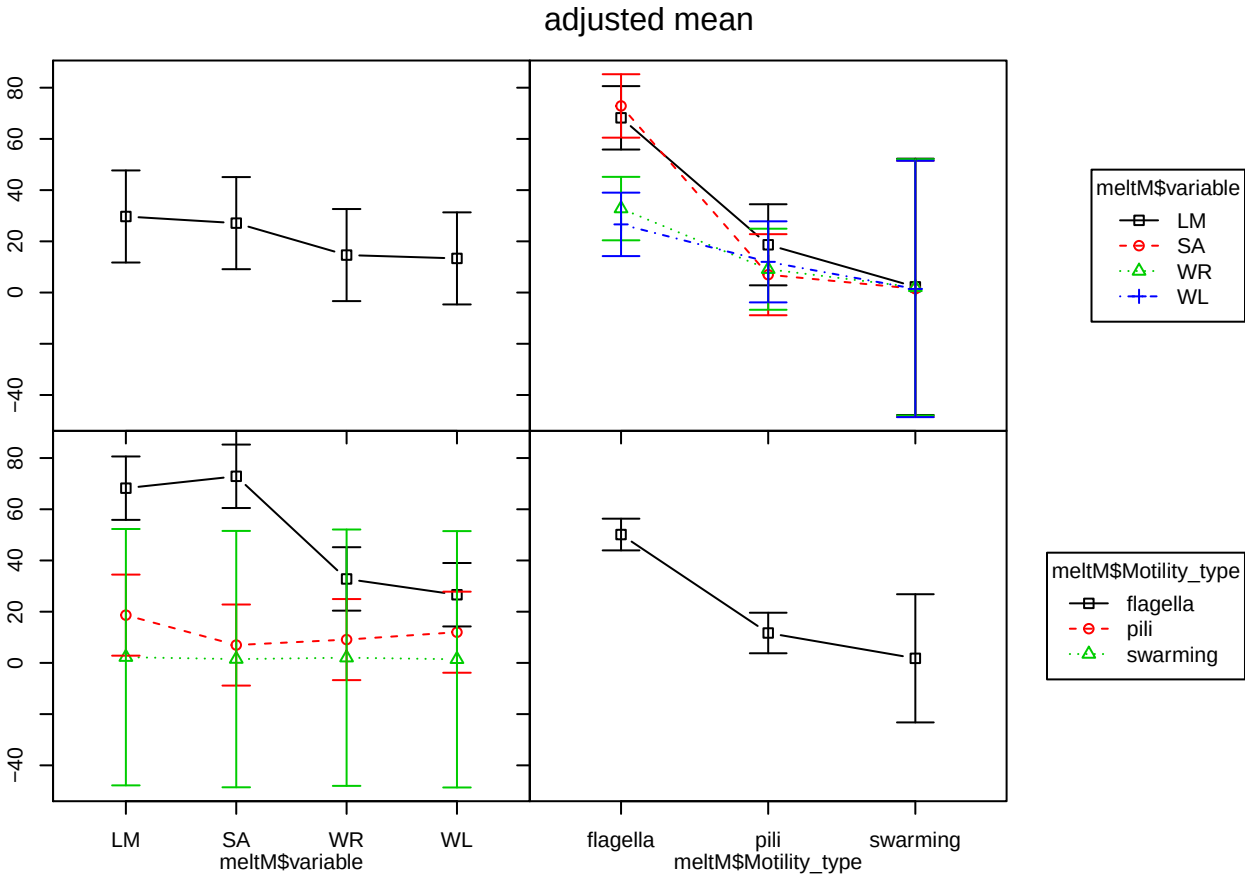


Figure S21. Interaction plots.

	Value	Df	Sum of Sq	F	Pr(>F)
LM-SA : flagella-pili	-16.32	1	2477.76	0.33	0.98
LM-WR : flagella-pili	25.89	1	6237.75	0.83	0.98
LM-WL : flagella-pili	34.94	1	11360.83	1.51	0.98
SA-WR : flagella-pili	42.21	1	16578.24	2.21	0.98
SA-WL : flagella-pili	51.26	1	24449.79	3.25	0.98
WR-WL : flagella-pili	9.05	1	762.19	0.10	0.98
LM-SA : flagella-swarming	-5.39	1	41.06	0.01	0.98
LM-WR : flagella-swarming	35.24	1	1755.41	0.23	0.98
LM-WL : flagella-swarming	40.78	1	2351.08	0.31	0.98
SA-WR : flagella-swarming	40.63	1	2333.41	0.31	0.98
SA-WL : flagella-swarming	46.17	1	3013.53	0.40	0.98
WR-WL : flagella-swarming	5.54	1	43.43	0.01	0.98
LM-SA : pili-swarming	10.93	1	162.89	0.02	0.98
LM-WR : pili-swarming	9.35	1	119.16	0.02	0.98
LM-WL : pili-swarming	5.84	1	46.51	0.01	0.98
SA-WR : pili-swarming	-1.58	1	3.41	0.00	0.98
SA-WL : pili-swarming	-5.09	1	35.32	0.00	0.98
WR-WL : pili-swarming	-3.51	1	16.78	0.00	0.98

Table S22. 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.

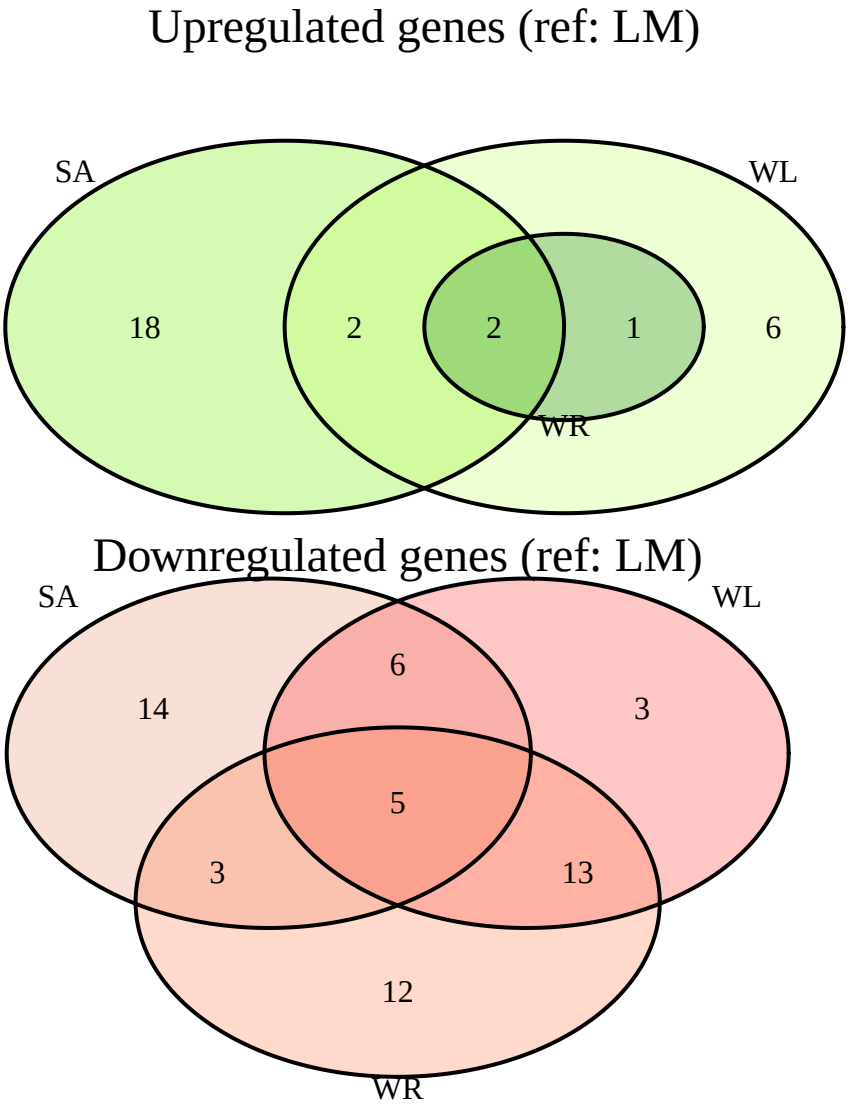


Figure S23. Example of Venn diagram for up- and downregulated genes. The number indicates the number of motility-associated genes up- (top) and downregulated (bottom) in the indicated condition when compared to LM.

Genome plots

Finally, we tried to visualize the clusters of motility-associated genes along the *Pseudomonas S5* genome with tools of the genoPlotR package (Lionel et al., Kultima, and Andersson, 2010). 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.

All anotated motility-associated genes

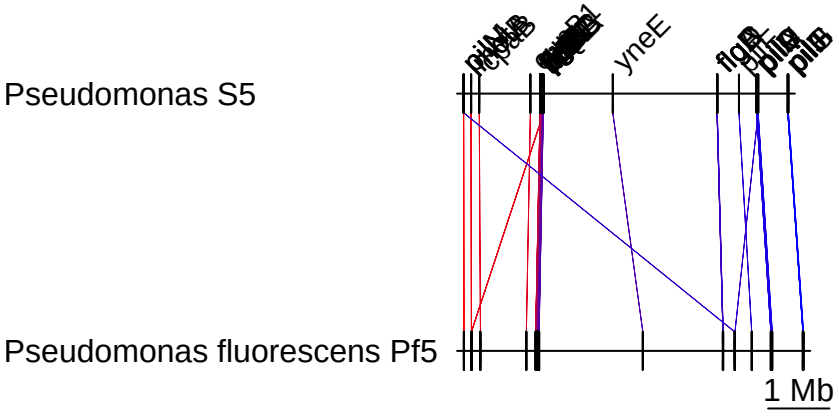


Figure S24. Genome plot for all motility-associated genes: comparison Pseudomonas protegens*S5 and* Pseudomonas fluorescens *Pf5.*

Flagellum-associated genes

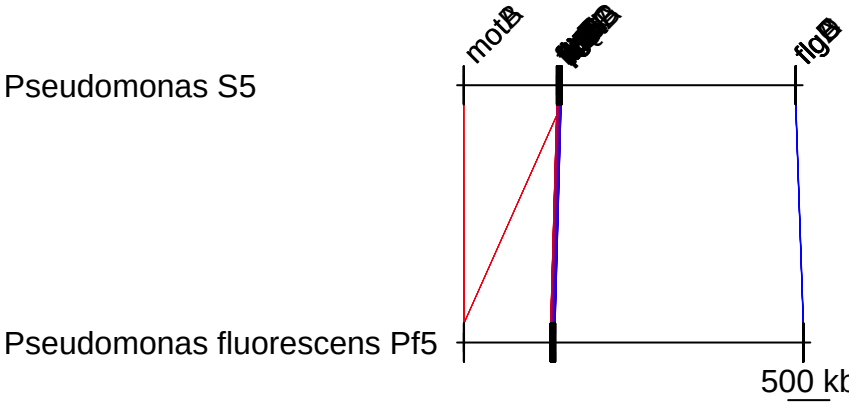


Figure S25. Genome plot for flagellum-associated genes: comparison Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

