Clustering of neuron responses

- 1 Cluster analysis.
  - 1.1 Assessment of clustering tendency in neuron responses.
  - 1.2 Identify optimal cluster number
  - 1.3 Stability of identified clusters
- 2 Assessment of variabilty within and between Animals and Clusters
  - 2.1 Analysis of Variance within and between Animals and Clusters
  - 2.2 Principal component analysis
- 3 References
- 4 Session Information

In this document we present further details of analysis on both the clustering presented in Jove et al and the variability of measurements between animals compared to those between defined neuron reponse clusters.

The rMarkdown file associated with this report and all required data can be found in the Jove_Vosshall_2020 repository Github https://github.com/VosshallLab/Jove_Vosshall_2020 (https://github.com/VosshallLab/Jove_Vosshall_2020).

1 Cluster analysis.

1.1 Assessment of clustering tendency in neuron responses.

Prior to investigation of neuron response patterns and clusters within the dataset, the dataset was evaluated to assess whether it was amenable to clustering. To investigate this, we used the Hopkins statistic to evaluate the clustering tendency of the neuron responses (Lawson R and Jurs P 1990).

Briefly, the Hopkins statistic is defined as below where x_i is the distance between closest neighbouring points in the real data ($x_i = dist(p_i, p_j)$) and the y_i represents the distance between closest neighbouring points in the simulated data ($y_i = dist(q_i, q_j)$).

The Hopkins statistic (H) is then calculated from $\sum_{i=1}^{n} y_i$ over the sum of $\sum_{i=1}^{n} x_i$ and $\sum_{i=1}^{n} y_i$

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

In datasets which are not amenable to clustering, the distances between neighbouring closest points will be close to the random dataset and the Hopkins statistic will approach 0. In a dataset with clusters present, the distances between neighbouring closest points will be low compared to the random dataset and the Hopkins statistic will be closer to 1.

We calculated the Hopkins statistic for our dataset using the factoextra R package and the get_clust_tendency function (Kassambara A and Mundt F. 2020). To show the significance of this clustering tendancy, the p-value for the Hopkins statistic was calculated using the beta distribution in base R (Adolfsson A et al 2018).

```
require(FactoMineR)
require(factoextra)
require(fitdistrplus)
nBoot <- 80
Res_Hopkins <- get_clust_tendency(t(swsw$Values), n = nBoot)

hopkins_stat_pvalue <- pbeta(Res_Hopkins$hopkins_stat,
                                  nBoot,nBoot,
                                  lower.tail = FALSE)

data.frame("Hopkins Statistic"=Res_Hopkins$hopkins_stat,
           "Hopkins Statistic_P-value"=hopkins_stat_pvalue) %>%
  datatable
```

Show 10 entries

Search:

	Hopkins.Statistic	Hopkins.Statistic_P.value

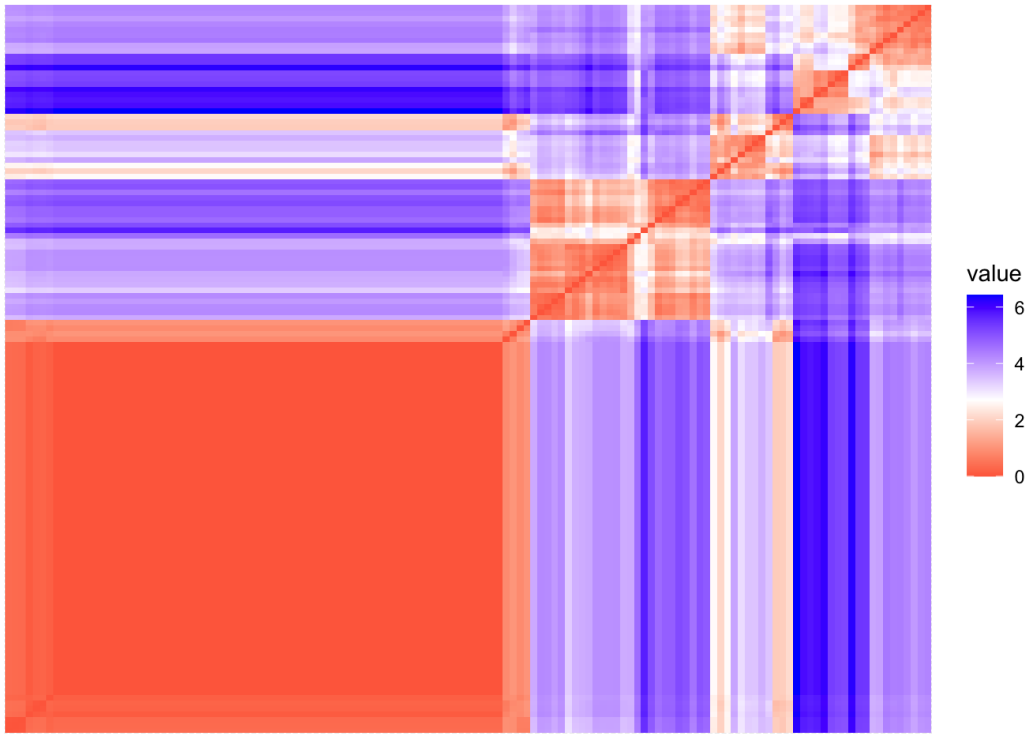
	Hopkins.Statistic	Hopkins.Statistic_P.value
1	0.904693221215838	4.01259038982349e-39

