

MetPlast Tutorial

Lucio D'Andrea

8/3/2021

Introduction

Plants as sessile organisms are unable to evade unfavorable growing conditions by simply moving away. Hence, they evolved a unique phenotypic plasticity that allows them to better adapt or survive to challenging environments. Metabolic plasticity is the ability of plants to biosynthesize a myriad of specialized compounds that allows them to cope with changes in their immediate surroundings. Thus, specialized metabolites are involved in a wide variety of ecological processes such as herbivorous attack, interaction with neighboring plants, as well as dealing with changes in light, or temperature conditions.

The evolution of plant metabolic plasticity has been mainly driven by gene duplication, (or even whole genome duplication) followed by neo-functionalization. Gene duplication has been proven to shape the evolution of several specialized metabolic pathways. However, the effect of WGD on the metabolic plasticity remains to be elucidated. Possibly, the duplication of the whole genome allowed plants to screen a wider phenotypic space under stress conditions, promoting innovation, rapid adaptation and ultimately, speciation.

Artificial selection processes have also influenced plant metabolic repertoire. Domestication, i.e., the process of selecting plants to increase their suitability to human requirements, as well as crop improvement has caused genetic bottlenecks and massive reduction of the allelic diversity. Thus, artificial selection has introduced quantitative changes in various nutrition compounds. For instance, studies on tomato domestication have shown a major reduction in the levels of the anti-nutritional steroidal glycoalkaloids in ripe fruits. Although, both natural and artificial selection have been pointed as major forces shaping the biosynthesis and accumulation of several specialized metabolites, the evaluation of their effects on the metabolome and metabolic plasticity are in their infancy.

Information theory provides a statistical framework that allows to quantify and evaluate metabolic plasticity. Metabolome diversity and specialization can be calculated based on the Shannon entropy of the metabolic frequency distribution. Shannon entropy is a useful parameter, that measures the information held in a set of data. Thus, its calculation can be used to estimate different parameters associated with a given metabolome: (1) H_j index, metabolome diversity; (2) Δ_j index, metabolic profile specialization; (3) S_i index, metabolic specificity of individual metabolites. The individual calculation of these parameters was successfully applied on LC-MS/MS data to understand the dynamics of different plant species' metabolomes.

Here, we present MetPlast, an R-package that integrates the calculation, and visualization of Shannon-based metabolic plasticity parameters. We evaluate the effect of crop domestication by comparing the proposed parameters between the domesticated *Solanum lycopersicum*, the semi-domesticated *S. lycopersicum* var. *cerisiforme* and the wild relative *S. pimpinellifolium*.

Upload MetPlast packages

Install and load the MetPlast package from GitHub:

```
# install.github ("danlucio86/MetPlast")

library("MetPlast")
```

Upload external packages

This package has several external dependencies. Hence, the user needs to install and upload the following packages:

1. dlookr
2. ggplot2
3. ggfortify
4. tidyverse
5. data.table
6. grid.Extra

```
# install.packages()
```

Then, upload them:

```
library("dlookr")
```

```
## Imported Arial Narrow fonts.
```

```
##  
## Attaching package: 'dlookr'
```

```
## The following object is masked from 'package:base':  
##  
##      transform
```

```
library("ggplot2")  
  
library("ggfortify")  
  
library("tidyverse")
```

```
## — Attaching packages ————— tidyverse 1.3.1 —
```

```
## ✓ tibble 3.1.4      ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.3      ✓ stringr 1.4.0  
## ✓ readr 2.0.1      ✓ forcats 0.5.1  
## ✓ purrr 0.3.4
```

```
## — Conflicts ————— tidyverse_conflicts() —  
## x tidyr::extract() masks dlookr::extract()  
## x dplyr::filter()  masks stats::filter()  
## x dplyr::lag()     masks stats::lag()
```

```
library("data.table")
```

```
##  
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':  
##  
## between, first, last
```

```
## The following object is masked from 'package:purrr':  
##  
## transpose
```

```
library("gridExtra")
```

```
##  
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':  
##  
## combine
```

```
library("FactoMineR")
```

```
library("factoextra")
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

Data

Uploading

This package takes as a data set a data frame containing the standardized quantity of different metabolites - measured as peak's intensity- in a given set of samples. It needs to be tidy in order to have the first column the "Compounds", and then all the samples. It is important to named the first column as "Compounds".

Upload in R the testing data set "Test.csv" obtained from Zhug et al 2018. This data set contains the metabolic profile of red-fruited tomato populations including three different species: *S. lycopersicum*, *S. lycopersicum* var. *cersiforme*, and *S. pimpinellifolium*. Zhug et al 2018 quantified fruit metabolites using a broadly targeted liquid chromatography-tandem mass spectrometry (LC-MS/MS)- based metabolic profiling method. The data set contains 301 different accessions (2 biological replicates each), expanding across different geographical distribution, passport information, between others. Thus, although through out the tutorial the samples are visualize based on the species identity, it is absolutely essential to keep in mind the different nature of the accessions.

```
Data <- read.csv2(file = "Test.csv", header = TRUE, row.names = "Compounds")  
  
library(rmarkdown)  
paged_table(head(Data))
```

S..lycopersicum
<dbl>

S..lycopersicum.1
<dbl>

S..lycopersicum.2
<dbl>

S..lycopersicum.3
<dbl>

	S..lycopersicum <dbl>	S..lycopersicum.1 <dbl>	S..lycopersicum.2 <dbl>	S..lycopersicum.3 <dbl>
SIFM0001	9561.45	290809.910	7753.55	NA
SIFM0002	7465.25	7567.723	15637.75	11051.8613
SIFM0003	7248.95	64658.588	1070.10	5109.6103
SIFM0004	7178.50	1528529.230	714.00	170.6112
SIFM0006	209964.25	165790.076	173637.20	46541.6085
SIFM0007	272210.25	170302.838	485893.45	295343.7568

6 rows | 1-6 of 403 columns

Replacing NAs values

When comparing the set of metabolites from different samples, it is pretty usual that some metabolites are sample-specific. Hence, it is expected to have missing values (NAs). In order to deal with this, The NAs values are replaced by the minimum value found in the data set divided by 1e6.

```
Data[is.na(Data)] <- (min(Data, na.rm=TRUE))/1000000
```

```
library(rmarkdown)
paged_table(head(Data))
```

	S..lycopersicum <dbl>	S..lycopersicum.1 <dbl>	S..lycopersicum.2 <dbl>	S..lycopersicum.3 <dbl>
SIFM0001	9561.45	290809.910	7753.55	3.000000e-10
SIFM0002	7465.25	7567.723	15637.75	1.105186e+04
SIFM0003	7248.95	64658.588	1070.10	5.109610e+03
SIFM0004	7178.50	1528529.230	714.00	1.706112e+02
SIFM0006	209964.25	165790.076	173637.20	4.654161e+04
SIFM0007	272210.25	170302.838	485893.45	2.953438e+05

6 rows | 1-6 of 403 columns

Initial statistical analysis

Testing normal distribution

Before proceeding with the analysis of the metabolic data, we recommend to test if each sample is normally distributed.

For example, here we performs Shapiro-Wilk to test whether the samples are normally distributed.

Considering an alpha value of 1 %, then if the Shapiro value is lower than 0.01 the null hypothesis is not reject and we can assume normal distribution.