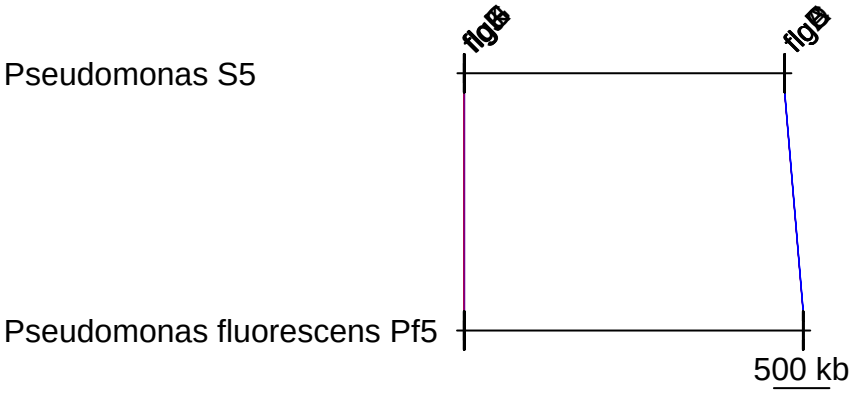


Figure S26. Genome plot for flg *family genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

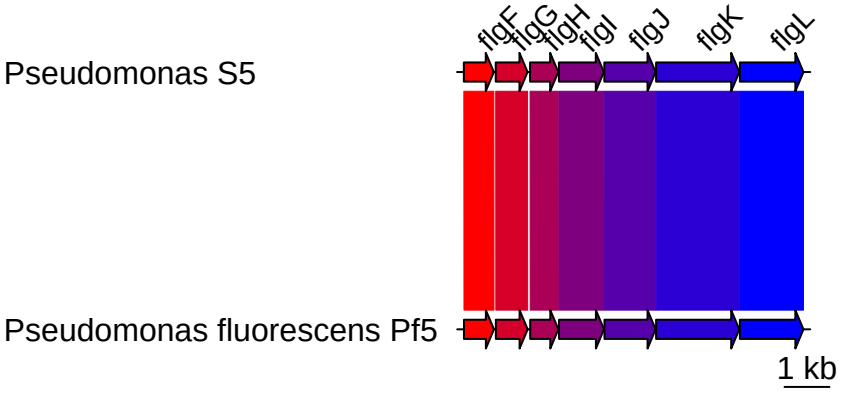


Figure S27. Genome plot for flg *family genes (close-up 1): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

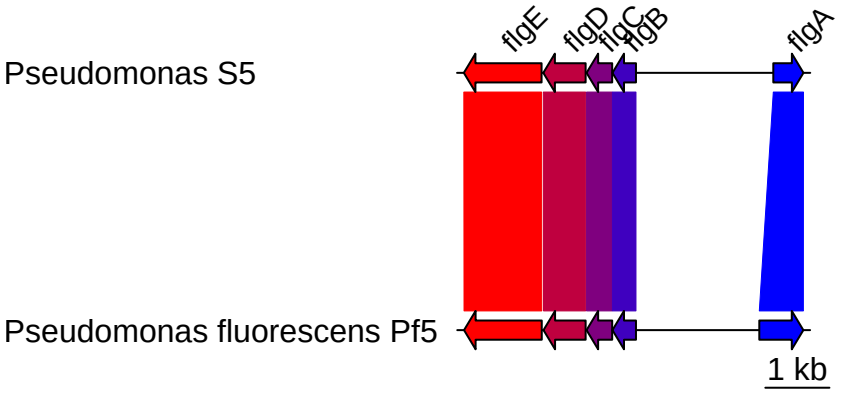


Figure S28. Genome plot for flg *family genes (close-up 2): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

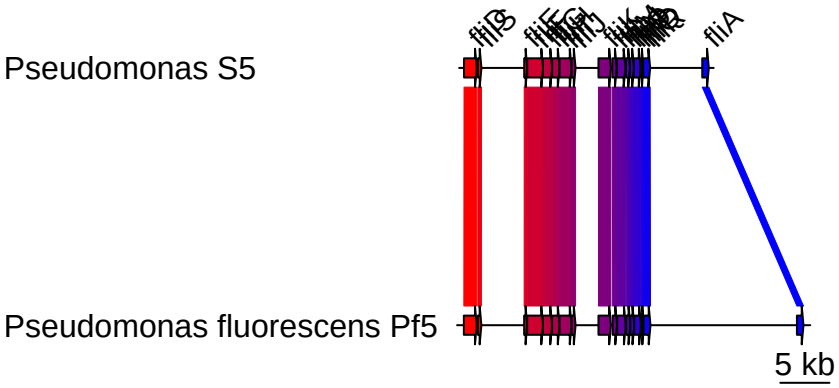


Figure S29. Genome plot for fli *family genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

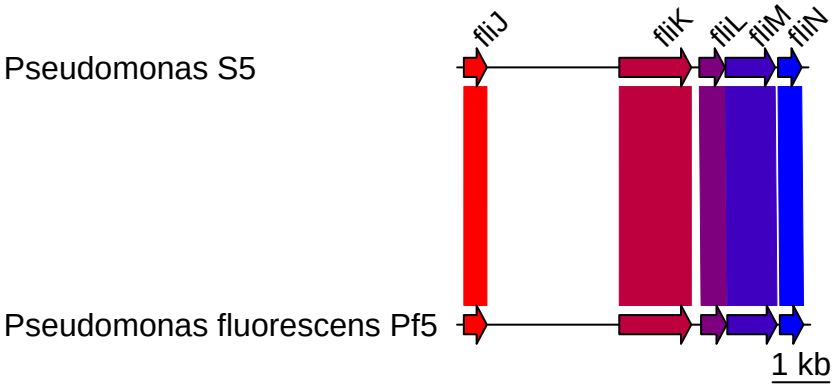


Figure S30. Genome plot for fli *family genes (close-up): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

Pilus-associated genes

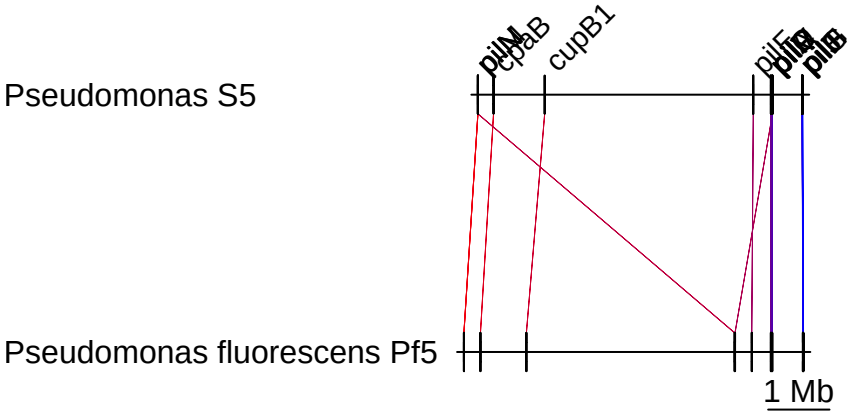


Figure S31. Genome plot for pilus-associated genes: comparison Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

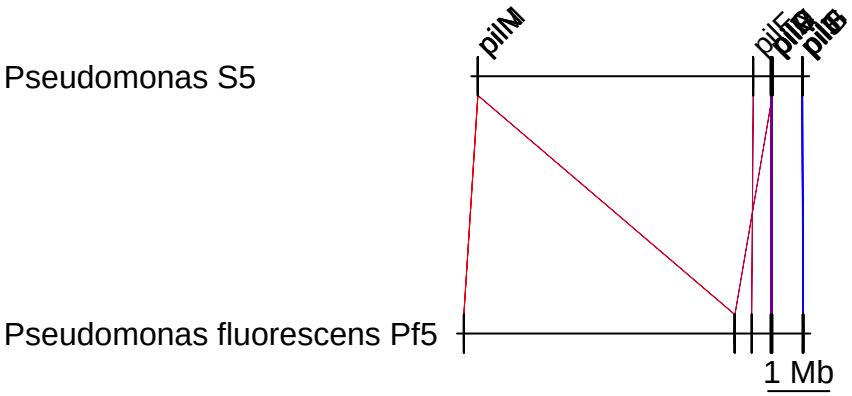


Figure S32. Genome plot for pil *family genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

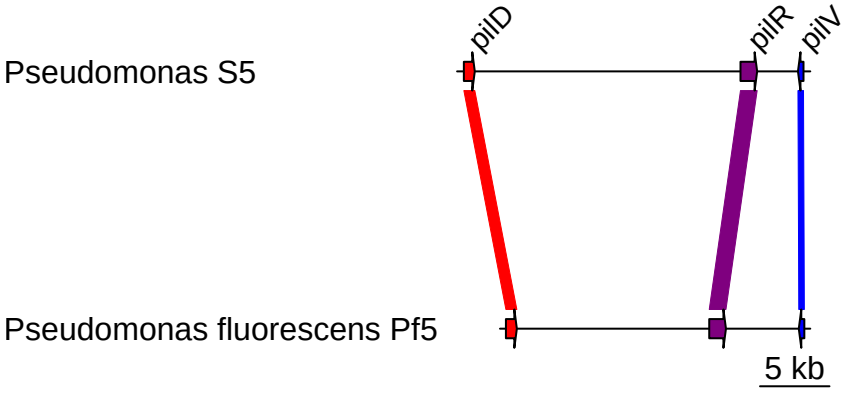


Figure S33. Genome plot for pil *family genes (close-up 1): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