Showing 1 to 1 of 1 entries

The derived Hopkins statistic of **0.9046932** showed the dataset contains suitable information for clustering.

The accompanying ordered dissimilarity image below illustrates the non-random structure in the dataset and vizualy suggests the presence of 5 clusters.

```
Res_Hopkins$plot
```



1.2 Identify optimal cluster number

Having demonstrated the dataset has a high clustering tendency, the optimal number of clusters within our dataset was established using the Silhouette method (Rousseeuw PJ. 1987).

The Silhouette method evaluates the similarity of cluster members to the similarity between clusters as below.

For all neurons, the dissimilarity for a cluster member to its own cluster , a_i, is calculated as the mean distance between a cluster member and all other members of that cluster. Further to this the minimun, mean dissimilarity of the cluster member to members of other clusters is calculated, b_i.

The silhouette width for each cluster member can then be calculated using the below formula.

$S_i = (b_i - a_i)/max(a_i, b_i)$

We performed Silhouette analysis using the NbClust R package (Charrad M et al, 2014) with potential cluster numbers in the range of 2 to 10 and selected the cluster number with the highest mean silhouette value across clusters.

```
require(NbClust)
require(cluster)

k <- NbClust(t(swsw$Values),distance = "euclidean",
          min.nc = 2, max.nc = 10,
          method = "complete", index ="silhouette")

# all(k$Best.partition == swsw$CllusterInfo$CutMembers)

data.frame("Optimal Cluster Number",k$Best.nc[1]) %>%
  set_colnames(value=NULL) %>%
  datatable(colnames = NULL,rownames = NULL)
```

Show 10 entries Search:

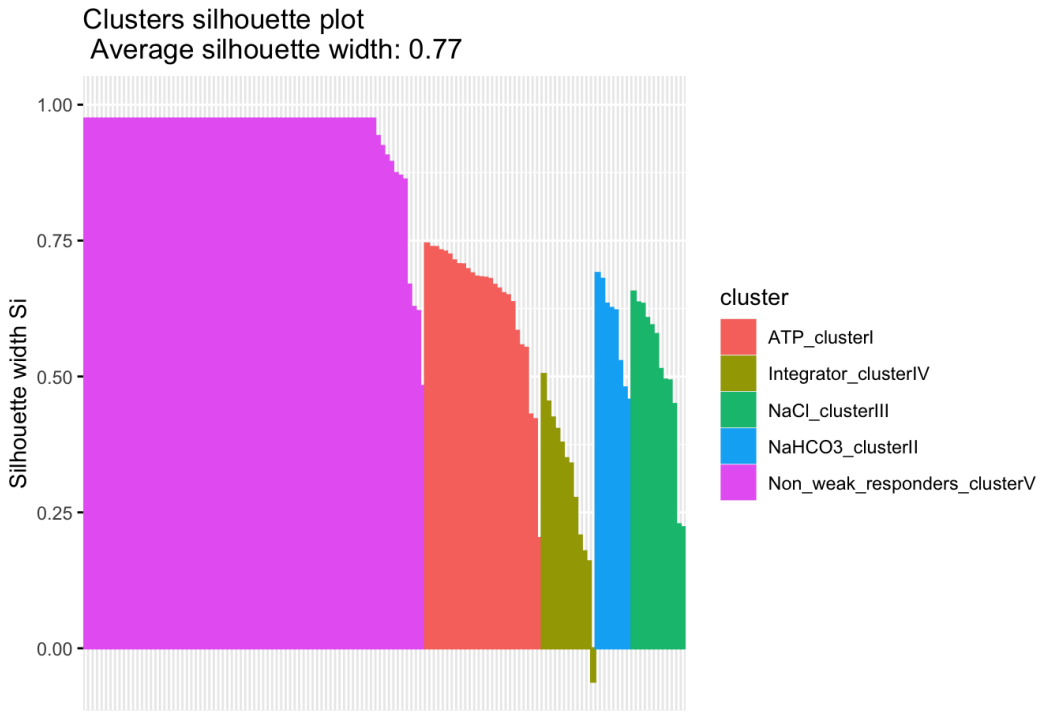
	Hopkins.Statistic	Hopkins.Statistic_P.value
1	0.904693221215838	4.01259038982349e-39