```
normality <- normality(Data)
```

#Based on these results we can conclude that all our samples have normal distribution (value >0.01). Moreover, most of them show a value greater than 0.05 providing higher robustness to the analysis.

Metabolic Parameters

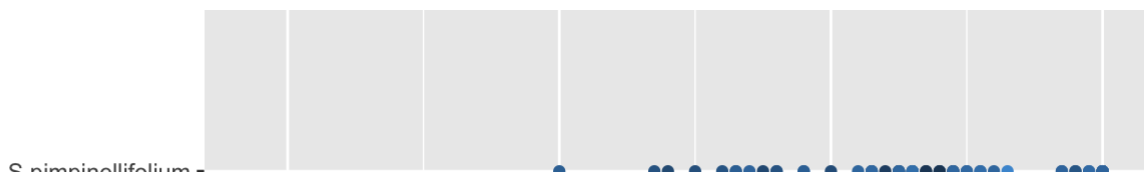
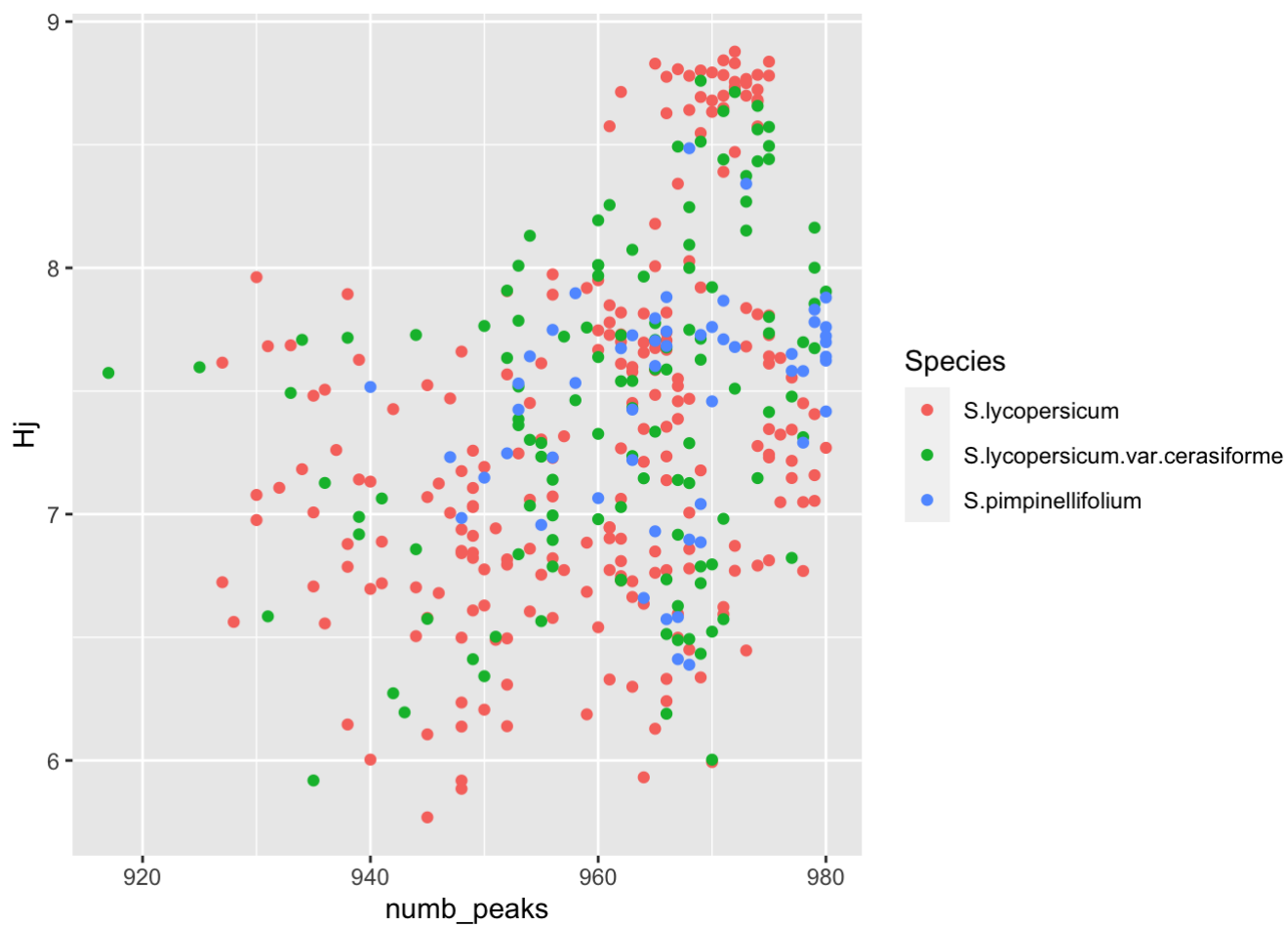
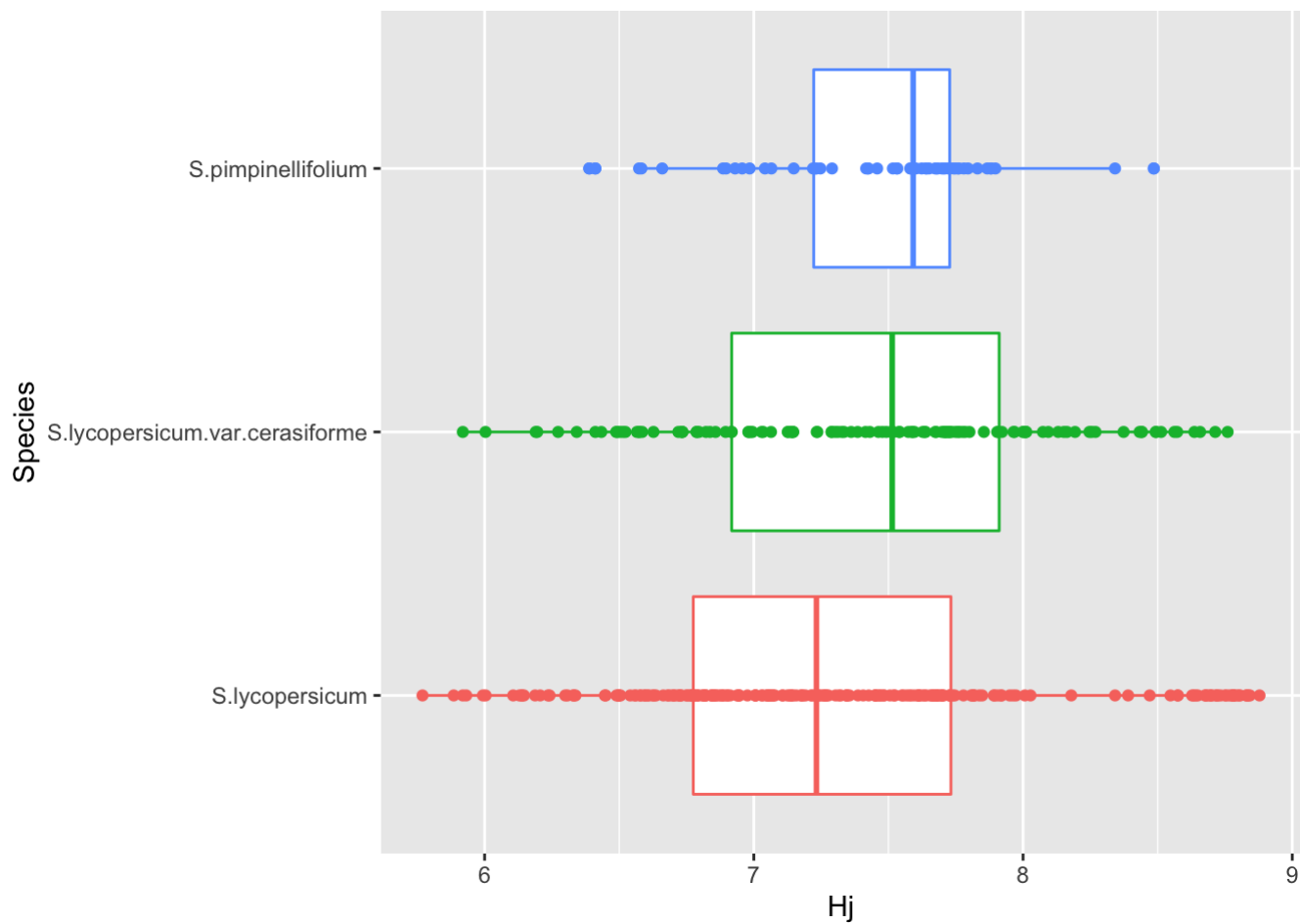
Metabolic Plasticity (MetDiv function)

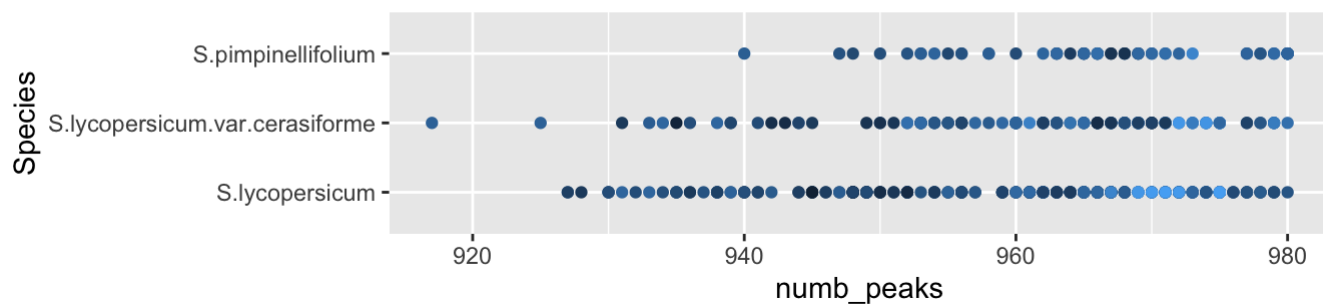
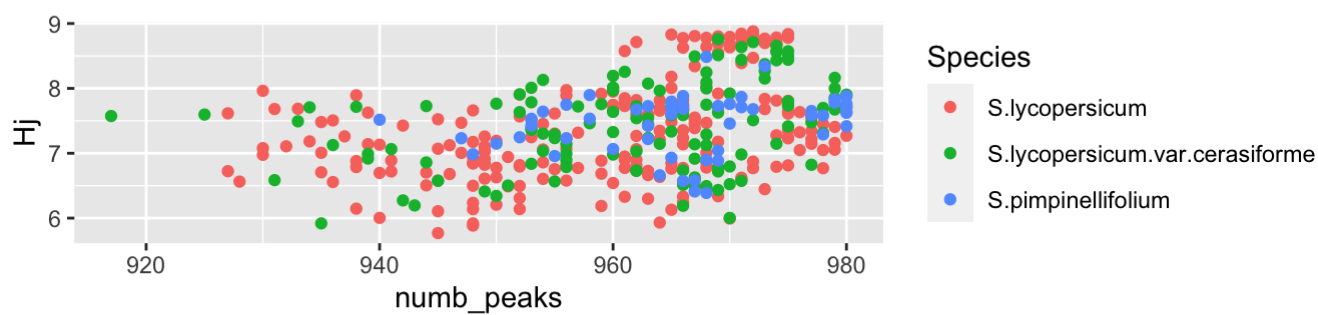
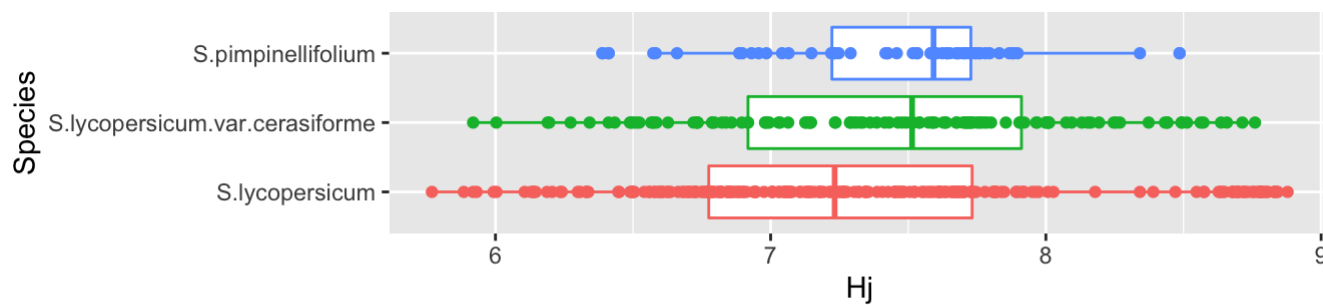
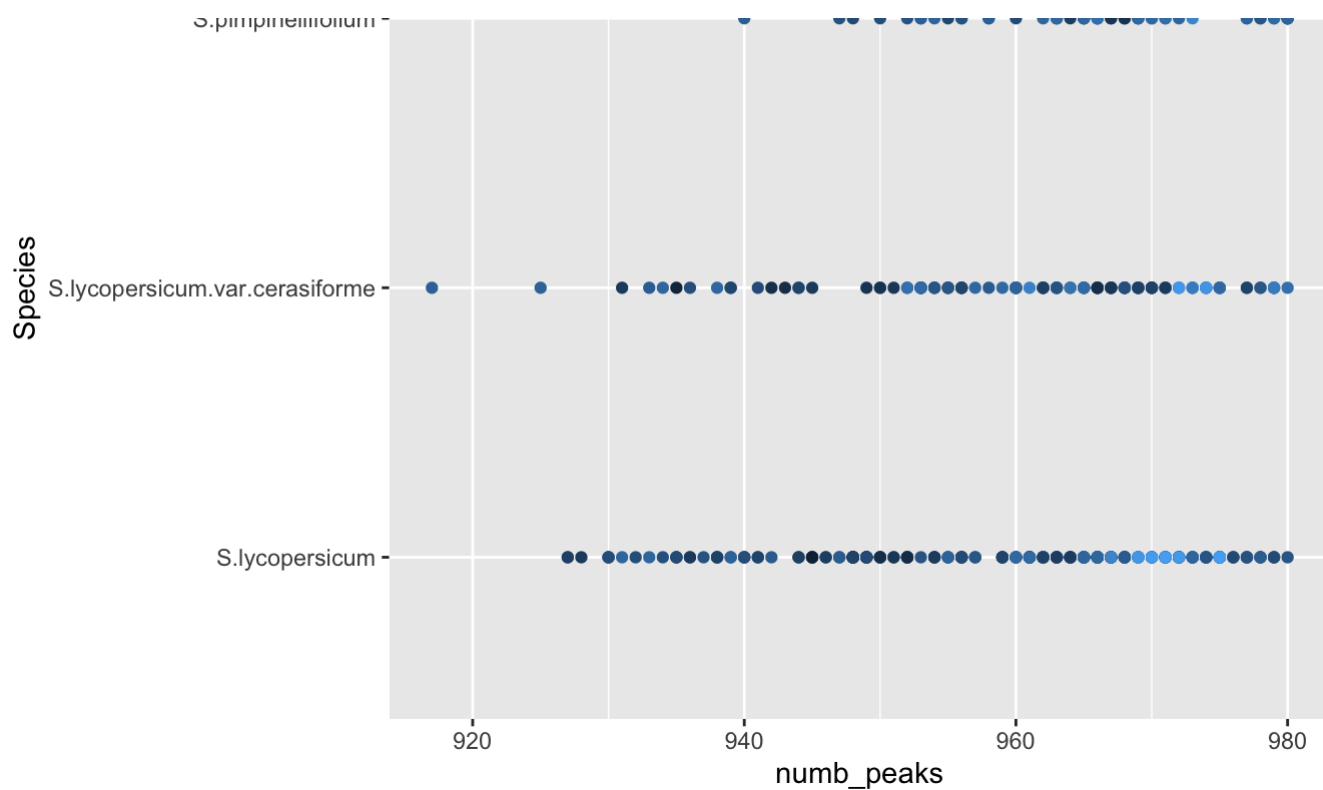
MetDiv() calculates METabolic Diversity (Hj) index based on Shannon entropy. The metabolic profile diversity is defined as the Shannon entropy using MS/MS metabolite frequency distribution in a sample (j). Hj can take any value between zero when only one metabolite is detected up to $\log_2(m)$, where all m metabolites are detected and accumulates at the same frequency: $1/m$.

This function returns a list of 5 objects that are described below.