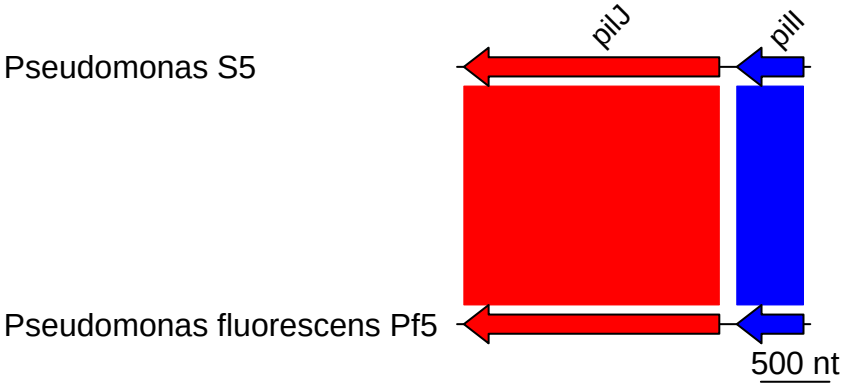


Figure S34. Genome plot for pil *family genes (close-up 2): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

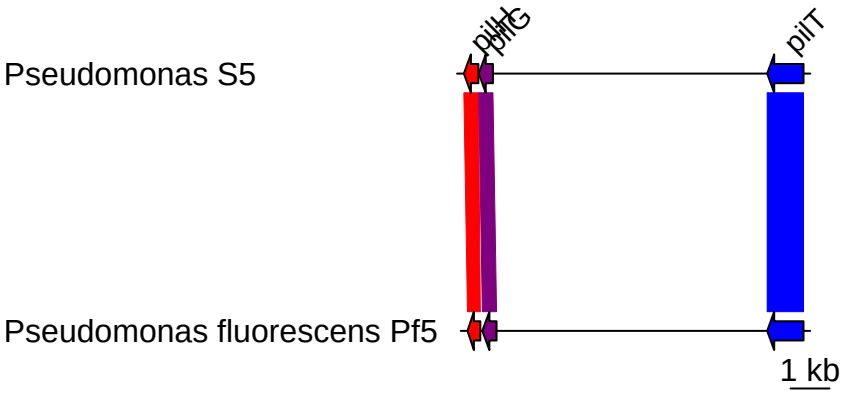


Figure S35. Genome plot for pil *family genes (close-up 3): comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

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)

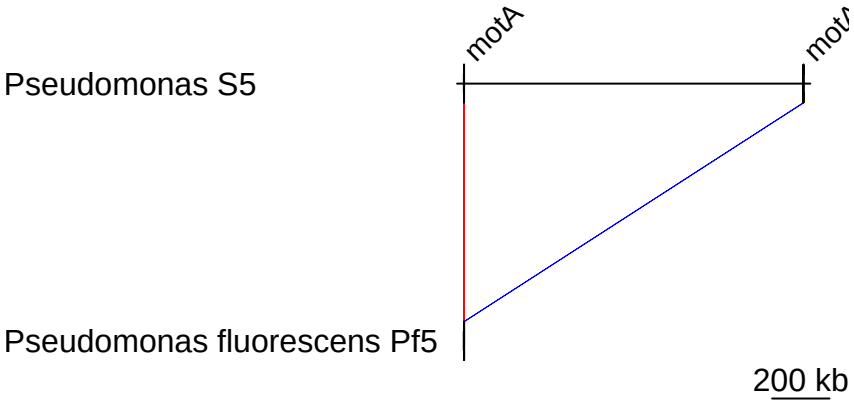


Figure S36. Genome plot for motA *genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

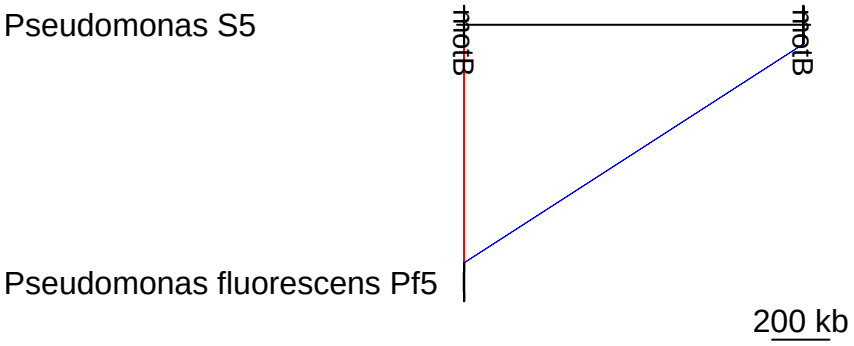


Figure S37. Genome plot for motB *genes: comparison* Pseudomonas protegens *S5 and* Pseudomonas fluorescens *Pf5.*

References

[1] S. M. Bache and H. Wickham. *magrittr: A Forward-Pipe Operator for R*. R package version 1.5. 2014.

[2] D. Borcard et al. *Numerical ecology with R*. Springer-Verlag New York, 2011.

[3] Y. Chen et al. "edgeR : differential expression analysis of digital gene expression data". In: *User ' s Guide* (2015), p. 104p.

[4] H. Chen. *VennDiagram: Generate High-Resolution Venn and Euler Plots*. R package version 1.6.17. 2016.

[5] H. De Rosario-Martinez. *phia: Post-Hoc Interaction Analysis*. R package version 0.2-1. 2015.

[6] L. Diray-Arce et al. "Transcriptome assembly, profiling and differential gene expression analysis of the halophyte Suaeda fruticosa provides insights into salt tolerance.". In: *BMC genomics* 16.1 (2015), p. 353.

[7] R. Kolde. *pheatmap: Pretty Heatmaps*. R package version 1.0.8. 2015.

[8] N. M. Krishnan et al. "De novo sequencing and assembly of Azadirachta indica fruit transcriptome". In: *Current Science* 101.12 (2011), pp. 1553-1561.

[9] G. Lionel et al., J. R. Kultima, et al. "genoPlotR: comparative gene and genome visualization in R". In: *Bioinformatics* 26.18 (2010), pp. 2334-2335.

[10] J. Oksanen et al. *vegan: Community Ecology Package*. R package version 2.3-5. 2016.

[11] Y. S. Rao et al. "Impact of GC content on gene expression pattern in chicken.". In: *Genetics, selection, evolution* 45.1 (2013), p. 9.

[12] D. Risso et al. "GC-Content Normalization for RNA-Seq Data". In: *BMC Bioinformatics* (2011), p. 17.

[13] M. D. Robinson et al. "edgeR: A Bioconductor package for differential expression analysis of digital gene expression data". In: *Bioinformatics* 26.1 (2009), pp. 139-140.

[14] A. E. Vinogradov. "Dualism of gene GC content and CpG pattern in regard to expression in the human genome: magnitude versus breadth". In: *Trends in Genetics* 21.12 (2005), pp. 633-639.

[15] H. Wickham. "Reshaping Data with the reshape Package". In: *Journal of Statistical Software* 21.12 (2007), pp. 1-20.

[16] H. Wickham and R. Francois. *dplyr: A Grammar of Data Manipulation*. R package version 0.4.3. 2015.

[17] H. Wickhamm. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.

[18] Y. Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, 2013.

Table S38. List of the annotated motility-associated genes.

A : Flagella-related genes

GENE ID	GENE NAME	FUNCTION
S5_genome_330	fliY	Cystine-binding periplasmic protein
S5_genome_628	motA	Reponse regulator receiver domain
S5_genome_629	motB	Flagellar motor protein
S5_genome_1770	FlgF	flagellar basal body rod protein FlgF
S5_genome_1771	flgG	Gene function: The basal body constitutes a major portion of the flagellar organelle and consists of four rings (L,P,S, and M) mounted on a central rod. The rod consists of about 26 subunits of FlgG in the distal portion, and FlgB, FlgC and FlgF are thought to build up the proximal portion of the rod with about 6 subunits each.
S5_genome_1772	flgH	Assembles around the rod to form the L-ring and probably protects the motor/basal body from shearing forces during rotation.
S5_genome_1773	flgI	Assembles around the rod to form the L-ring and probably protects the motor/basal body from shearing forces during rotation.
S5_genome_1774	flgJ	flagellar rod assembly protein
S5_genome_1775	flgK	Involved in flagellum assembly and cell motility
S5_genome_1776	flgL	flagellar rod assembly protein
S5_genome_1786	pseB	
S5_genome_1788	pseF	Catalyzes the final step in the biosynthesis of pseudaminic acid, a sialic-acid-like sugar that is used to modify flagellin. Mediates the activation of pseudaminic acid with CMP by forming CMP-pseudaminic acid (By similarity). .
S5_genome_1789	pseG	Nucleotide sugar hydrolase that catalyzes the fourth step in the biosynthesis of pseudaminic acid, a sialic-acid-like sugar that is used to modify flagellin. Mediates the removal of UDP from C-1 of UDP-2,4-diacetamido-2,4,6-trideoxy-beta-L-altropyranose forming

		2,4-diacetamido-2,4,6-trideoxy-beta-L-altropyranose.
S5_genome_1790	pseI	Pseudaminic acid synthase: Catalyzes the fifth step in the biosynthesis of pseudaminic acid, a sialic-acid-like sugar that is used to modify flagellin. Catalyzes the condensation of phosphoenolpyruvate with 2,4-diacetamido-2,4,6-trideoxy-beta-l-altropyranose, forming pseudaminic acid. .
S5_genome_1792	maf2	putative membrane protein
S5_genome_1795	flaG	flagellar protein FlaG
S5_genome_1796	fliD	flagellar hook protein FliD
S5_genome_1794	hag	flagellin
S5_genome_1797	fliS	flagellar biosynthesis protein FliS
S5_genome_1798	fliT	flagellar assembly protein FliT
S5_genome_1802	fliE	flagellar hook-basal body protein FliE
S5_genome_1803	fliF	flagellar M-ring protein
S5_genome_1804	fliG	flagellar motor switch protein
S5_genome_1805	fliH	flagellar assembly protein FliH
S5_genome_1806	fliI	Flagellum-specific ATP synthase
S5_genome_1807	fliJ	flagellar biosynthesis chaperone
S5_genome_1811	fliK	flagellar hook-length control protein
S5_genome_1812	fliL	flagellar basal body protein FliL
S5_genome_1813	fliM	Flagellar motor switch protein
S5_genome_1814	fliN	FliN is one of three proteins (FliG, FliN, FliM) that form the rotor-mounted switch complex (C ring), located at the base of the basal body. This complex interacts with the CheY and CheZ chemotaxis proteins, in addition to contacting components of the motor that determine the direction of flagellar rotation (By similarity).
S5_genome_1815	fliO	Involved in flagellar biosynthesis and adherence. May have a role in assisting the proper localization of the various flagellar components and in the localization and assembly of the adhesin
S5_genome_1816	fliP	Flagellar biosynthetic protein FliP
S5_genome_1817	fliQ	flagellar biosynthesis protein FliQ
S5_genome_1818	fliR	flagellar motor protein
S5_genome_1819	flhB	Flagellar biosynthetic protein