Showing 1 to 1 of 1 entries Previous 1 Next

From the analysis of putative clusters, the optimal cluster numbers was **5** and the mean silhouette width was **0.7694**

We can visualise the result of our silhouette analysis using the factoextra package to show the distibution of silhouette widths for all members of each cluster. From the visualisation it can be seen that one single neuron in cluster 3 can be considered mis-clustered with a silhouette width less than 0.

```
require("cluster")
sil <- silhouette(cutree(swsw$CllusterInfo$tree_col,5),dist(t(swsw$Values)))
# all(sil[,1] == swsw$CllusterInfo$CutMembers)
p <- fviz_silhouette(sil,print.summary = FALSE,label = FALSE)
p$layers <- p$layers[1]
# Update numbers to Cluster IDs
p$data[,1] <- factor(clusterMap$Cluster[match(paste0("Cluster_",p$data[,1]),clusterMap$ID)])
p
```



1.3 Stability of identified clusters

Following the selection of the optimal number of clusters, we can assess the stability of these clusters. To evaluate the stability of these clusters we assess the boostrap distribution of the Jaccard coefficient of resampled versus original data (Hennig C. 2007, Hennig C. 2008).

Clusters showing a Jaccard bootstrap mean of less than 0.5 can be considered unstable and so unreliable and an average Jaccard bootstrap mean across clusters above 0.85 shows a highly stable clustering (Hennig C. 2007).

Here we used the fpc R package's clusterboot function (Hennig C. 2020) following the recommendations of 100 bootstraps to calculate the Jaccard bootstrap mean for all clusters.

```
require(fpc)
# cpx <- clusterboot(t(swsw$Values),k=5,B=10000,clustermethod=hclustCBI,method="complete",scale=FALSE)
cpx <- clusterboot(t(swsw$Values),k=5,B=100,clustermethod=hclustCBI,
                   method="complete",
                   scaling=FALSE,
                   seed = 1)

# all(cpx$partition == swsw$CllusterInfo$CutMembers)
```

All clusters identified have Jaccard bootstrap mean values above 0.7 indicating a set of stable of clusters and an average Jaccard bootstrap mean across clusters of **0.8727142**.

```
data.frame(Cluster2=paste0("Cluster_",1:5),Jaccard_Boostrap_Mean=cpx$bootmean) %>%
  merge(clusterMap,by=1) %>% dplyr::select(-Cluster2) %>% dplyr::select(Cluster,everything()) %>%
  datatable
```

Show 10 **entries** **Search:**

	Cluster	Jaccard_Boostrap_Mean
1	Non_weak_responders_clusterV	0.980854332966848
2	ATP_clusterI	0.991826281389749
3	Integrator_clusterIV	0.704919302919303
4	NaHCO3_clusterII	0.892855006105006
5	NaCl_clusterIII	0.793116231807408