```
Hj <- MetDiv (Data)
```

```
## [1] "To use MetPar function store this calculation as Hj <- MetDiv (Data)"
```





```
## [[1]]
## TableGrob (3 x 1) "arrange": 3 grobs
##   z      cells      name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (2-2,1-1) arrange gtable[layout]
## 3 3 (3-3,1-1) arrange gtable[layout]
##
## [[2]]
##           Species      Hj numb_peaks
## 1:   S.lycopersicum 7.660288      948
## 2:   S.lycopersicum 7.634306      976
## 3:   S.lycopersicum 7.597500      963
## 4:   S.lycopersicum 7.505589      936
## 5:   S.lycopersicum 8.006721      965
## ---
## 398: S.pimpinellifolium 7.879622      980
## 399: S.pimpinellifolium 7.866917      971
## 400: S.pimpinellifolium 7.639788      980
## 401: S.pimpinellifolium 7.881527      966
## 402: S.pimpinellifolium 7.780209      979
```

It returns a list with 5 objects:

1. View(Hj[[2]]) display a data frame with the Hj value (col2) and number of peaks (compounds) (col3) per species (col1).

```
print(Hj[[2]])
```

```
##           Species      Hj numb_peaks
## 1:   S.lycopersicum 7.660288      948
## 2:   S.lycopersicum 7.634306      976
## 3:   S.lycopersicum 7.597500      963
## 4:   S.lycopersicum 7.505589      936
## 5:   S.lycopersicum 8.006721      965
## ---
## 398: S.pimpinellifolium 7.879622      980
## 399: S.pimpinellifolium 7.866917      971
## 400: S.pimpinellifolium 7.639788      980
## 401: S.pimpinellifolium 7.881527      966
## 402: S.pimpinellifolium 7.780209      979
```

In our example the Hj factor can range between 0 - when only one metabolite is detected- and $\log_2(980) = 9.93$ - when all the detected metabolites accumulates at the same frequency.

This data frame is then used to generate a series of plots showing the behavior and inter-dependency of the calculated variables: Species, Hj, and number of compounds.

2. A boxplot depicting the variation of Hj per species;

The box plot shows the calculated metabolic diversity (Hj) for each metabolome grouped by species.

```
print(Hj[[2]] %>% filter(Species=="S.lycopersicum"))
```



```
##           Species      Hj numb_peaks
##  1: S.lycopersicum 7.660288      948
##  2: S.lycopersicum 7.634306      976
##  3: S.lycopersicum 7.597500      963
##  4: S.lycopersicum 7.505589      936
##  5: S.lycopersicum 8.006721      965
## ---
## 224: S.lycopersicum 7.682045      931
## 225: S.lycopersicum 7.674575      965
## 226: S.lycopersicum 6.809461      962
## 227: S.lycopersicum 7.470007      947
## 228: S.lycopersicum 6.663730      963
```

```
print(Hj[[2]] %>% filter(Species=="S.lycopersicum.var.cerasiforme"))
```

```
##           Species      Hj numb_peaks
##  1: S.lycopersicum.var.cerasiforme 8.246400      968
##  2: S.lycopersicum.var.cerasiforme 7.573731      917
##  3: S.lycopersicum.var.cerasiforme 8.562138      974
##  4: S.lycopersicum.var.cerasiforme 7.509756      972
##  5: S.lycopersicum.var.cerasiforme 7.540030      962
## ---
## 116: S.lycopersicum.var.cerasiforme 6.189766      966
## 117: S.lycopersicum.var.cerasiforme 8.713749      972
## 118: S.lycopersicum.var.cerasiforme 6.574938      945
## 119: S.lycopersicum.var.cerasiforme 7.757948      959
## 120: S.lycopersicum.var.cerasiforme 7.492333      933
```