S5_genome_1820	flhA	Flagellar biosynthesis protein
S5_genome_1821	flhF	Necessary for flagella biosynthesis. May be involved in translocation of the flagellum
S5_genome_1823	fliA	This sigma factor controls the expression of flagella-related genes. Sigma factors are initiation factors that promote the attachment of RNA polymerase to specific initiation sites and are then released.
S5_genome_1828	motA	flagellar motor protein
S5_genome_1829	motB	flagellar motor protein
S5_genome_4569	flgE	flagellar hook protein FlgE
S5_genome_4570	flgD	flagellar basal body rod modification protein FlgD
S5_genome_4571	flgC	flagellar basal body rod protein FlgC
S5_genome_4572	flgB	flagellar basal body rod protein FlgB
S5_genome_4575	flgA	Flagella basal body P-ring formation protein FlgA
S5_genome_4576	FlgM	flagellar biosynthesis anti-sigma factor FlgM
S5_genome_4577	FlgN	flagellar biosynthesis protein FlgN
S5_genome_4578	ygcR	flagellar brake protein YcgR
S5_genome_5834	no_name	cysteine ABC transporter substrate-binding protein

B : Pilus-related genes

GENE ID	GENE NAME	FUNCTION
S5_genome_518	pilM	
S5_genome_519	pilN	pilus assembly protein PilN
S5_genome_520	pilO	type IV pilus biogenesis protein PilO
S5_genome_521	pilP	pilus assembly protein PilP
S5_genome_521	pilQ	type IV pilus biogenesis protein PilQ
S5_genome_756	cpaF	pilus assembly protein
S5_genome_757	cpaA	pilus assembly protein, protease CpaA
S5_genome_764	cpaE	pilus assembly protein CpaE
S5_genome_765	cpaF	
S5_genome_767	no_name	pilin; Flp/Fap pilin component family protein
S5_genome_766	cpaB	pilus assembly protein CpaB
S5_genome_1619	cupB1	type I plus biogensis protein cupB1
S5_genome_1621	htrE ?	Part of the yadCKLM-htrE-yadVN fimbrial operon. Could contribute to adhesion to various surfaces in specific environmental niches. Probably involved in the export and assembly of fimbrial subunits across the outer membrane.
S5_genome_4011	no_name	type I pilus usher pathway chaperone CsuC
S5_genome_4437	htrE	Part of the yadCKLM-htrE-yadVN fimbrial operon. Could contribute to adhesion to various surfaces in specific environmental niches. Probably involved in the export and assembly of fimbrial subunits across the outer membrane.

S5_genome_4938	pilF	pilus assembly protein PilW
S5_genome_5238	pilA	fimbrial protein
S5_genome_5239	pilC	Type IV pilus assembly protein pilC
S5_genome_5240	pilD	Type 4 prepilin-like proteins leader peptide-processing enzyme
S5_genome_5261	pilR	type 4 fimbriae expression regulatory protein PilR
S5_genome_5262	pilE	type IV pilus biogenesis protein
S5_genome_5263	-	type IV pilus-associated protein
S5_genome_5264	pilX	pilus assembly protein PilX
S5_genome_5265	pilW	pilus assembly protein PilW
S5_genome_5266	pilV	pilus assembly protein PilV
S5_genome_5747	no_name	sensor histidine kinase (part of the two component system)
S5_genome_5748	pilJ	protein pilJ
S5_genome_5749	pilI	type IV pilus biogenesis protein PilI
S5_genome_5750	pilH	gene function: type IV pilus response regulator/twitching mobility protein
S5_genome_5751	pilG	pilus assembly protein PilG
S5_genome_5761	pilT	Twitching motility protein