Showing 1 to 5 of 5 entries Previous 1 Next

2 Assessment of variabilty within and between Animals and Clusters

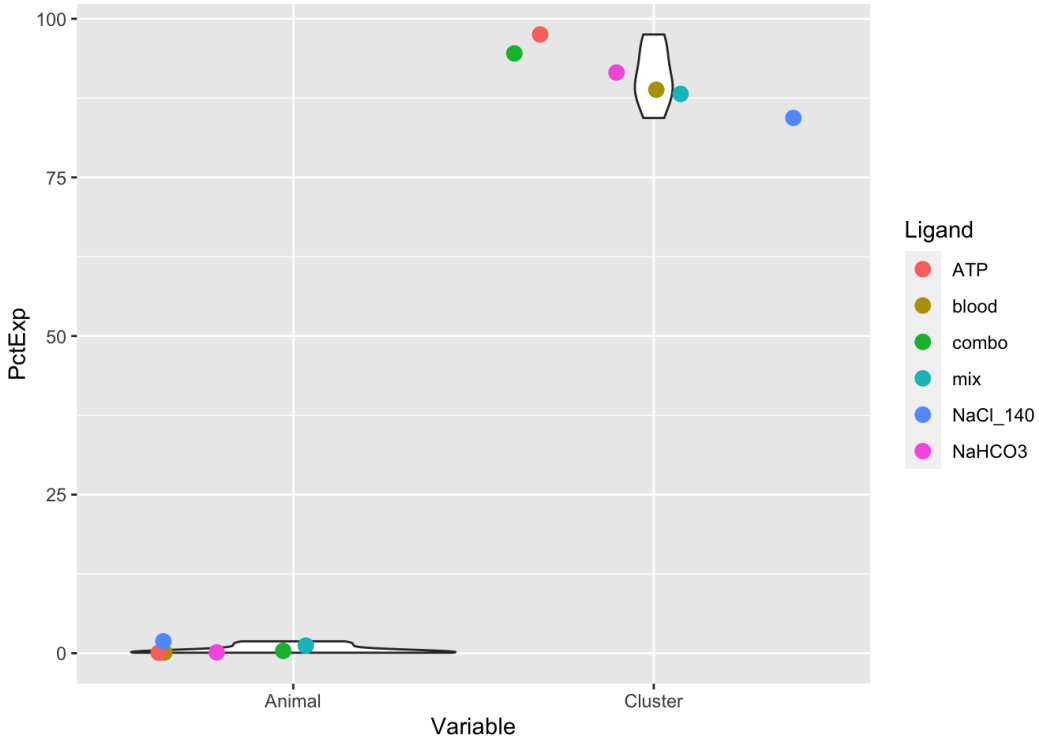
The hierarchical clustering approach used in this study is agnostic to animal information and therefore was assessed to ensure no biases in clustering are associated to animal specific effects.

2.1 Analysis of Variance within and between Animals and Clusters

To assess the variance between/within animals and clusters, an analysis of variance was performed on the response variables with both animal and cluster included as factors in the model.

Below we perform the analysis of variance in base R.

```
require(DT)

VariableModelRes <- list()
VariableModelResWithCluster <- list()
for(v in 1:length(rownames(swsw$Values))){
ModelVaribles <- data.frame(Response=swsw$Values[v,,drop=TRUE] %>% unlist,
                                          Cluster=swsw$CllusterInfo$CutMembers$Cluster,
                                          Animal=rownames(swsw$CllusterInfo$CutMembers) %>%
                                            gsub("_Neuron.*","",.) %>%
                                            gsub("Movie","Animal",.)
                                          %>% factor)

modelFit_ReponseAndAnimal <- lm(ModelVaribles,formula = Response~Animal)
af <- anova(modelFit_ReponseAndAnimal)
afss <- af$"Sum Sq"
VariableModelRes[[v]] <- cbind(af,PctExp=afss/sum(afss)*100)[-2,5:6]

modelFit_ReponseAndAnimalAndCluster <- lm(ModelVaribles,formula = Response~Cluster+Animal)
af <- anova(modelFit_ReponseAndAnimalAndCluster)
afss <- af$"Sum Sq"
VariableModelResWithCluster[[v]] <- cbind(af,PctExp=afss/sum(afss)*100)[-3,5:6]

}

PerCentVarianceExplained <- do.call(rbind,VariableModelRes) %>%
  mutate(Ligand = rownames(swsw$Values))
PerCentVarianceExplained_ClusterAndAnimal <- do.call(rbind,VariableModelResWithCluster) %>% rownames_to_column(va
r = "Variable") %>% mutate(Variable=gsub("\\d$","",Variable)) %>%
  mutate(Ligand = rep(rownames(swsw$Values),each=2)) %>%
  filter(Ligand!="glucose")
PerCentVarianceExplained_ClusterAndAnimal %>%
  ggplot(aes(y=PctExp,x=Variable))+geom_violin()+geom_jitter(aes(colour=Ligand),size=3)
```