```
print(Hj[[2]] %>% filter(Species=="S.pimpinellifolium"))
```

##	Species	Hj	numb_peaks
## 1:	S.pimpinellifolium	7.710207	971
## 2:	S.pimpinellifolium	7.830755	979
## 3:	S.pimpinellifolium	7.678856	972
## 4:	S.pimpinellifolium	7.231682	947
## 5:	S.pimpinellifolium	8.341614	973
## 6:	S.pimpinellifolium	6.956529	955
## 7:	S.pimpinellifolium	7.581924	977
## 8:	S.pimpinellifolium	6.984498	948
## 9:	S.pimpinellifolium	7.581232	978
## 10:	S.pimpinellifolium	7.247196	952
## 11:	S.pimpinellifolium	7.290521	978
## 12:	S.pimpinellifolium	7.148240	950
## 13:	S.pimpinellifolium	7.601886	965
## 14:	S.pimpinellifolium	7.424486	953
## 15:	S.pimpinellifolium	8.485690	968
## 16:	S.pimpinellifolium	7.065124	960
## 17:	S.pimpinellifolium	7.795545	965
## 18:	S.pimpinellifolium	7.516646	940
## 19:	S.pimpinellifolium	7.219950	963
## 20:	S.pimpinellifolium	6.573533	966
## 21:	S.pimpinellifolium	7.641003	954
## 22:	S.pimpinellifolium	7.041364	969
## 23:	S.pimpinellifolium	7.748477	956
## 24:	S.pimpinellifolium	6.582335	967
## 25:	S.pimpinellifolium	7.230464	956
## 26:	S.pimpinellifolium	6.896792	968
## 27:	S.pimpinellifolium	7.897217	958
## 28:	S.pimpinellifolium	6.411262	967
## 29:	S.pimpinellifolium	7.424794	963
## 30:	S.pimpinellifolium	6.659703	964
## 31:	S.pimpinellifolium	7.530833	953
## 32:	S.pimpinellifolium	6.930415	965
## 33:	S.pimpinellifolium	7.533007	958
## 34:	S.pimpinellifolium	6.388631	968
## 35:	S.pimpinellifolium	7.683146	966
## 36:	S.pimpinellifolium	6.885783	969
## 37:	S.pimpinellifolium	7.704399	965
## 38:	S.pimpinellifolium	7.417259	980
## 39:	S.pimpinellifolium	7.742226	966
## 40:	S.pimpinellifolium	7.698313	980
## 41:	S.pimpinellifolium	7.673863	962
## 42:	S.pimpinellifolium	7.650729	977
## 43:	S.pimpinellifolium	7.726455	963
## 44:	S.pimpinellifolium	7.723264	980
## 45:	S.pimpinellifolium	7.458244	970
## 46:	S.pimpinellifolium	7.623584	980
## 47:	S.pimpinellifolium	7.760158	970
## 48:	S.pimpinellifolium	7.759824	980
## 49:	S.pimpinellifolium	7.727816	969
## 50:	S.pimpinellifolium	7.879622	980
## 51:	S.pimpinellifolium	7.866917	971
## 52:	S.pimpinellifolium	7.639788	980
## 53:	S.pimpinellifolium	7.881527	966
## 54:	S.pimpinellifolium	7.780209	979
##	Species	Hj	numb_peaks

It can be observed that the domesticated *S. lycopersicum* H_j indexes display a higher variance, ranging from 5.76 to 8.87, compared with *S. lycopersicum* var. *ceresiformis* (5.91 - 8.75) and the wild relative *S. pimpinellifolium* (6.38 - 8.48). These differences might be due to technical and biological reasons, such as sampling, accessions and the domestication process. Additionally, a few outliers can be observed.

The median values show that there is a higher H_j median value when the degree of domestication is lower. Possibly indicating that in average and considering this data set, it is more likely to get a more diverse metabolome as the degree of domestication is lower.

3. A point plot depicting the dependency between the number of peaks and H_j;

The H_j factor depends mostly on two different parameters: (a) The number of peaks, (b) the frequency of each peak in the whole data set. This plot shows to what extent the H_j increases based on the number of peak in the species under evaluation. In general, it can be expected to observe a curve where the H_j values reaches a plateau.

4. A point plot depicting the dependency between the number of peaks and Species;

As mentioned before, the H_j factor depends mostly on two different parameters including the number of peaks. In this plot we can observe that the variation on the compounds detected in the different species have a similar behavior as the H_j factor. Hence, we can infer that the higher level of metabolic diversity in the less domesticated species, could be related with a loss in the capacity of synthesizing certain compounds.

5. A grid with all the plots (point 2, 3 and 4).

Summary: These differences might be related with the geographical origins, consumption type or improvement status in the *S. lycopersicum* and *S. lycopersicum* var. *ceresiformis*. To evaluate this possibility we can include categorical data. As an example please, upload the file `categories.csv`, and store it as a vector called "categories":

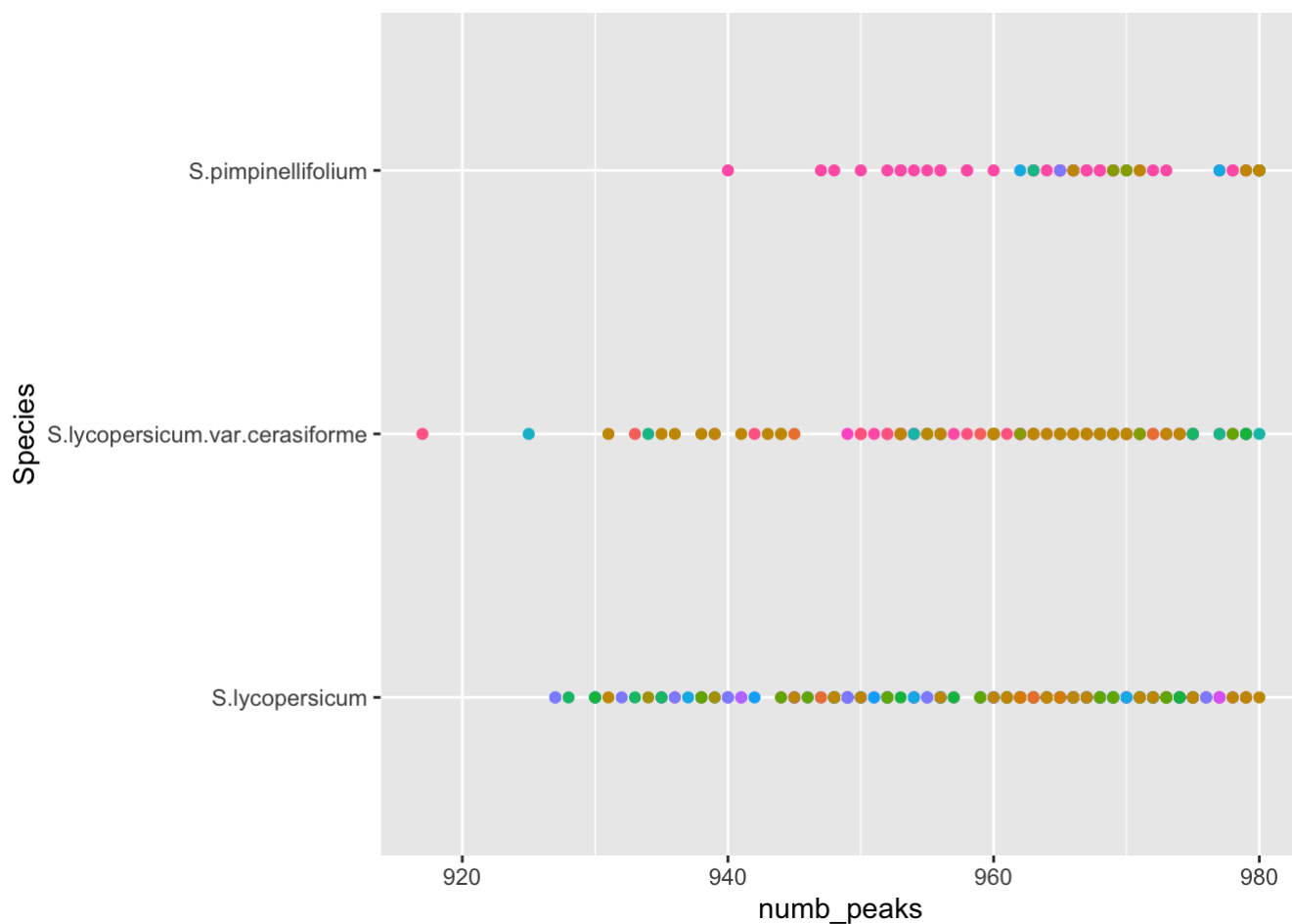
```
categories <- read.csv2(file="categories.csv")
```

Then, it can be added to the data frame generated by `MetDiv()`

```
Data_categ <- cbind (Hj[[2]], categories)
```

Finally, the same plots generated by `MetDiv()` can be generated using the package `ggplot2`. As an example:

```
ggplot(Data_categ, aes(num_peak, Species, color = Categories)) + geom_point() + theme(legend.position = "none")
```



This analysis shows that the categorical data used explain at some extend the high variance observe within some species. The very same strategy can be applied in the following functions.

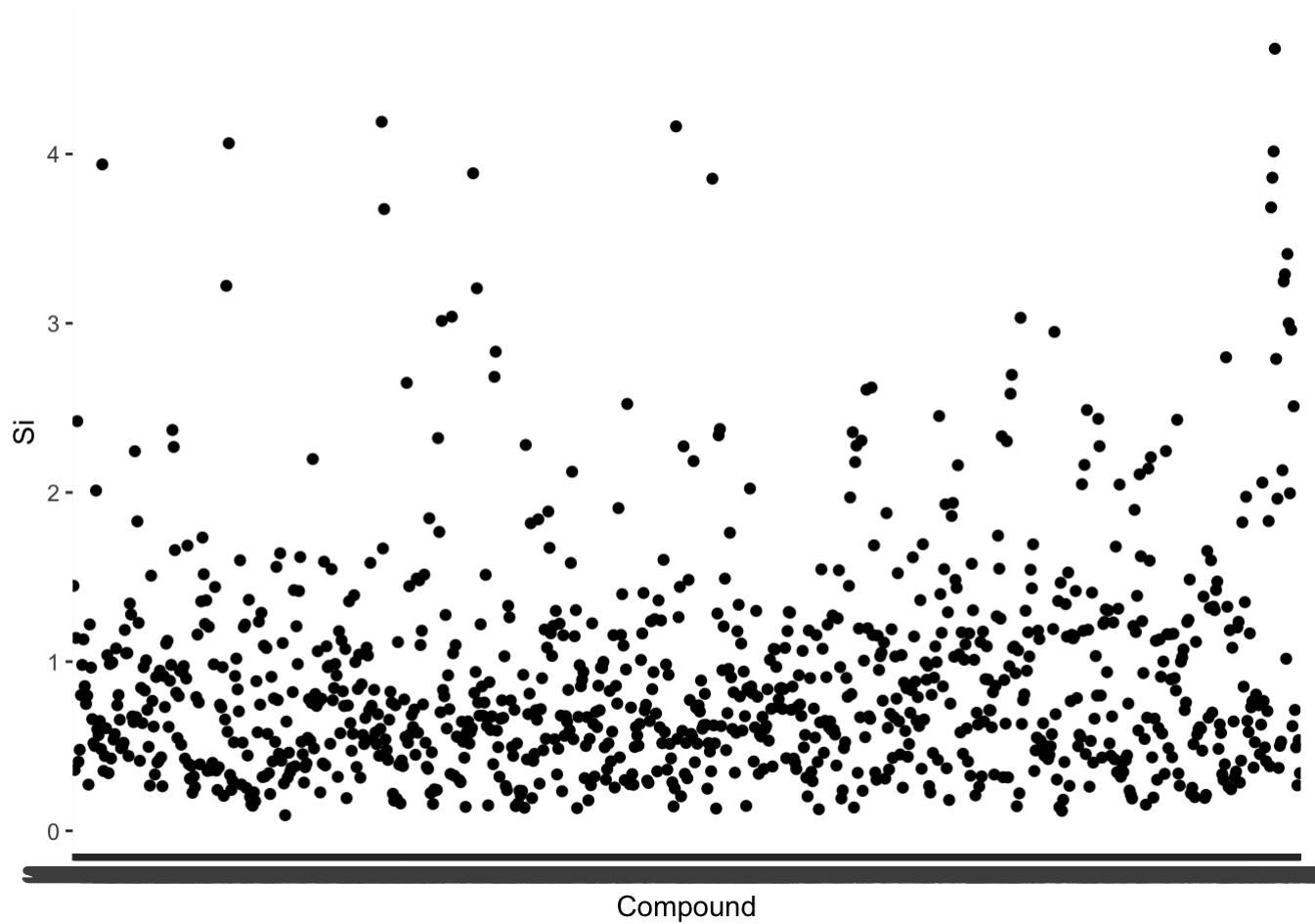
Metabolite and Metabolome Specialization Index (MetSpec function)

MetSpec calculates METabolite SPECialization (S_i) and METabolome SPECialization (δ_j) indexes. Metabolic specificity (S_i) is defined as the specificity of a particular MS/MS metabolite (i) among a set of samples (j). Metabolome specialization δ_j is measured for each j th sample, as the average of the MS/MS specificity.

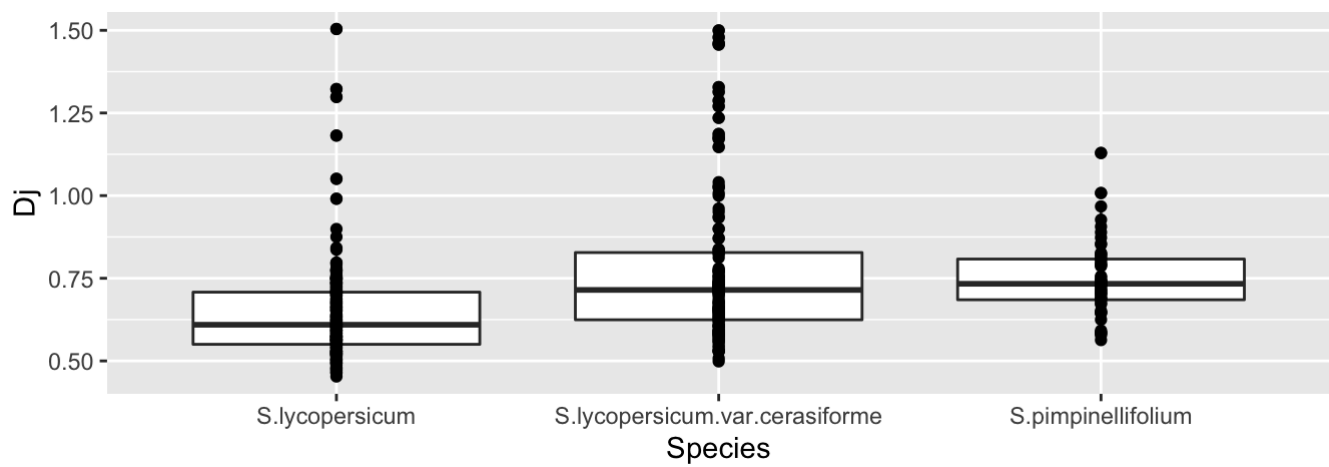
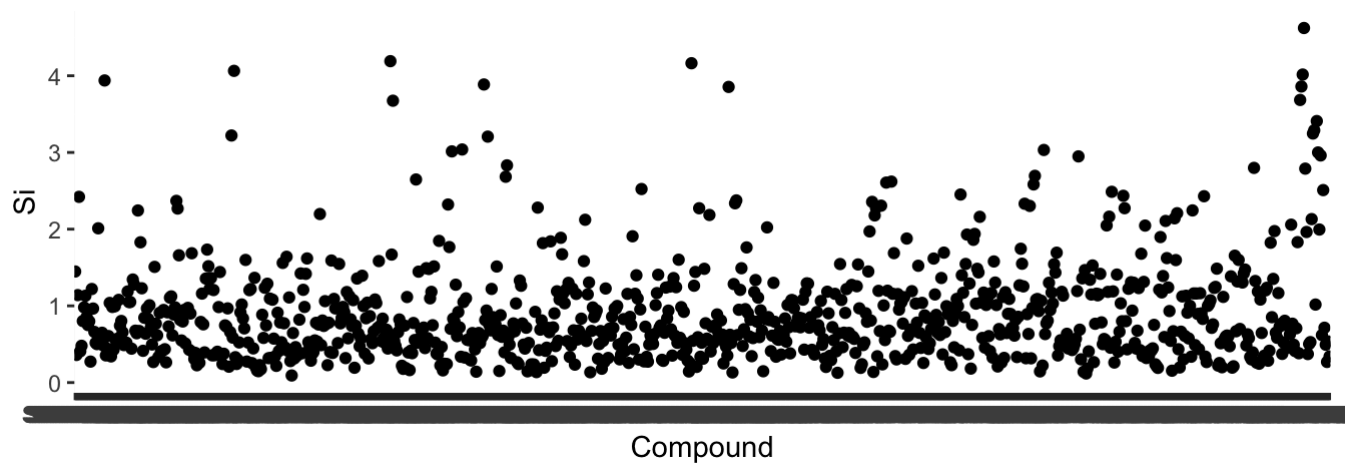
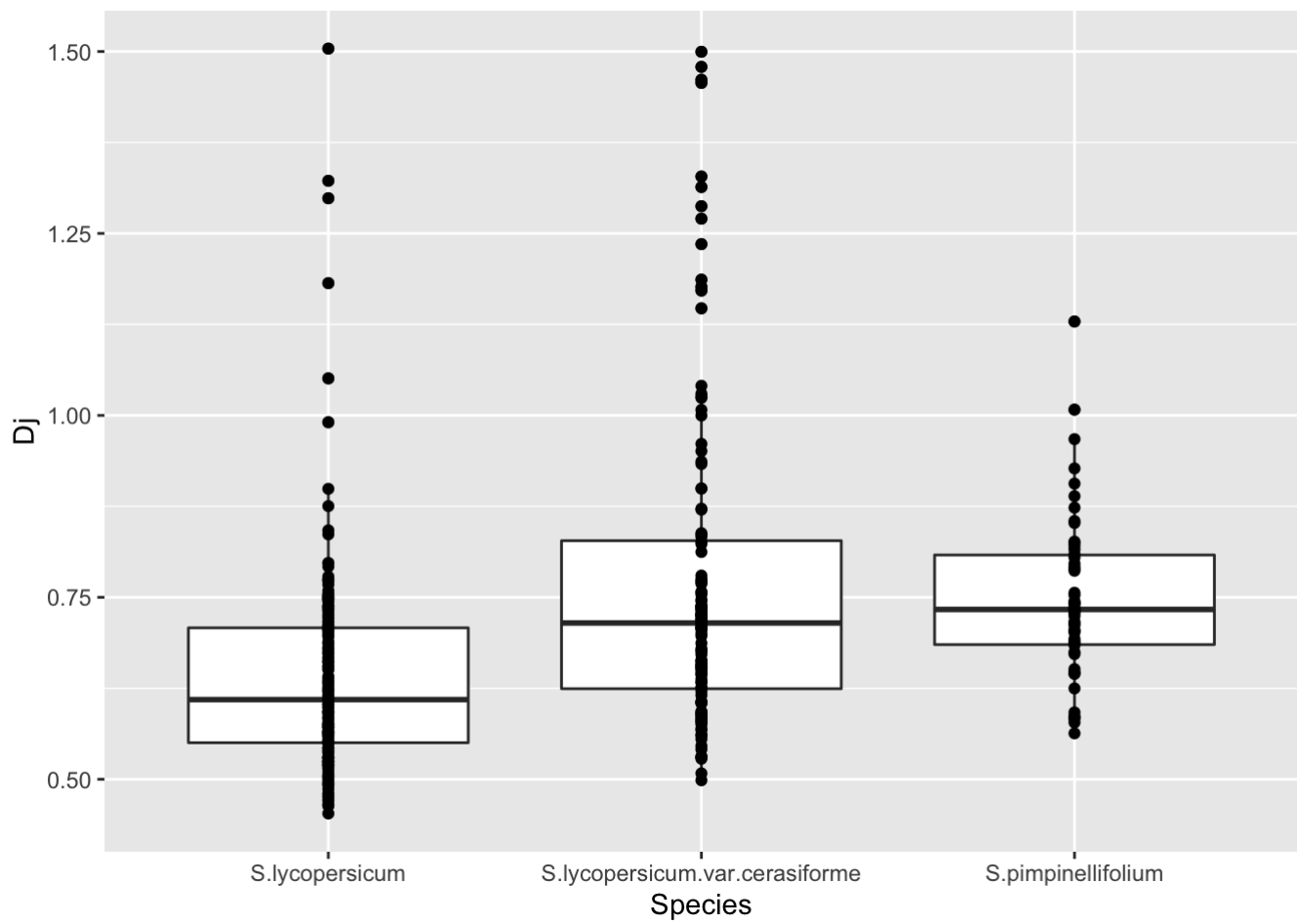
It returns a list with 4 objects that are described below.