C : Swarming-related specific genes

GENE ID	GENE NAME	FUNCTION
S5_genome_1579	rssA	
S5_genome_2958	yneE	swarming motility YneE
S5_genome_3916	rssA2	swarming motility regulation sensor protein RssA

```
---
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))
```

```
##############################################################################
##### RNA-SEQ DATA ANALYSIS - PSEUDOMONAS S5 - R FUNCTIONS
##############################################################################
##### Spring 2016 - MLS - UNIL - Marie Zufferey
##### !!! some hard-coded parameters, file shape and formats not checked
library(edgeR)
library(readr)
library(ggplot2)
library(pheatmap)
library(reshape2)
library(rtracklayer)
library(magrittr)
library(dplyr)
library(grid)
library(ggthemes)
library(genoPlotR)
library(gtable)
library(tm)           # wc
library(SnowballC)    # wc
library(wordcloud)    # wc
library(RColorBrewer) # wc
setwd("PATH_TO_FOLDER")

##############################################################################
###### Read abundance files
##############################################################################

getRawCounts <- function(abd_fld){
  # abd_fld <- path_to_abundances <- "../data/abundances/"
  files <- dir(abd_fld, pattern=".tsv$")
  # paste the path to your files - make sure that you have the path in front the files
  files <- paste0(abd_fld, files)
  samples <- paste0(rep(c('LM','SA','WL','WR'), each = 4), rep(1:4,4))

  # read kallisto files again to get counts of transcripts
  transcripts <- readDGE(files,
                         columns = c(1,4),
                         group = rep(c('LM','SA','WL','WR'), each = 4),
                         labels = samples)

  # get counts of transcripts
  tr_counts <- transcripts$counts
  return(tr_counts)
}

##############################################################################
###### Create DGEList object
##############################################################################

getRawData <- function(abd_fld, threshold=T){

    # get counts of transcripts
    tr_counts <- getRawCounts(abd_fld)
    tr_cpms <- cpm(tr_counts)
    keep <- rowSums(tr_cpms > 1) >= 4
    tr_counts_clean <- tr_counts[keep,]
#    sum(!keep)
    dt <- DGEList(counts = tr_counts_clean,
                  group = rep(c('LM','SA','WL','WR'), each = 4))

    return(dt)
}


##############################################################################
###### "Normalization" (library size and dispersion)
##############################################################################

getDGE <- function(abd_fld, threshold=T){

  dt <- getRawData(abd_fld, threshold=T)

  dt <- calcNormFactors(dt)
  dt <- estimateCommonDisp(dt)
```

```
  dt <- estimateTagwiseDisp(dt)

  return(dt)
}

##############################################################################
###### Get mean for each conditions (mean of replicates)
##############################################################################

getMeanData <- function(data){
  data %<>% as.data.frame
  LMcol <- colnames(data)[regexpr("LM", colnames(data))>0]
  data$LM <- rowMeans(subset(data, select = LMcol), na.rm = TRUE)
  SAcol <- colnames(data)[regexpr("SA", colnames(data))>0]
  data$SA <- rowMeans(subset(data, select = SAcol), na.rm = TRUE)
  WRcol <- colnames(data)[regexpr("WR", colnames(data))>0]
  data$WR <- rowMeans(subset(data, select = WRcol), na.rm = TRUE)
  WLcol <- colnames(data)[regexpr("WL", colnames(data))>0]
  data$WL <- rowMeans(subset(data, select = WLcol), na.rm = TRUE)
  dataMean <- data[,c("LM", "SA", "WR", "WL")]
  return(dataMean)
}

##############################################################################
###### RPKM normalization (retrieve from Kamil and Andrea's tutorial)
##############################################################################

## rpkm calculations
myrpkm <- function(counts, lengths) {
  rate <- counts / lengths
  return(rate / sum(counts) * 1e9)
}

##############################################################################
###### Pairwise exact test of differential expression
##############################################################################

# returns a dataframe with i.a. logFC, logCPM, pval, FDR
# for a pairwise exact test for a given pair of conditions
# genes are identified with "Transcript" column (e.g. "S5_genome_4011")
# a column "Pair" is added with information about conditions tested (e.g. "LM/SA")
pairTestGenes <- function(dt, sel_genes, cond1, cond2){
  # dt is the DGEList object
  # data_annot <- annot <- read.csv("../data/annot_mot.csv", sep=",")
  # cond1 <- "LM";cond2 <- "SA"
  # cond1 and cond2 the conditions to test
  # sel_genes <- data_annot$Gene_position
  # the genes we want to select (position = transcript name)
  # this imports the .tsv files in your R environment
  # select only the motility
  de.tr <- exactTest(dt, pair = c(cond1, cond2))
  tT.transcripts <- topTags(de.tr, n = nrow(dt), p.value = 0.05)
  det.list <- tT.transcripts$table
  det.list %<>% as.data.frame
  det.list$Transcript <- row.names(det.list)
  det.list$Pair <- rep(paste0(cond1,"/", cond2), nrow(det.list))
  det.list <- det.list[which(rownames(det.list) %in% sel_genes),]
  rownames(det.list) <- NULL
  return(det.list)
}


##############################################################################
###### DIFFERENT FUNCTIONS FOR COLOR SETTING
##############################################################################

# Function to set colors according to motility type
# (use for ggplot axis labels)
setMyCol_mot <- function(mot){
  if(is.na(mot)){           # must be the first condition tested !
    return("black")
  }
  else if(mot == "pili"){
    return("darkblue")
```

```
  }else if(mot =="flagella"){
    return("red3")
  }else if(mot =="swarming"){
    return("forestgreen")
  }else{
    return("black")
  }
}

setMyCol_smear <- function(mot){
  if(is.na(mot)){           # must be the first condition tested !
    return("black")
  }
  else if(mot == "pili"){
    return("darkblue")
  }else if(mot =="flagella"){
    return("gold")
  }else if(mot =="swarming"){
    return("forestgreen")
  }else{
    return("black")
  }
}

setMyFill <- function(mot){
  if(is.na(mot)){           # must be the first condition tested !
    return(NA)
  }
  else if(mot == "pili"){
    return("darkblue")
  }else if(mot =="flagella"){
    return("red3")
  }else if(mot =="swarming"){
    return("forestgreen")
  }else{
    return(NA)
  }
}


# Function to set colors according to significance
# (use for ggplot dots)
setMyCol_fdr <- function(fdr){
  if(fdr < 0.05){
    return("green")
  }else{
    return("gray48")
  }
}

# Function to set colors according to sign of FC
# (used for barplot)
setMyCol_fc <- function(x){
  if(is.na(x)){
    return("black")
  }else if(x<0){
    return("red")
  }else if(x>0){
    return("green")
  }else{
    return("black")
  }
}

# Function to set colors according to sign of the strand
# (used for line of logCPM, plot with 2 y axis)
setMyCol_strand <- function(x){
  if(is.na(x)){
    return("black")
  }else if(x=="+"){
    return("maroon3")
  }else if(x=="-"){
    return("mediumaquamarine")
  }else{
```

```
    return("black")
  }
}

setMyCol_PCA <- function(x){
  if(is.na(x)){
    return("black")
  }else if(x=="flagella"){
    return("tomato")
  }else if(x=="pili"){
    return("lightblue")
  }else if(x=="swarming"){
    return("yellowgreen")
  }else{
    return("black")
  }
}
setMyPch_PCA <- function(x){
  if(is.na(x)){
    return(4)
  }else if(x=="flagella"){
    return(1)
  }else if(x=="pili"){
    return(2)
  }else if(x=="swarming"){
    return(3)
  }else{
    return(4)
  }
}


#############################################################################
###### HEATMAPS FOR DIFFERENTIAL EXPRESSION
#############################################################################

heatmapPairs <- function(data, gbkData, annotData, tit){
  # retrieve the chromosomal position
  # data <- allPairs
  # annotData <- annot
  # gbkData <- gbkData
  # tit <- "logFC motility associated genes all pairs of conditions"
  #nrow(data)  #298
  data %<>% left_join(., gbkData, by =c("Transcript"="Locus_tag"))
  #nrow(data)  #298
  data %<>% left_join(., annotData, by =c("Transcript"="Gene_position"))

  data$Start %<>% as.character %>% as.numeric  # because stored as factor
  data <- data[order(data$Start),]  # order by starting position

  # need "unique" for the labels (and their colors) of the x axis
  # because x-axis in ggplot is the start position
  # duplicated because of different conditions
  uniqData <- unique(data[,c("Start", "Motility_type", "Gene_name")])
  # nrow(uniqData) # 82
  # -> ok, tested with labels=1:82 in ggplot axis.text.x

  # set colors of x-axis labels according to motility type
  labCol <- sapply(uniqData$Motility_type, function(x){setMyCol_mot(x)})

  # replace LM/SA -> SA vs. LM
  data$Pair <- gsub('(^.{2})/(.{2}$)', '\\2 vs. \\1', data$Pair)

  p <- ggplot(data, aes(x=Pair,y=as.factor(Start)))+
    geom_tile(aes(fill = logFC))+
    scale_fill_gradient2(low="red", high="green", mid="black", name="logFC")+
    scale_y_discrete("Genes",labels=uniqData$Gene_name)+
    scale_x_discrete("Conditions", expand=c(0,0))+
    ggtitle(tit)+
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.background = element_rect(fill="gray"),
          legend.title = element_text(face="bold"),
          axis.text.x = element_text(angle=45, hjust=1, size=10, colour="black"),
```

```
          axis.text.y = element_text(size=10,colour=labCol),
          axis.title.y = element_text(face="bold", colour="#990000", size=15),
          axis.title.x = element_text(face="bold", colour="#990000", size=15),
          plot.title =  element_text(colour="darkslateblue", size=20))
  return(p)
}


##########################################################################
###### SCATTERPLOT MATRIX OF DIFFERENTIAL EXPRESSION COMPARISONS
##########################################################################

# written by François Gillet ("Numerical ecology with R") !!!
# -> MZ: color changed
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col="steelblue", ...)
}

# https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/pairs.html
# -> MZ: changed behaviour with NA and spearman as method,
# -> the sign and the size
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- (cor(x, y,use= "pairwise.complete.obs", method="spearman"))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * (r+0.1))
}


##########################################################################
###### VOLCANO PLOT FOR ALL GENES (motility genes coloured; if wished, top X genes labelled)
##########################################################################
# with label for the top X genes if plotAnnot=T, label the myT genes most differentially expressed
volcanoAllPoints<- function(dt, annot, cond1, cond2, plotAnnot=F, myT=5){
  #abd_fld <- "../data/abundances/"
  #dt <- getDGE(abd_fld)
  #cond1 = "LM"
  #cond2 = "SA"

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

  tT.transcripts <- topTags(de.tr, n = nrow(dt), p.value = 0.05)
  det.list <- tT.transcripts$table

  allDF <- det.list
  allDF$Locus <- rownames(allDF)
  rownames(allDF) <- NULL
  allDF$log10p <- -log10(allDF$FDR)

  allExp <- full_join(allDF, annot, by=c("Locus"="Gene_position"))
  colPoints <- sapply(allExp$Motility_type, function(x){setMyCol_mot(x)})

  colFill <- sapply(allExp$Motility_type, function(x){setMyFill(x)})