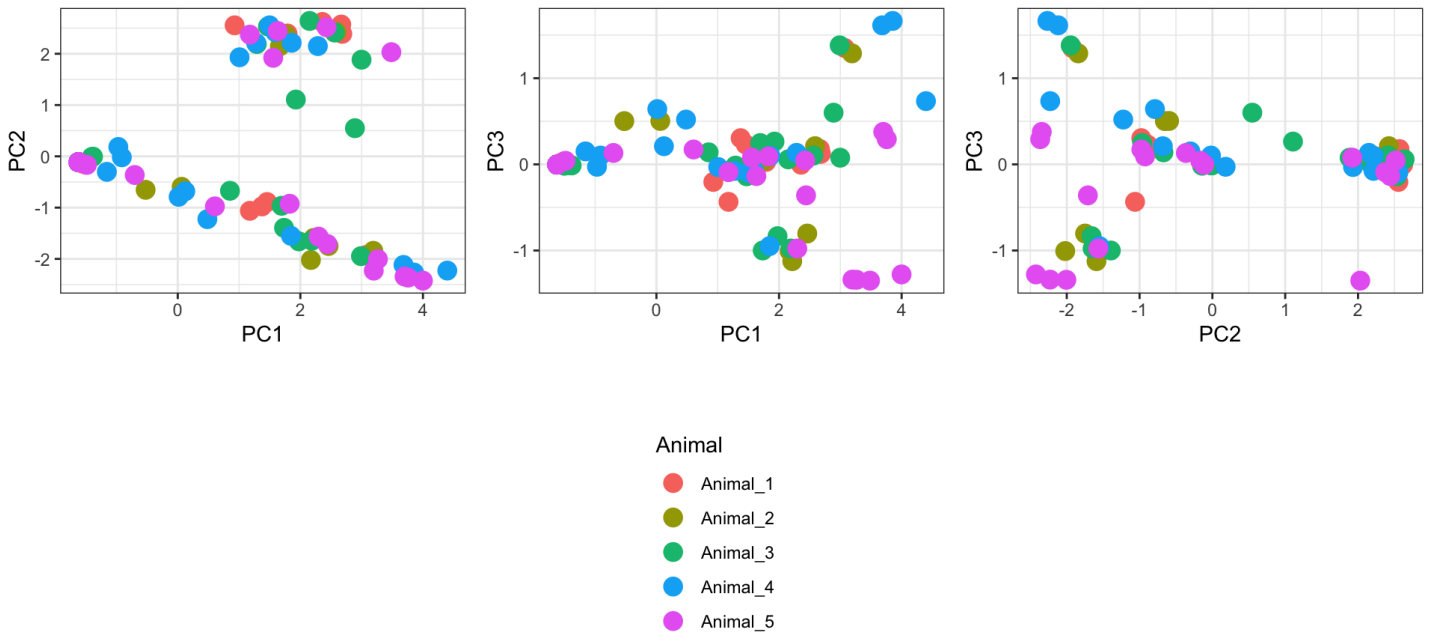
The total variance explained by Cluster membership for all responses (shown above) can be seen to be above 84% whereas the total explained by animal is less than 2% for all responses. This strongly highlights the small contribution to total variances by animal and the robust responses seen with clusters.

2.2 Principal component analysis

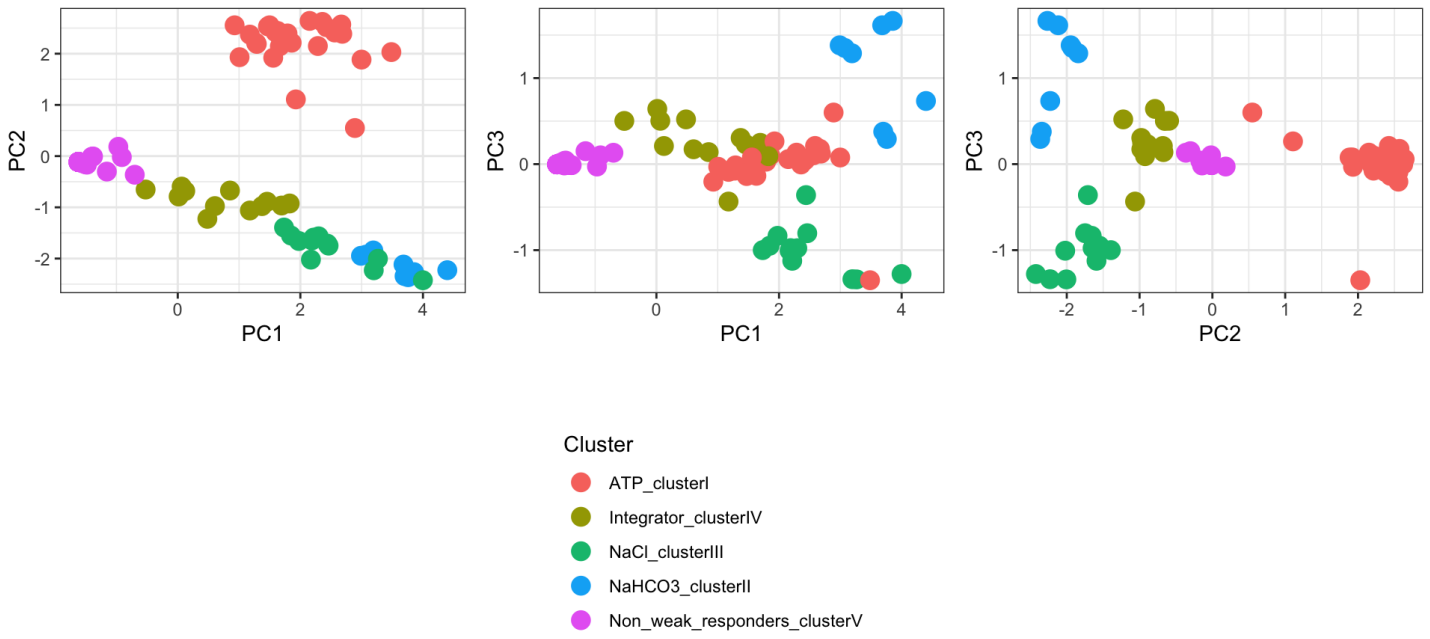
We may also review the major sources of variance by principal component analysis.