```
Dj <- MetSpec (Data)
```

```
## [1] "THIS PARAMETER CAN NOT BE EXTRAPOLATED TO OTHER DATA SETS OR SUBSETS"
## [1] "To use MetPar function store this calculation as Dj <- MetSpec (Data)"
```



```
##           Species      Dj
##  1:    S.lycopersicum 0.5904608
##  2:    S.lycopersicum 0.6552455
##  3:    S.lycopersicum 0.6507864
##  4:    S.lycopersicum 0.5651353
##  5:    S.lycopersicum 0.7064863
##  ---
## 398: S.pimpinellifolium 0.7331459
## 399: S.pimpinellifolium 0.6747583
## 400: S.pimpinellifolium 0.6893213
## 401: S.pimpinellifolium 0.8093060
## 402: S.pimpinellifolium 0.7529746
```



```
## [[1]]
## TableGrob (2 x 1) "arrange": 2 grobs
##   z      cells      name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (2-2,1-1) arrange gtable[layout]
##
## [[2]]
##      Compound      Si
## 1: S1FM0001 1.4480607
## 2: S1FM0002 0.3598622
## 3: S1FM0003 1.1401669
## 4: S1FM0004 2.4211992
## 5: S1FM0006 0.4082558
## ---
## 976: S1FM1996 0.7147087
## 977: S1FM1997 0.4936772
## 978: S1FM1998 0.2691070
## 979: S1FM1999 0.5358869
## 980: S1FM2000 0.3410207
##
## [[3]]
##      Species      Dj
## 1: S.lycopersicum 0.5904608
## 2: S.lycopersicum 0.6552455
## 3: S.lycopersicum 0.6507864
## 4: S.lycopersicum 0.5651353
## 5: S.lycopersicum 0.7064863
## ---
## 398: S.pimpinellifolium 0.7331459
## 399: S.pimpinellifolium 0.6747583
## 400: S.pimpinellifolium 0.6893213
## 401: S.pimpinellifolium 0.8093060
## 402: S.pimpinellifolium 0.7529746
```

It returns a list with 4 objects:

1. A data frame with the Si value per compound;

View(Dj[[2]]) allows the user to obtain a data frame with each compound Si value. This values indicate how specific a metabolite is in a given data set. Si will be zero if the metabolite accumulates at the same frequency in all samples and will be maximum with $\log_2(m)$, i.e. if the metabolite exclusively accumulated in a single sample. In our example, theoretically range between 0 and $\log_2(980)=9.93$

A closer look to the values, shows that Si ranges from 0.092 (m = SIFM0252) to 4,62 (m = SIFM1980), indicating that the latter was detected in a fewer amount of samples compare with SIFM0252.

2. A data frame with the δ_j per Species;

View(Dj[[3]]) allows the user to obtain a data frame with each compound metabolome specialization value (δ_j). This values indicate how specialize a metabolome is, given a data set. δ_j varies from 0 if all metabolites that accumulates in the sample are completely unspecific ($S_i = 0$ for all) up to a maximum of $\log_2(m)$, when all metabolites accumulating in a sample are not synthesized anywhere else.

3. A point plot depicting the Si value of each compound;

This point plot is a visual representation of the data frame Dj[[2]], where each metabolite specificity is quantified (Si). It can be observe that the vast majority of the compounds has a Si value between 0 and 1, indicating a similar accumulation of the different metabolites across the data set.

4. A point plot depicting the δ_j per species

This boxplot is a visual representation of the dataframe $D_j[[3]]$, where the metabolome specialization (D_j) of each species are measured. In our example, it there is a higher variation in the domesticated landraces compared with the wild relative *S. pimpinellifolium*. Thus, we might need to consider some categorical data, as explained before, to confidently assess differences in the specialization levels.

Metabolite Specialization Analysis (MetliteSpec function)

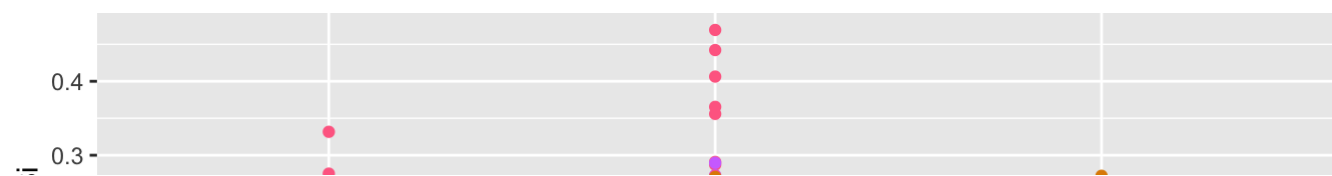
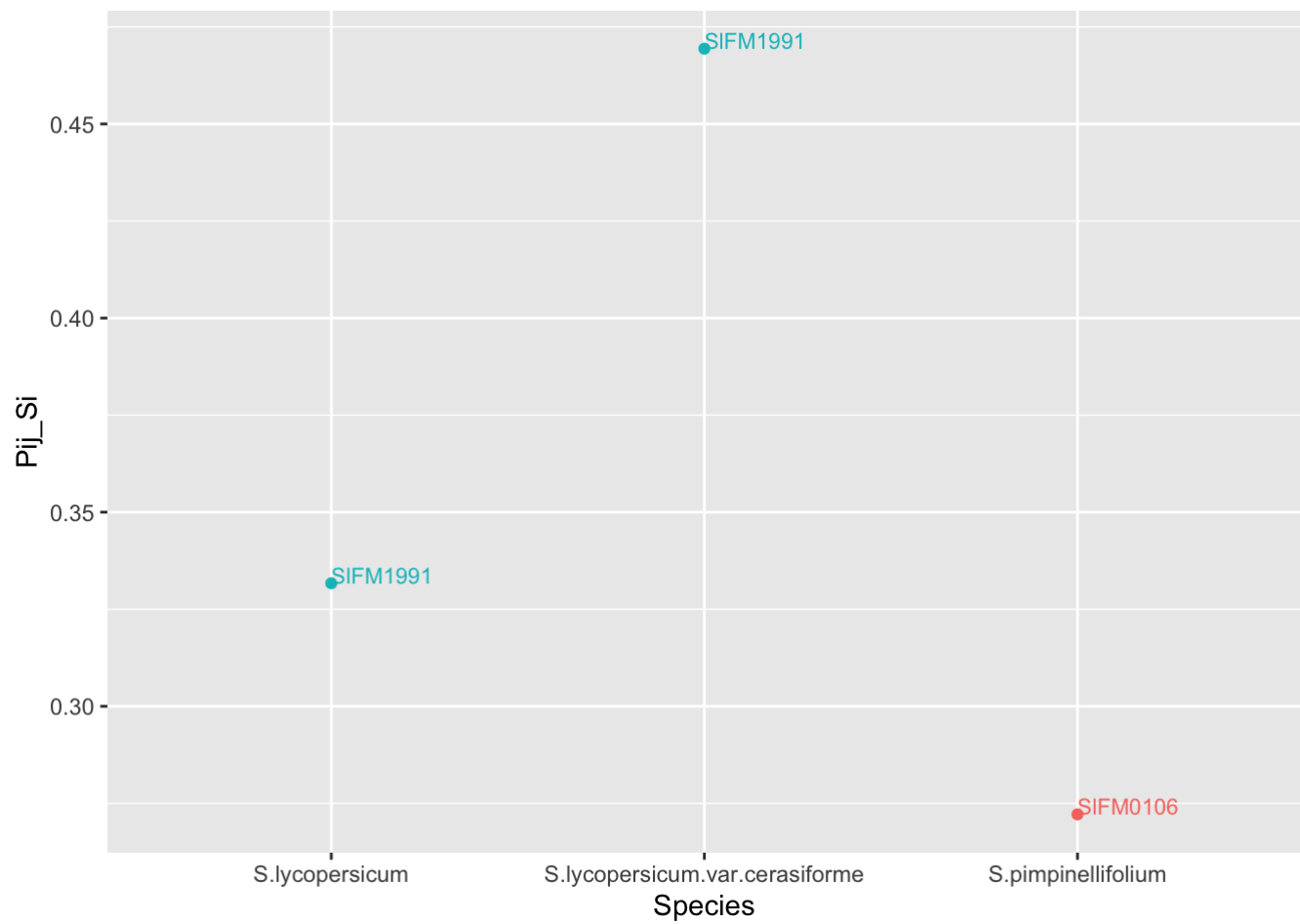
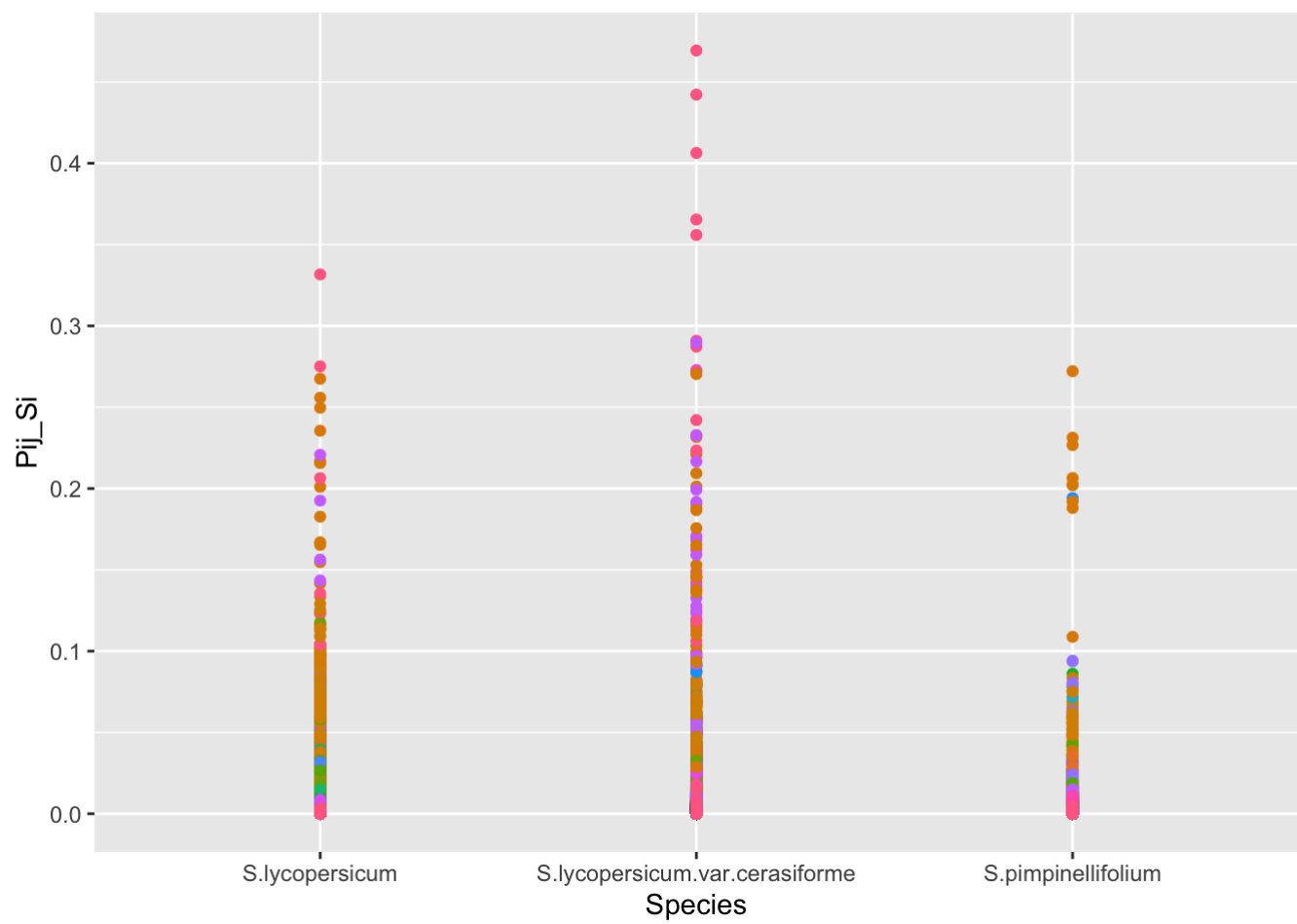
MetliteSpec calculates the contribution of METaboLITE SPECialization factor ($P_{ij}.S_i$) to the Metabolome specialization index.

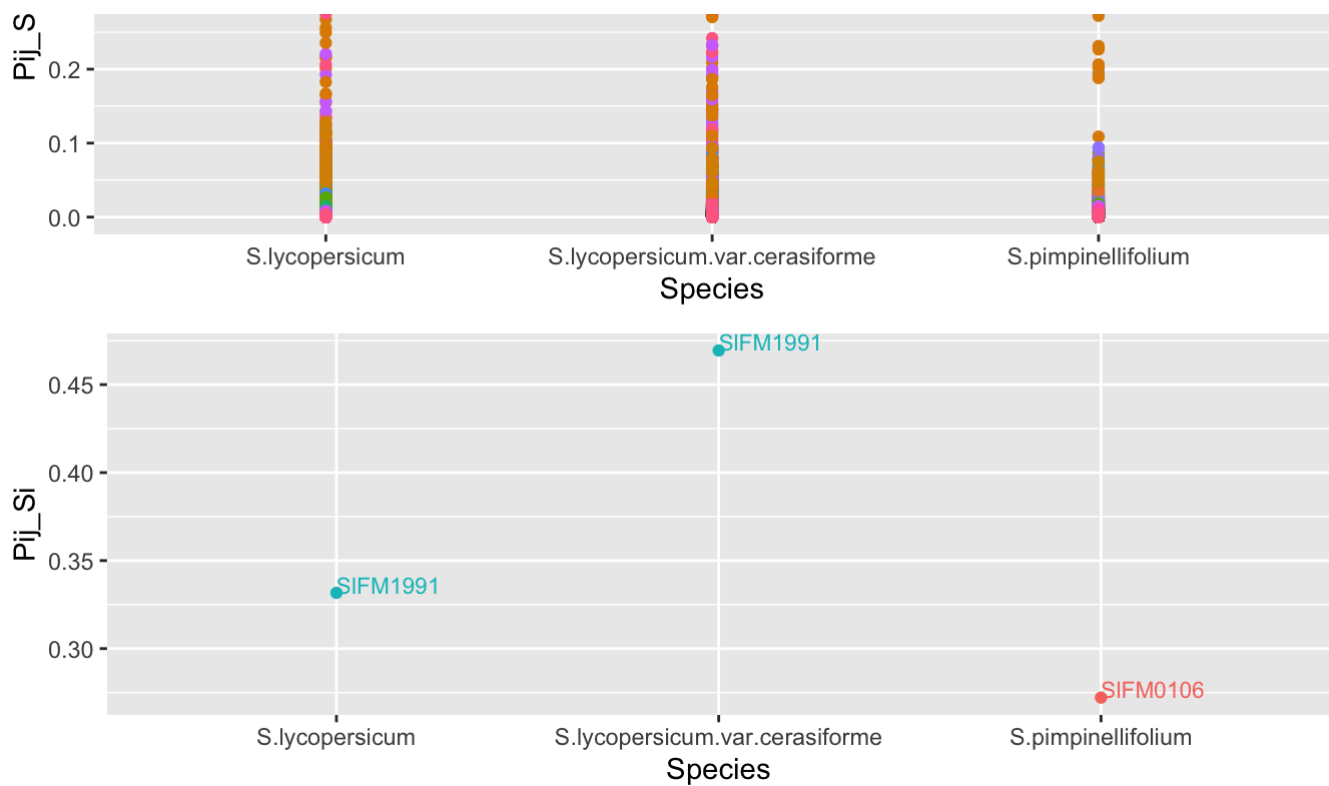
Metabolite specialization factor ($P_{ij}.S_i$) is defined as product of the a metabolite specilization index (S_i) and the frequency of the metabolite in a given sample (P_{ij}).

This function, returns a list with 4 objects explained below:

```
MetliteSpec <- MetliteSpec(Data)
```

```
## [1] "THIS PARAMETER CAN NOT BE EXTRAPOLATED TO OTHER DATA SETS OR SUBSETS"
## # A tibble: 393,960 × 3
##   Species      Compounds    Pij_Si
##   <chr>      <chr>      <dbl>
## 1 S.lycopersicum SlFM0001  0.000103
## 2 S.lycopersicum SlFM0002  0.0000200
## 3 S.lycopersicum SlFM0003  0.0000614
## 4 S.lycopersicum SlFM0004  0.000129
## 5 S.lycopersicum SlFM0006  0.000637
## 6 S.lycopersicum SlFM0007  0.000969
## 7 S.lycopersicum SlFM0008  0.000903
## 8 S.lycopersicum SlFM0009  0.0000799
## 9 S.lycopersicum SlFM0010  0.000603
## 10 S.lycopersicum SlFM0011  0.000605
## # ... with 393,950 more rows
## # A tibble: 3 × 3
## # Groups:   Species [3]
##   Species      Compounds    Pij_Si
##   <chr>      <chr>      <dbl>
## 1 S.lycopersicum      SlFM1991  0.332
## 2 S.lycopersicum.var.cerasiforme SlFM1991  0.469
## 3 S.pimpinellifolium    SlFM0106  0.272
```



```
## [[1]]
## TableGrob (2 x 1) "arrange": 2 grobs
##   z      cells      name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (2-2,1-1) arrange gtable[layout]
##
## [[2]]
## # A tibble: 393,960 x 3
##   Species      Compounds    Pij_Si
##   <chr>         <chr>      <dbl>
## 1 S.lycopersicum SlFM0001  0.000103
## 2 S.lycopersicum SlFM0002  0.0000200
## 3 S.lycopersicum SlFM0003  0.0000614
## 4 S.lycopersicum SlFM0004  0.000129
## 5 S.lycopersicum SlFM0006  0.000637
## 6 S.lycopersicum SlFM0007  0.000969
## 7 S.lycopersicum SlFM0008  0.000903
## 8 S.lycopersicum SlFM0009  0.0000799
## 9 S.lycopersicum SlFM0010  0.000603
## 10 S.lycopersicum SlFM0011  0.000605
## # ... with 393,950 more rows
##
## [[3]]
## # A tibble: 3 x 3
## # Groups:   Species [3]
##   Species      Compounds    Pij_Si
##   <chr>         <chr>      <dbl>
## 1 S.lycopersicum      SlFM1991    0.332
## 2 S.lycopersicum.var.cerasiforme SlFM1991    0.469
## 3 S.pimpinellifolium      SlFM0106    0.272
```

This function returns a list with 4 objects:

1. A data frame with the Pij.Si values per species and compound;

View(MetliteSpec[[2]]) allows to observe the contribution of each metabolite in each species to the Dj value.