  ytit <-expression(bold(paste("-",log[10]~" p-value (FDR)")))
  co <- c("darkblue",  "red3", "forestgreen")
  limx <- max(abs(allExp$logFC[which(!is.na(allExp$logFC))]))
  p <-  ggplot(allExp, aes(x=logFC, y=log10p,group=1))+
    geom_point(size=2, colour=colPoints, fill=colFill, shape=21)+
    #scale_y_continuous(bquote("-log"~ [10]~ "(Pval)"))+
    scale_y_continuous(ytit)+
    scale_x_continuous("log 2 fold change", limits=c(-limx,limx))+
    scale_fill_manual(name="ello", values=co)+
    ggtitle(paste0("Volcano plot ", cond2, " vs. ", cond1))+
```

```
    theme(axis.title.y = element_text( 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())
  ### Add locus label for the 5 gene with most changing expression
  # have tried first with the annotation
  # but the most changing expression was not annotated (so after that see blast perl script !!!)
  #fction <- gbkData[,c("Product", "Gene_id", "Locus_tag")]
  #fctionData <- full_join(fction, allExp, by=c("Locus_tag"="Locus"))

  if(plotAnnot){

    topX <- allExp[order(abs(allExp$logFC), decreasing=T),][1:myT,]

    for(i in 1:myT){
      p <- p + ggplot2::annotate("text", x = topX$logFC[i], y = (topX$log10p[i]+3), size=4,
                        label = gsub("S5_genome_", "",topX$Locus[i]), colour="darkorange4")
    }
  }
  return(p)
}



##############################################################################
###### VOLCANO PLOTS OF MOTILITY-ASSOCIATED GENES WITH LABELS
##############################################################################

volcanoMotilityPointsAnnot <- function(dt, annot, cond1, cond2){
  #abd_fld <- "../data/abundances/"
  #dt <- getDGE(abd_fld)
  #cond1 = "LM"
  #cond2 = "SA"
  # Do the normalization according to the dispersions of the genes.

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

  tT.transcripts <- topTags(de.tr, n = nrow(dt), p.value = 0.05)
  det.list <- tT.transcripts$table

  allDF <- det.list
  allDF$Locus <- rownames(allDF)
  rownames(allDF) <- NULL
  allDF$log10p <- -log10(allDF$FDR)
  allExp <- full_join(allDF, annot, by=c("Locus"="Gene_position"))
  allExp <- allExp[!is.na(allExp$Motility_type),]
  ytit <-expression(bold(paste("-",log[10]~" p-value (FDR)")))
  p <-  ggplot(allExp, aes(x=logFC, y=log10p,group=1,label=Gene_name))+
    geom_label(aes(fill = factor(Motility_type)), show.legend=T, size=3, color="black")+
    guides(fill=guide_legend(title=NULL))+
    scale_y_continuous(ytit)+
    scale_x_continuous("Log 2 fold change",
    #                   expand=c(0,0))+
    limits=c(-2.75,2.75))+
# limits = c(-max(abs(allExp$logFC)),max(abs(allExp$logFC))))+
#     limits=c(-max(abs(allExp$logFC))-1,-max(abs(allExp$logFC))+1))+
  scale_fill_manual(name = "Motility type", values=c("tomato", "lightblue","yellowgreen"))+
  #scale_color_manual(name = "Motility type", values=c("red3", "darkblue","forestgreen"))+
    ggtitle(paste0("Volcano plot ", cond2, " vs. ", cond1))+
    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())
  return(p)
}

#########################################################################
###### BARPLOT FOR KEGG PATHWAYS
#########################################################################
# print only the pathways for which at least threshold occurences

keggHisto <- function(dt, annot, path_S5, cond1, cond2, threshold){

    de.tr <- exactTest(dt, pair = c(cond1, cond2))
    tT.transcripts <- topTags(de.tr, n = nrow(dt), p.value = 0.05)
    det.list <- tT.transcripts$table
    det.list$S5_genome_id <- rownames(det.list)
    # suspense
    dataKegg <- inner_join(det.list, path_S5, by="S5_genome_id")

    countPos <- plyr::count(dataKegg[which(dataKegg$logFC>0),], "path_name")
    colnames(countPos) <- c("path_name", "freq.pos")
    countNeg <- plyr::count(dataKegg[which(dataKegg$logFC<0),], "path_name")
    colnames(countNeg) <- c("path_name", "freq.neg")

    allCounts <- full_join(countPos, countNeg, "path_name")

    plotCounts <- melt(allCounts,
                       id.vars = "path_name" )
    colnames(plotCounts) <- c("path_name", "variable", "value")

    plotCounts <- plotCounts[order(plotCounts$value, decreasing=T),]

    if(nrow(plotCounts)>20){
      plotCounts <- plotCounts[c(1:20),]
    }
#     plotCounts <- plotCounts[which(plotCounts$value>threshold),]
    plotCounts <- plotCounts[which(plotCounts$path_name!="<NA>"),]
    tit = paste0("Up- and downregulation by pathway (", cond1, "/", cond2, ")")
    p <- ggplot(plotCounts,
                aes(x=path_name, y=value, fill=factor(variable)))+
      geom_bar(stat="identity", position="dodge")+
      coord_flip()+
      ggtitle(tit)+
      ylab("Number of occurences")+
      #scale_y_continuous(limits = c(0, max(plotCounts$value)))+
      xlab("Pathways")+
      scale_fill_manual(name="",values=c("green", "red"),
                        label=c("Upregulation",
                                "Downregulation")) +
      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=16,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())

    return(p)
}

#########################################################################
###### BARPLOT GO CATEGORIES
#########################################################################
# print only the categories for which at least threshold occurences
# if withMot = T => print motility category bar, regardless of its number of occurence

goHisto <- function(dt, annot, go_S5, cond1, cond2, threshold, withMot=F){

  de.tr <- exactTest(dt, pair = c(cond1, cond2))
  tT.transcripts <- topTags(de.tr, n = nrow(dt), p.value = 0.05)
  det.list <- tT.transcripts$table
  det.list$S5_genome_id <- rownames(det.list)
```

```
  # suspense
  dataGO <- inner_join(det.list, go_S5, by="S5_genome_id")

  countPos <- plyr::count(dataGO[which(dataGO$logFC>0),], "GO.Term")
  colnames(countPos) <- c("GO.Term", "freq.pos")
  countNeg <- plyr::count(dataGO[which(dataGO$logFC<0),], "GO.Term")
  colnames(countNeg) <- c("GO.Term", "freq.neg")

  allCounts <- full_join(countPos, countNeg, "GO.Term")

  plotCounts <- melt(allCounts,
                     id.vars = "GO.Term" )
  plotCounts <- na.omit(plotCounts)
  colnames(plotCounts) <- c("GO.Term", "variable", "value")
  if(withMot){
    plotCounts <- plotCounts[which(plotCounts$value>threshold | plotCounts$GO.Term=="motility"),]
  }else{
    plotCounts <- plotCounts[which(plotCounts$value>threshold),]
  }
  plotCounts <- plotCounts[which(plotCounts$GO.Term!="<NA>"),]
  tit = paste0("Up- and downregulation by GO (", cond2, " vs. ", cond1, ")")
  p <- ggplot(plotCounts,
              aes(x=GO.Term, y=value, fill=factor(variable)))+
    geom_bar(stat="identity", position="dodge")+
    coord_flip()+
    ggtitle(tit)+
    ylab("Number of occurences")+
    #scale_y_continuous(limits = c(0, max(plotCounts$value)))+
    xlab("GO category")+
    scale_fill_manual(name="",values=c("green", "red"),
                      label=c("Upregulation",
                              "Downregulation")) +
    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=16,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())
  return(p)
}

#############################################################################
###### UP- AND DOWNREGULATION OF GENES
#############################################################################
# firstly used 2 y-axis with the right y-axis giving average log CPM (one line on the plot)
# but later, this was removed
# adapted from http://heareresearch.blogspot.ch
# 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
# dataLMSA <- pairTestGenes(dt, annot$Gene_position , "LM", "SA")  #1
# tit="logFC and logCPM (SA vs. LM)"
# annotData <- annot
# dataPair = table for pairwise test
# dataPair <- dataLMSA

fc_barAndCpm_line <- function(dataPair, annotData, gbkData, tit, pt=T){
  #### DATA PREPARATION
  data <- left_join(dataPair, gbkData, by =c("Transcript"="Locus_tag"))
  #nrow(data)  #49
  data %<>% left_join(., annotData, by =c("Transcript"="Gene_position"))

  data$Start %<>% as.character %>% as.numeric  # because stored as factor
  data <- data[order(data$Start),]  # order by starting position

  # set colors of x-axis labels according to motility type
  labCol <- sapply(data$Motility_type, function(x){setMyCol_mot(x)})
  # set colors of line point according to strand
  pointCol  <- sapply(data$Strand, function(x){setMyCol_strand(x)})
```

```
  # set colors of bars according to sign of logFC
  barCol <- sapply(data$logFC, function(x){setMyCol_fc(x)})

  #### PLOT
  grid.newpage()
  # the two plots
  pFC <- ggplot(data, aes(x=as.factor(Start), y=logFC)) +
    scale_y_continuous("log 2 fold change", limits = c(-max(abs(data$logFC)),max(abs(data$logFC))))+
    scale_x_discrete("Genes", labels=data$Gene_name)+
    ggtitle(tit)+
    geom_bar(stat="identity",position = "identity", colour=barCol, fill=barCol) +
    geom_hline(yintercept=1,linetype="dashed", color="gray48")+
    geom_hline(yintercept=-1,linetype="dashed", color="gray48")+
#     theme_few()+
  # theme(axis.text.x = element_text(angle = 90, hjust = 1, colour=labCol,vjust=0.5),
  theme(axis.text.x = element_text(angle = 90, hjust = 1, colour=labCol,vjust=0.5, size=5),
          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_line(color = "red"))
#           panel.grid.minor.x=element_line(colour="black", size=1))


  pCPM <- ggplot(data, aes(x=as.factor(Start),y= logCPM, group=1)) +
    geom_line(colour = "darkorange4") +
#     geom_point(size=2, colour=pointCol)+
    scale_x_discrete("Genes", labels=data$Gene_name)+
    scale_y_continuous("mean log CPM")+
    theme_few()+
    theme(panel.background = element_rect(fill = NA),
          axis.title.y = element_text(colour="darkorange4", angle=90))

  if(pt){
    pCPM <- pCPM + geom_point(size=2, colour=pointCol)
  }

  # extract gtable
  g1 <- ggplot_gtable(ggplot_build(pFC))
  g2 <- ggplot_gtable(ggplot_build(pCPM))

  # overlap the panel of 2nd plot on that of 1st plot
  pp <- c(subset(g1$layout, name == "panel", se = t:r))
  g <- gtable_add_grob(g1, g2$grobs[[which(g2$layout$name == "panel")]], pp$t,
                       pp$l, pp$b, pp$l)

  # axis tweaks
  ia <- which(g2$layout$name == "axis-l")
  ga <- g2$grobs[[ia]]
  ax <- ga$children[[2]]
  ax$widths <- rev(ax$widths)
  ax$grobs <- rev(ax$grobs)
  ax$grobs[[1]]$x <- ax$grobs[[1]]$x - unit(1, "npc") + unit(0.15, "cm")
  g <- gtable_add_cols(g, g2$widths[g2$layout[ia, ]$l], length(g$widths) - 1)
  g <- gtable_add_grob(g, ax, pp$t, length(g$widths) - 1, pp$b)

  ia2 <- which(g2$layout$name == "ylab")
  ga2 <- g2$grobs[[ia2]]
  ga2$rot <- 90
  g <- gtable_add_cols(g, g2$widths[g2$layout[ia2, ]$l], length(g$widths) - 1)
  g <- gtable_add_grob(g, ga2, pp$t, length(g$widths) - 1, pp$b)

 return(pFC)
#return(g)
}



##########################################################################
###### BLOXPLOT FOR ALL CONDITIONS, BY MOTILITY TYPE
##########################################################################

#data = getMeanData(dt$counts)
#data$Transcript <- rownames(data)
#data = left_join(annot, data, by=c("Gene_position" = "Transcript"))
```

```
boxplotMotGenes <- function(data, annot, maintit, chemo=F, allRep = F){
  meltData <- melt(data, id=c("Gene_position", "Gene_name", "Motility_type"))
  meltData$value <- log(meltData$value)
  meltData$Var <- substr(meltData$variable, 1, 2)
  if(chemo){
  fillCol <-c( "orangered2", "dodgerblue3", "forestgreen","thistle")
  } else{
    fillCol <- c( "orangered2", "dodgerblue3", "forestgreen")
  }
  if(allRep){