PCA analysis was applied to the neuron responses using base R and the FactoMineR package (Le S et al 2008) to visualise the contibution of animal or cluster to derived principal components.

```
#PCA
require(grid)
require(ggplot2)
p_animal <- plotAsPCA(swsw,GroupToPlot = "Animal")
```




```
p_cluster <- plotAsPCA(swsw,GroupToPlot = "Cluster")
```



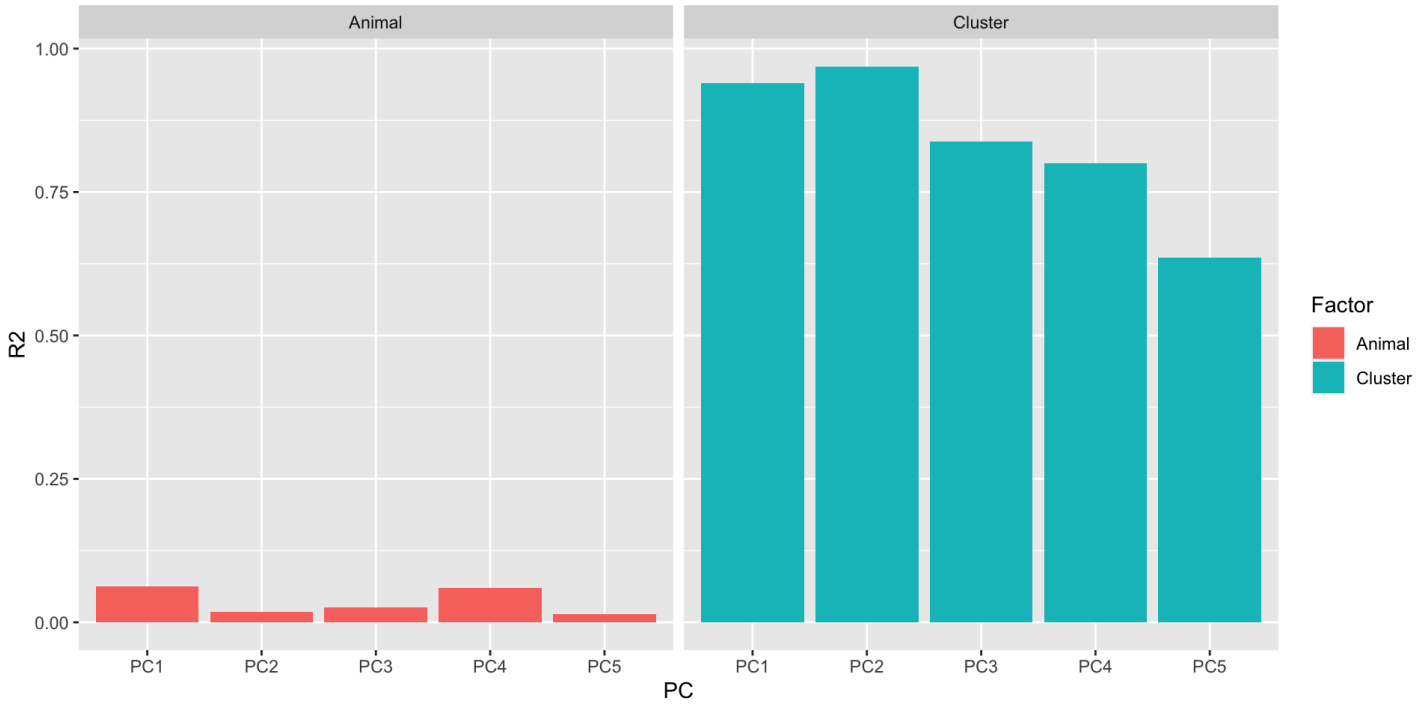

```
# grid.arrange(p_animal[[2]][[1]],p_animal[[2]][[2]],p_animal[[2]][[3]],ncol=1)
# grid.arrange(p_cluster[[2]][[1]],p_cluster[[2]][[2]],p_cluster[[2]][[3]])
```

From the comparison of PCA plots above we can see that the differing clusters are well distributed and defined by the major PCs in the data where as the animals are distributed throughout PCs.

This illustrates that the major sources of variation in the data is defined by clusters and not by variability between animals.

Following the visualisation of PCs, the correlation and significance of association of cluster membership and animal to the principal components were assessed in FactoMineR (Le S et al 2008).

```
require(factoextra)
require(FactoMineR)
require(tidyverse)
require(DT)
myPlot <- PCA(data.frame(t(swsw$Values),Cluster=unlist(swsw$CllusterInfo$CutMembers),Animal=factor(gsub("_Neur.*"
,"",unlist(rownames(swsw$CllusterInfo$CutMembers))))),quali.sup = c(8,9),graph = FALSE)
# print(plot.PCA(myPlot, choix = c("var"),axes = c(1,2), shadowtext = TRUE))
# print(plot.PCA(myPlot, choix = c("var"),axes = c(2,4), shadowtext = TRUE))
# print(plot.PCA(myPlot, choix = c("var"),axes = c(2,5), shadowtext = TRUE))

dimensionsPCA <- FactoMineR::dimdesc(myPlot,axes = c(1:5),proba = 1)
lapply(dimensionsPCA[grep("Dim",names(dimensionsPCA))],function(x)x$quali) %>%
  do.call(rbind,.) %>%  as.data.frame %>%
  rownames_to_column(var = "ResponseType") %>%
  mutate(PC=paste0("PC",rep(1:5,each=2))) %>%
  mutate(Factor=gsub("\\.*\\d","",.$ResponseType)) %>%
  dplyr::select(PC,Factor,R2) %>%
  ggplot(aes(x=PC,y=R2,fill=Factor))+geom_bar(stat = "identity",position="dodge")+facet_wrap(~Factor)
```



Cluster membership can be seen to highly significantly correlated with all principal components but animal showed low correlation and no significant association with any principal components. This further illustrates the major sources of variation can be associated with cluster membership and not with animal.

```
lapply(dimensionsPCA[grep("Dim",names(dimensionsPCA))],function(x)x$quali) %>%
  do.call(rbind,.) %>%  as.data.frame %>%
  rownames_to_column(var = "ResponseType") %>%
  mutate(PC=paste0("PC",rep(1:5,each=2))) %>%
  mutate(ResponseType=gsub("\\.*\\d","",.$ResponseType)) %>%
  dplyr::select(PC,ResponseType,p.value) %>%
  mutate(PadJ=p.adjust(p.value,method = "bonferroni")) %>%
  dplyr::select(-p.value) %>%
  spread(PC,PadJ) %>%
  datatable
```

Show 10 **entries** **Search:**

	ResponseType	PC1	PC2	PC3	PC4	PC5
1	Animal	0.775413046167282	1	1	0.871676960766336	1
2	Cluster	1.28351993370992e-76	5.4220369747065e-95	5.21315535603433e-49	4.4457735302356e-43	2.00929955133382e-26

Showing 1 to 2 of 2 entries

Previous 1 Next

3 References

Lawson RG, and Jurs PC. 1990. New Index for Clustering Tendency and Its Application to Chemical Problems. Journal of Chemical Information and Computer Sciences 30 (1): 36–41.