    my_x <- "variable"
  }else{
    my_x <- "Var"
  }
  p <-ggplot(meltData, aes_string(x=my_x, y="value", fill="Motility_type")) +
    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)
  return(p)
}

###########################################################################
###### MULTIPLOT GGPLOT
###########################################################################

# used for plotting multiple ggplots on the same window
# Source : http://www.cookbook-r.com/
# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
```

```
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}


#########################################################################
###### RDA PLOT
#########################################################################
# used to plot RDA result

rdaFct <- function(mes, env){
  # where mes is the response matrix and env the explanatory variables
  mes<-as.data.frame(scale(mes))
  env<-as.data.frame(scale(env))
  # rda(Y,X,W) where Y is the response matrix,X is the matrix of explanatory variables
  # and W is an optional matrix of covariables
  rda.site<-rda(mes~.,env)   #same as : rda(mes,env), but need formula for anova
  #summary(rda.site)
  coef(rda.site)
  # percentage of variance explained by axis 1
  a1 <- rda.site$CCA$eig[1]/rda.site$tot.chi
  xl =paste0("RDA1 - ", round(a1*100,2), "%")
  # percentage of variance explained by axis 2
  a2 <- rda.site$CCA$eig[2]/rda.site$tot.chi
  yl =paste0("RDA2 - ", round(a2*100,2), "%")
  # R-squared and adjusted-R2
  (R2 <- RsquareAdj(rda.site)$r.squared)
  (R2_adj <- RsquareAdj(rda.site)$adj.r.squared)
  plot(rda.site, xlab=xl, ylab=yl)
}


#########################################################################
###### GENO PLOT R
#########################################################################
# used to plot RDA result
# used to compare genome position of motility-associated genes Pseud. S5 and Pseud. f. Pf5

geneMapMot <- function(P_mot,agl=45, mt="") {
  names1 <- P_mot$fonc_short_pf
  names2 <- P_mot$fonc_short_pf
  starts1 <- as.numeric(as.character(P_mot$Start_ps))
  starts2 <- as.numeric(as.character(P_mot$start_pf))
  ends1 <- as.numeric(as.character(P_mot$End_ps))
  ends2 <- as.numeric(as.character(P_mot$end_pf))
  strands1 <- as.character(P_mot$Strand_ps)
  strands2 <- as.character(P_mot$strand_pf)
  strands1 <- sapply(strands1, numStrand)
  strands2 <- sapply(strands2, numStrand)

  #cols1 <- palette(rainbow(87))
  cols1 <- colorRampPalette(c("red", "blue"))(length(starts1))
  cols2 <- colorRampPalette(c("red", "blue"))(length(starts2))
  df1 <- data.frame(name=names1, start=starts1, end=ends1, strand=strands1, col=cols1)
  dna_seg1 <- dna_seg(df1)

  df2 <- data.frame(name=names2, start=starts2, end=ends2, strand=strands2, col=cols2)
  dna_seg2 <- dna_seg(df2)
  #plot_gene_map(list(dna_seg1, dna_seg2))
  df3 <- df1
  colnames(df3) %<>% paste0(., "1")
  df4 <- df2
  colnames(df4) %<>% paste0(., "2")
  df5 <- cbind(df3,df4)
  df5 <- df5[,c(2,3,7,8)]
```

```
  df5$col <- colorRampPalette(c("red", "blue"))(nrow(df5))
  comparison1 <- as.comparison(df5)
  dna_segs = list(dna_seg1, dna_seg2)
  mid_pos <- middle(dna_segs[[1]])

  annot <-  annotation(x1=mid_pos, x2=rep(NA, length(mid_pos)), rot=rep(agl, length(mid_pos)),
                       text=dna_segs[[1]]$name)

  plot_gene_map(dna_segs=dna_segs, comparisons=list(comparison1), annotations=annot, scale=T,
                annotation_cex=0.9, main = mt,
                dna_seg_labels=c("Pseudomonas S5","Pseudomonas fluorescens Pf5"))
}


#########################################################################
###### PCA plots - Gillet & Borcard 2012
#########################################################################

"cleanplot.pca" <- function(res.pca, mycol="black",ax1=1, ax2=2, point=FALSE,
                            ahead=0.07, cex=0.7)
{
  # A function to draw two biplots (scaling 1 and scaling 2) from an object
  # of class "rda" (PCA or RDA result from vegan's rda() function)
  #
  # License: GPL-2
  # Authors: Francois Gillet & Daniel Borcard, 24 August 2012
  # MODIFICATION MZUFFEREY: add % for each axis
  require("vegan")

  e1 <- round((res.pca$CA$eig[ax1]/sum(res.pca$CA$eig)*100),2)
  e2 <- round((res.pca$CA$eig[ax2]/sum(res.pca$CA$eig)*100),2)

  xl <- paste0("PC",ax1, " - ", e1, "%")
  yl <- paste0("PC",ax2, " - ", e2, "%")

  par(mfrow=c(1,2))
  p <- length(res.pca$CA$eig)

  # Scaling 1: "species" scores scaled to relative eigenvalues
  sit.sc1 <- scores(res.pca, display="wa", scaling=1, choices=c(1:p))
  spe.sc1 <- scores(res.pca, display="sp", scaling=1, choices=c(1:p))
  plot(res.pca, choices=c(ax1, ax2), display=c("wa", "sp"), type="n",
       main="PCA - scaling 1", scaling=1, xlab=xl, ylab=yl)
  if (point)
  {
    points(sit.sc1[,ax1], sit.sc1[,ax2], pch=20, col=mycol)
    text(res.pca, display="wa", choices=c(ax1, ax2), cex=cex, pos=3, scaling=1, col=mycol)
  }
  else
  {
    text(res.pca, display="wa", choices=c(ax1, ax2), cex=cex, scaling=1, col=mycol)
  }
  text(res.pca, display="sp", choices=c(ax1, ax2), cex=cex, pos=4,
       col="red", scaling=1)
  arrows(0, 0, spe.sc1[,ax1], spe.sc1[,ax2], length=ahead, angle=20, col="red")
  pcacircle(res.pca)

  # Scaling 2: site scores scaled to relative eigenvalues
  sit.sc2 <- scores(res.pca, display="wa", choices=c(1:p))
  spe.sc2 <- scores(res.pca, display="sp", choices=c(1:p))
  plot(res.pca, choices=c(ax1,ax2), display=c("wa","sp"), type="n",
       main="PCA - scaling 2",xlab=xl, ylab=yl, col=mycol)
  if (point) {
    points(sit.sc2[,ax1], sit.sc2[,ax2], pch=20, col=mycol)
    text(res.pca, display="wa", choices=c(ax1 ,ax2), cex=cex, pos=3, col=mycol)
  }
  else
  {
    text(res.pca, display="wa", choices=c(ax1, ax2), cex=cex, col=mycol)
  }
  text(res.pca, display="sp", choices=c(ax1, ax2), cex=cex, pos=4, col="red")
  arrows(0, 0, spe.sc2[,ax1], spe.sc2[,ax2], length=ahead, angle=20, col="red")
}
```

```
"pcacircle" <- function (pca)
{
  # Draws a circle of equilibrium contribution on a PCA plot
  # generated from a vegan analysis.
  # vegan uses special constants for its outputs, hence
  # the 'const' value below.

  eigenv <- pca$CA$eig
  p <- length(eigenv)
  n <- nrow(pca$CA$u)
  tot <- sum(eigenv)
  const <- ((n - 1) * tot)^0.25
  radius <- (2/p)^0.5
  radius <- radius * const
  symbols(0, 0, circles=radius, inches=FALSE, add=TRUE, fg=2)
}



#*****************************************************************************************
#*****************************************************************************************
# FURTHER FUNCTIONS FINALLY NOT USED NEITHER FOR THE REPORT NOR FOR THE SUPP. DATA
#*****************************************************************************************
#*****************************************************************************************

#************************************************
#################### PLOT 1: line for 1 condition
#************************************************
# plot line of the logFC value for comparison between given 2 conditions, for motility-associated genes
# (in fact not useful because can be done with next function ...)
## plot for logfc data for a given pair of conditions
# separately for each test
# x-axis: genes ordered by their position along chromosome
# y-axis: logFC
# dots colored by significance, x-axis label by motility type
# with line connecting the dots
# annotData -> contains match between transcript ID and gene ID
# gbkData -> contains at least the 2 columns "Start" "Locus_tag" (=transcript ID)
# tit -> string with title for the plot
# returns the created plot

oneTestLinePlot <- function(data, gbkData, annotData, tit){

  data %<>% left_join(., gbkData, by =c("Transcript"="Locus_tag"))

  data %<>% left_join(., annotData, by =c("Transcript"="Gene_position"))

  data$Start %<>% as.character %>% as.numeric  # because stored as factor
  data <- data[order(data$Start),]  # order by starting position

  # set colors of x-axis labels according to motility type
  labCol <- sapply(data$Motility_type, function(x){setMyCol_mot(x)})

  # set colors of the points according to significance (FDR)
  pointCol  <- sapply(data$FDR, function(x){setMyCol_fdr(x)})

  p <- ggplot(data, aes(x=as.factor(Start), y=logFC, group=1))+
    geom_point(size=3, colour=pointCol)+
    geom_line(colour="hotpink")+
    scale_y_continuous("Log2 fold change")+
    scale_x_discrete("Motility genes", labels=data$Gene_name)+
    ggtitle(tit)+
    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,
colour=labCol),
          plot.title =  element_text(colour="darkslateblue", size=15),
          panel.grid.minor.y=element_blank(),
          panel.grid.major.y=element_blank())
  return(p)
}
```

```
#************************************************
#################### PLOT 2: many lines in one plot (log2FC for pairs of conditions)
#************************************************
# plot the lines of logFC values for all comparisons for motility-associated genes
# same parameters as the previous function
# used to plot in the same figure more than one pair of conditions tested
# x-axis: genes ordred by chromosomal position
# y-axis log2fc
# colour of the dots: green if significant (FDR), else gray

manyLinePlot <- function(data, gbkData, annotData, tit){
  data %<>% left_join(., gbkData, by =c("Transcript"="Locus_tag"))
  #nrow(data)  #298
  data %<>% left_join(., annotData, by =c("Transcript"="Gene_position"))

  data$Start %<>% as.character %>% as.numeric  # because stored as factor
  data <- data[order(data$Start),]  # order by starting position

  # need "unique" for the labels (and their colors) of the x axis
  # because x-axis in ggplot is the start position
  # duplicated because of different conditions
  uniqData <- unique(data[,c("Start", "Motility_type", "Gene_name")])
  # nrow(uniqData) # 82
  # -> ok, tested with labels=1:82 in ggplot axis.text.x

  # set colors of x-axis labels according to motility type
  labCol <- sapply(uniqData$Motility_type, function(x){setMyCol_mot(x)})

  # set colors of the points according to significance (FDR)
  # is.numeric(data$FDR) # TRUE -> ok
  pointCol  <- sapply(data$FDR, function(x){setMyCol_fdr(x)})

  p <- ggplot(data, aes(x=as.factor(Start), y=logFC, group=Pair))+
    geom_point(size=3, colour=pointCol, aes(group=Pair))+
    geom_line(aes(colour=Pair,group=Pair))+
    scale_color_discrete(name="Tested pairs")+
    scale_y_continuous("Log2 fold change")+
    # scale_x_discrete("Motility genes", labels=1:82)+
    scale_x_discrete("Motility genes", labels=uniqData$Gene_name)+
    ggtitle(tit)+
    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,
colour=labCol),
          plot.title =  element_text(colour="darkslateblue", size=15),
          legend.title = element_text(face="bold"),
          panel.grid.minor.y=element_blank(),
          panel.grid.major.y=element_blank())

  return(p)
}

#************************************************
#################### PLOT 3: smear plot for 2 given conditions, colour by motility
#************************************************
# custom smearplot function (at the end used built-in edgeR function in the supp. data)

smearPlotsAllPoints <- function(dt, annot, cond1, cond2){

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

  # allDF <- tT.transcripts$table   # => no one near 0 log2fc
  allDF <- de.tr$table
  allDF$Locus <- rownames(allDF)
  rownames(allDF) <- NULL
  allExp <- full_join(allDF, annot, by=c("Locus"="Gene_position"))