Adolfsson A, Ackerman M, Brownstein NC. 2018. To cluster, or not to cluster: An analysis of clusterability methods. Pattern Recognition (88),13-26

Hennig C 2020. fpc: Flexible Procedures for Clustering. R package version 2.2-5. https://CRAN.R-project.org/package=fpc (https://CRAN.R-project.org/package=fpc)

Hennig C. 2007. Cluster-wise assessment of cluster stability. Computational Statistics and Data Analysis, 52, 258-271.

Hennig C. 2008. Dissolution point and isolation robustness: robustness criteria for general cluster analysis methods. Journal of Multivariate Analysis 99, 1154-1176.

Charrad M, Ghazzali N, Boiteau V, Niknafs A. 2014. NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. Journal of Statistical Software 61: 1–36.

Rousseeuw PJ. 1987. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics. 20: 53–65.

Kassambara A and Mundt F 2020. factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.7. https://CRAN.R-project.org/package=factoextra (https://CRAN.R-project.org/package=factoextra)

Le S, Josse J, Husson F 2008. FactoMineR: An R Package for Multivariate Analysis. Journal of Statistical Software, 25(1), 1-18. 10.18637/jss.v025.i01

4 Session Information

```
sessionInfo()
```

```
## R version 3.6.3 Patched (2020-03-11 r78147)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Catalina 10.15.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] gridExtra_2.3       gplots_3.0.3        magrittr_1.5
##  [4] rio_0.5.16          fitdistrplus_1.0-14 npsurv_0.4-0.1
##  [7] lsei_1.2-0.1        survival_3.1-8      MASS_7.3-51.5
## [10] RColorBrewer_1.1-2  pheatmap_1.0.12     fpc_2.2-5
## [13] cluster_2.1.0       NbClust_3.0         DT_0.13
## [16] forcats_0.5.0       stringr_1.4.0       dplyr_0.8.5
## [19] purrr_0.3.3         readr_1.3.1         tidyr_1.0.2
## [22] tibble_3.0.0        tidyverse_1.3.0     FactoMineR_2.3
## [25] factoextra_1.0.7    ggplot2_3.3.0
##
## loaded via a namespace (and not attached):
##  [1] colorspace_1.4-1     ggsignif_0.6.0       ellipsis_0.3.0
##  [4] class_7.3-15         modeltools_0.2-23    mclust_5.4.6
##  [7] fs_1.4.0             rstudioapi_0.11      ggpubr_0.2.5
## [10] farver_2.0.3         ggrepel_0.8.2        flexmix_2.3-15
## [13] fansi_0.4.1          lubridate_1.7.4      xml2_1.3.2
## [16] splines_3.6.3        leaps_3.1            robustbase_0.93-6
## [19] knitr_1.28           jsonlite_1.6.1       broom_0.5.5
## [22] kernlab_0.9-29       dbplyr_1.4.2         compiler_3.6.3
## [25] httr_1.4.1           backports_1.1.5      assertthat_0.2.1
## [28] Matrix_1.2-18        cli_2.0.2            htmltools_0.4.0
## [31] tools_3.6.3          gtable_0.3.0         glue_1.3.2
## [34] Rcpp_1.0.4           cellranger_1.1.0     vctrs_0.2.4
## [37] gdata_2.18.0         nlme_3.1-144         crosstalk_1.1.0.1
## [40] xfun_0.12            openxlsx_4.1.4       rvest_0.3.5
## [43] lifecycle_0.2.0      gtools_3.8.2         DEoptimR_1.0-8
## [46] scales_1.1.0         hms_0.5.3            parallel_3.6.3
## [49] yaml_2.2.1           curl_4.3             stringi_1.4.6
## [52] caTools_1.18.0       zip_2.0.4            rlang_0.4.5
## [55] pkgconfig_2.0.3      prabclus_2.3-2       bitops_1.0-6
## [58] evaluate_0.14        lattice_0.20-38      htmlwidgets_1.5.1
## [61] labeling_0.3         cowplot_1.0.0        tidyselect_1.0.0
## [64] R6_2.4.1             generics_0.0.2       DBI_1.1.0
## [67] pillar_1.4.3         haven_2.2.0          foreign_0.8-75
## [70] withr_2.1.2          scatterplot3d_0.3-41 nnet_7.3-12
## [73] modelr_0.1.6         crayon_1.3.4         KernSmooth_2.23-16
## [76] rmarkdown_2.1        readxl_1.3.1         data.table_1.12.8
## [79] reprex_0.3.0         digest_0.6.25        diptest_0.75-7
## [82] flashClust_1.01-2    stats4_3.6.3         munsell_0.5.0
```