  colPoints <- sapply(allExp$Motility_type, function(x){setMyCol_mot(x)})
  lm <- max(abs(allExp$logFC))
  p <-  ggplot(allExp, aes(x=logCPM, y=logFC,group=1))+
    geom_point(size=2, colour=colPoints)+
    scale_y_continuous("Log 2 fold change",limits = c(-lm,lm))+
    scale_x_continuous("Average logCPM")+
```

```
    ggtitle(paste0("Smear plot ", cond1, "/", cond2))+
    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())
  return(p)
}

#************************************************
##### PLOT 4 : smear plot with colour for given 2 conditions to test - motility-associated genes only
#************************************************
# custom smearplot function (at the end used built-in edgeR function in the supp. data)
# plot only motility-associated genes

smearPlotsMotility <- function(dt, annot, cond1, cond2){
  de.tr <- exactTest(dt, pair = c(cond1, cond2))
  allDF <- de.tr$table
  allDF$Locus <- rownames(allDF)
  rownames(allDF) <- NULL
  allExp <- full_join(allDF, annot, by=c("Locus"="Gene_position"))
  allExp <- allExp[!is.na(allExp$Motility_type),]
  lm <- max(abs(allExp$logFC))
  p <-  ggplot(allExp, aes(x=logCPM, y=logFC,group=1))+
    geom_point(aes(color=factor(Motility_type)),size=2)+
    scale_y_continuous("Log 2 fold change",limits = c(-lm,lm))+
    scale_x_continuous("Average logCPM")+
    scale_color_manual(name = "Motility type", values=c("red3", "forestgreen", "darkblue"))+
    ggtitle(paste0("Smear plot ", cond1, "/", cond2))+
    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())
  return(p)
}

#************************************************
##### PLOT 5 : idem plot 4 but with labels
#************************************************
# custom smearplot function (with labels instead of points)

smearPlotsMotility_label <- function(dt, annot, cond1, cond2){
  de.tr <- exactTest(dt, pair = c(cond1, cond2))
  allDF <- de.tr$table
  allDF$Locus <- rownames(allDF)
  rownames(allDF) <- NULL
  allExp <- full_join(allDF, annot, by=c("Locus"="Gene_position"))
  allExp <- allExp[!is.na(allExp$Motility_type),]
  lm <- max(abs(allExp$logFC))
  p <-  ggplot(allExp, aes(x=logCPM, y=logFC,group=1, label=Gene_name))+
    geom_label(aes(fill = factor(Motility_type)), show.legend=T)+
    guides(fill=guide_legend(title=NULL))+
    scale_y_continuous("Log 2 fold change",limits = c(-lm,lm))+
    scale_x_continuous("Average logCPM")+
    ggtitle(paste0("Smear plot ", cond1, "/", cond2))+
    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())
  return(p)
}

#************************************************
##### PLOT 6 : volcano plot for motility associated genes
#************************************************
# volcano plot for motility-associated genes (points not labels)

volcanoMotilityPoints<- function(dt, annot, cond1, cond2){

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

  tT.transcripts <- topTags(de.tr, n = nrow(dt), p.value = 0.05)
  det.list <- tT.transcripts$table

  allDF <- det.list
  allDF$Locus <- rownames(allDF)
  rownames(allDF) <- NULL
  allDF$log10p <- -log10(allDF$FDR)
  allExp <- full_join(allDF, annot, by=c("Locus"="Gene_position"))
  allExp <- allExp[!is.na(allExp$Motility_type),]
  p <-  ggplot(allExp, aes(x=logFC, y=log10p,group=1))+
    geom_point(aes(color=factor(Motility_type), name="Motility type"),size=2)+
    scale_y_continuous("Log 2 fold change")+
    scale_x_continuous("Average logCPM")+
    scale_color_manual(name = "Motility type", values=c("red3", "forestgreen", "darkblue"))+
    ggtitle(paste0("Smear plot ", cond1, "/", cond2))+
    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())
  return(p)
}

#************************************************
##### PLOT 7 : BARPLOT FC
#************************************************
# barplot similar to the one used for the report

fc_bar <- function(dataPair, annotData, gbkData, tit){
  #### DATA PREPARATION
  data <- left_join(dataPair, gbkData, by =c("Transcript"="Locus_tag"))
  #nrow(data)  #49
  data %<>% left_join(., annotData, by =c("Transcript"="Gene_position"))

  data$Start %<>% as.character %>% as.numeric  # because stored as factor
  data <- data[order(data$Start),]  # order by starting position

  # set colors of x-axis labels according to motility type
  labCol <- sapply(data$Motility_type, function(x){setMyCol_mot(x)})

  # set colors of bars according to sign of logFC
  barCol <- sapply(data$logFC, function(x){setMyCol_fc(x)})

  #### PLOT
  pFC <- ggplot(data, aes(x=as.factor(Start), y=logFC)) +
    scale_y_continuous("log 2 fold change", limits = c(-max(abs(data$logFC)),max(abs(data$logFC))))+
    scale_x_discrete("Genes", labels=data$Gene_name)+
    ggtitle(tit)+
    geom_bar(stat="identity",position = "identity", colour=barCol, fill=barCol) +
    theme_few()+
    theme(axis.text.x = element_text(angle = 90, hjust = 1, colour=labCol),
          plot.title =  element_text(colour="darkslateblue", size=15))

  return(pFC)
}
```

```
#************************************************
##### Plot 8: word clouds
#************************************************
# used for the presentation 1st slide
# adapted from https://georeferenced.wordpress.com

wordCld <- function(x, myStopW="",minF=1, maxW=200){
  text <- as.character(x)
  # corpus = liste de documents
  # Charger les données comme un corpus
  docs <- Corpus(VectorSource(text))
  toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
  docs <- tm_map(docs, toSpace, "/")
  docs <- tm_map(docs, toSpace, "@")
  docs <- tm_map(docs, toSpace, "\\|")
  # convert lower case
  docs <- tm_map(docs, content_transformer(tolower))
  # remove numbers
  docs <- tm_map(docs, removeNumbers)
  # remove stop words english
  docs <- tm_map(docs, removeWords, stopwords("english"))
  # remove stop words passed in parameters
  docs <- tm_map(docs, removeWords, myStopW)
  # remove punctuation signs
  docs <- tm_map(docs, removePunctuation)
  # remove white spaces
  docs <- tm_map(docs, stripWhitespace)

  dtm <- TermDocumentMatrix(docs)
  m <- as.matrix(dtm)
  v <- sort(rowSums(m),decreasing=TRUE)
  d <- data.frame(word = names(v),freq=v)
  wordcloud(words = d$word, freq = d$freq, min.freq = minF,
            max.words=maxW, random.order=FALSE, rot.per=0.35, random.color=F,
            colors=c
('#9e0142','#d53e4f','#f46d43','#fdae61','#fee08b','#e6f598','#abdda4','#66c2a5','#3288bd','#5e4fa2'))
  # author colors:
  # palette(rainbow(8)))
  # colors=brewer.pal(12, "Spectral"))
  #colorRampPalette(c("red", "blue"))(20))#brewer.pal(8, "Dark2"))
  # palette(rainbow(6))     # six color rainbow
  # (palette(gray(seq(0,.9,len = 25)))) #grey scale
}
```

```
################################################################################
##### RNA-SEQ DATA ANALYSIS - PSEUDOMONAS S5 - MAIN FIGURES OF THE REPORT
################################################################################
##### Spring 2016 - MLS - UNIL - Marie Zufferey
##### !!! some hard-coded parameters, file shape and formats not checked
rm(list=ls())
setwd("PATH_TO_DIRECTORY")
outfolder = "YOUR_OUTFOLDER"
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)

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

annot <- read.csv("../data/annot_mot.csv", sep=",")
annot$Gene_position!="S5_genome_1619"
annot <- annot[-which(annot$Gene_position=="S5_genome_4011"),]
annot <- annot[-which(annot$Gene_position=="S5_genome_1619"),]
rawannot <- read.csv("../data/annot_mot.csv", sep=",")
rawannot <- annot[-which(rawannot$Gene_position=="S5_genome_1619"),]
S5_stat <- read.csv("../data/Pseud_S5_stat.txt", sep="\t")
gbkData <- read.csv("../data/S5_gbk_short.csv", sep=",")
abd_fld <- "../data/abundances/"
dt <- getDGE(abd_fld)  # compute it once here # normalized for dispersion !!!
# dt after  estimateTagwiseDisp(dt)  !!!!


##### 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 <- annot[-which(annot$Gene_position=="S5_genome_4011"),]
annot <- annot[-which(annot$Gene_position=="S5_genome_1619"),]

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)

# 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$Gene_position , "SA", "WR")  #5
dataSALM <- pairTestGenes(dt, annot$Gene_position , "SA", "LM")  #1b
dataWLWR <- pairTestGenes(dt, annot$Gene_position , "WL", "WR")  #6
dataWLSA <- pairTestGenes(dt, annot$Gene_position , "WL", "SA")  #4b
dataWLLM <- pairTestGenes(dt, annot$Gene_position , "WL", "LM")  #4b


#*****************************************
# MATRIX OF PLOTS
#*****************************************
# 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)
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', .)

png(paste0(outfolder,"/scatterplotMatrix_all.png"))
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)
dev.off()

# => we choose cond1=LM, cond2=SA
# => and cond1=SA, cond2=WR

#*******************************************
# VOLCANO PLOTS WITH ALL DATA
#*******************************************
# with label for the top 5 genes
png(paste0(outfolder,"/volcanoAll_LMSA.png"))
volcanoAllPoints(dt, annot, "LM", "SA", plotAnnot=T, myT=3)  %>% plot
dev.off()
png(paste0(outfolder,"/volcanoAll_SAWR.png"))
volcanoAllPoints(dt, annot, "SA", "WR")  %>% plot
dev.off()

#*******************************************
# VOLCANO PLOTS WITH MOTILITY ASSOCIATED GENES
#*******************************************

svg(paste0(outfolder,"/volcanoMot_LMSA.svg"))
volcanoMotilityPointsAnnot(dt, annot, "LM", "SA")
dev.off()

svg(paste0(outfolder,"/volcanoMot_SAWR.svg"))
volcanoMotilityPointsAnnot(dt, annot, "SA", "WR")
dev.off()

#*******************************************
# BOX PLOT FOR MOTILITY ASSOCIATED GENES
#*******************************************
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"))

# WITH OWN DEFINED "MYRPKM" **********************
motRP <- myrpkm(mot_data[,grep("1|2|3|4", colnames(mot_data))], mot_data$Length)

# WITH edgeR "RPKM" **********************
# get DGE object
dt <- getDGE(abd_fld)
dt <- calcNormFactors(dt)
temp <- dt$counts

# retrieve the length
getN <- temp
getN %<>% as.data.frame
getN$Tr <- rownames(getN)
getN <- left_join(getN, S5_stat, by=c("Tr"="Seq_tag"))

# rpkm
temp <- rpkm(temp, getN$Length)
temp %<>% as.data.frame
#temp$Tr <- rownames(temp)
temp <- temp[which(rownames(temp) %in% mot_data$Gene_position),]
temp <- temp[match(mot_data$Gene_position, rownames(temp)),]
motRP <- temp

# we want to compare across genes and across conditions -> RPKM
```

```
#
mot_data[,grep("1|2|3|4", colnames(mot_data))]  <- motRP

# 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"),]

png(paste0(outfolder,"/boxplot_Mot.png"))
boxplotMotGenes(data_mot, annot, "Global expression mot. genes", chemo=F)
dev.off()

png(paste0(outfolder,"/boxplot_withChem.png"))
boxplotMotGenes(mot_data2, annot, "Global expression mot. genes (with chemo.)", chemo=T)
dev.off()

#*******************************************
# LINE PLOTS WITH MOTILITY ASSOCIATED GENES
#*******************************************
#************ cond1=SA, cond2=WL
# Draw it for SAWL, motility associated genes
dataSAWR <- pairTestGenes(dt, annot$Gene_position , "SA", "WR")  #1
tit <- "log 2 FC (WR vs. SA - motility associated genes)"
png(paste0(outfolder,"/2axis_SAWR_mot.png"))
fc_barAndCpm_line(dataSAWR, annot, gbkData, tit, pt=F) %>% grid.draw
dev.off()

dataLMSA <- pairTestGenes(dt, annot$Gene_position , "LM", "SA")  #1
tit <- "log 2 FC (SA vs. LM - motility associated genes)"
png(paste0(outfolder,"/2axis_LMSA_mot.png"))
fc_barAndCpm_line(dataLMSA, annot, gbkData, tit,pt=F) %>% grid.draw
dev.off()
```