```
print(MetliteSpec[[2]])
```

```
## # A tibble: 393,960 × 3
##   Species      Compounds    Pij_Si
##   <chr>        <chr>      <dbl>
## 1 S.lycopersicum SlFM0001  0.000103
## 2 S.lycopersicum SlFM0002  0.0000200
## 3 S.lycopersicum SlFM0003  0.0000614
## 4 S.lycopersicum SlFM0004  0.000129
## 5 S.lycopersicum SlFM0006  0.000637
## 6 S.lycopersicum SlFM0007  0.000969
## 7 S.lycopersicum SlFM0008  0.000903
## 8 S.lycopersicum SlFM0009  0.0000799
## 9 S.lycopersicum SlFM0010  0.000603
## 10 S.lycopersicum SlFM0011  0.000605
## # ... with 393,950 more rows
```

2. A data frame with the highest Pij.Si compound value per species;

View(MetliteSpec[[3]]) extracts the higher Pij.Si between all samples. Hence, indicating which is the metabolite that shows the higher degree of specificity in the data set under analysis.

NOTE: This function clusters the samples based on Species. In this particular example where there are more than one accessions per species, this information needs to be interpreted with caution.

```
print(MetliteSpec[[3]])
```

```
## # A tibble: 3 × 3
## # Groups:   Species [3]
##   Species      Compounds    Pij_Si
##   <chr>        <chr>      <dbl>
## 1 S.lycopersicum      SlFM1991  0.332
## 2 S.lycopersicum.var.cerasiforme SlFM1991  0.469
## 3 S.pimpinellifolium SlFM0106  0.272
```

3. A point plot depicting the Pij.Si value of each species

This plot is a visual representation of the data frame MetliteSpec[[2]]. In our example, it can be observed that the contribution of each metabolite in the specialization of each metabolome. This plot not only provides a general overview of each metabolite contribution to the Dj, but also facilitates the visualization of some important features such as clustering. The most obvious case happens in *S. pimpinellifolium*, where two sets of metabolite specialization clusters can be observed. Hence, it can be inferred that the high Dj value is mostly due to a small amount of highly specific metabolites.

4. A point plot depicting the highest Pij.Si value per species

This plot is a visual representation of the data frame MetliteSpec[[3]]. Here we can observe that the most specialized metabolite in *S. lycopersicum* and *S. lycopersicum* var. *cerasiforme* is the same one (SlFM1991), whereas for the wild relative *S. pimpinellifolium* is SlFM0106.

** Please, be aware that this data indicates the levels of specialization but not the levels of a certain compound **

Metabolic Plasticity Parameters Summary

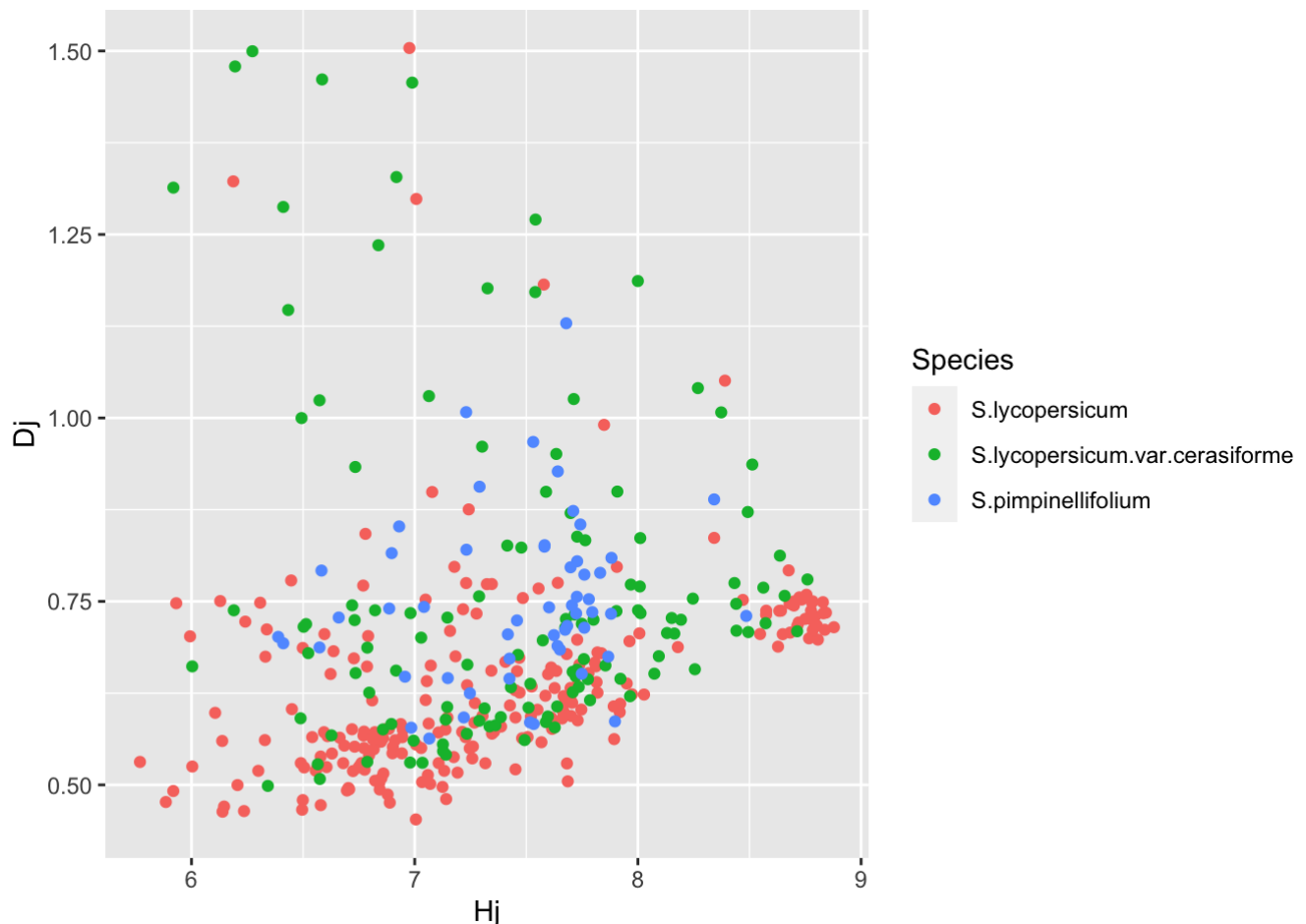
This functions generates a table summarizing the Hj, number of peaks, and Dj parameters per species.

It uses the data frames generated by MetDiv() and MetSpec(), a generates a new data frame that allows the pairwise comparison of the different parameters.

It returns a list with two objects explained below.

```
MetPar <- MetPar(Data)
```

```
## [1] "MetDiv table must be store as Hj, and MetSpec as Dj"
## [[1]]
##           Species      Hj numb_peaks      Dj
## 1:   S.lycopersicum 7.660288      948 0.5904608
## 2:   S.lycopersicum 7.634306      976 0.6552455
## 3:   S.lycopersicum 7.597500      963 0.6507864
## 4:   S.lycopersicum 7.505589      936 0.5651353
## 5:   S.lycopersicum 8.006721      965 0.7064863
## ---
## 398: S.pimpinellifolium 7.879622      980 0.7331459
## 399: S.pimpinellifolium 7.866917      971 0.6747583
## 400: S.pimpinellifolium 7.639788      980 0.6893213
## 401: S.pimpinellifolium 7.881527      966 0.8093060
## 402: S.pimpinellifolium 7.780209      979 0.7529746
##
## [[2]]
```



This function returns two objects:

1. A data frame with all the summarized information.

View(MetPar[[1]]) includes col1: Species; col2: Hj; col3: numb_peaks; col4: Dj.

```
print(MetPar[[1]])
```

```
##           Species      Hj numb_peaks      Dj
##  1:   S.lycopersicum 7.660288      948 0.5904608
##  2:   S.lycopersicum 7.634306      976 0.6552455
##  3:   S.lycopersicum 7.597500      963 0.6507864
##  4:   S.lycopersicum 7.505589      936 0.5651353
##  5:   S.lycopersicum 8.006721      965 0.7064863
##  ---
## 398: S.pimpinellifolium 7.879622      980 0.7331459
## 399: S.pimpinellifolium 7.866917      971 0.6747583
## 400: S.pimpinellifolium 7.639788      980 0.6893213
## 401: S.pimpinellifolium 7.881527      966 0.8093060
## 402: S.pimpinellifolium 7.780209      979 0.7529746
```

This data frame can be use to evaluate the statistical significance of the observed differences :

```
# Statistical Analysis

## Extracting Parameters data frame

MetPar_df <- as.data.frame(MetPar[[1]])

## ANOVA analysis

library (ggpubr)

compare_means(Hj ~ Species, data = MetPar_df)
```

```
## # A tibble: 3 × 8
##   .y.   group1      group2      p p.adj p.format p.signif method
##   <chr> <chr>      <chr>      <dbl> <dbl> <chr>      <chr>      <chr>
## 1 Hj    S.lycopersicum S.lycopersicum... 0.0902  0.18 0.090      ns      Wilco...
## 2 Hj    S.lycopersicum S.pimpinellifol... 0.0424  0.13 0.042      *      Wilco...
## 3 Hj    S.lycopersicum.v... S.pimpinellifol... 0.949   0.95 0.949      ns      Wilco...
```

2. A plot to visualize the dependency between Hj and Dj.

This point plot shows the dependency between the metabolic diversity (Hj) and the metabolome specialization (Dj) in each sample. In this case there is not a clear general trend. However, this plot suggests that accessions from different species might have a different trend. Thus, while *S. lycopersicum* var. *cerasiforme* shows a tendency to have a more specialized metabolome in those accessions with lower metabolic diversity, there is not a clear correlation in the case of *S. lycopersicum*.

Metabolic Plasticity Statistics

Besides the general statistical methods, MetPlast package includes two extra statistical parameters. MetStats functions generates a table summarizing the Hj, and the divergence associated to each Hj (HRj) and the Kullback–Leibler divergence (Divj). This function takes the data frame generated by `Hj <- MetDiv()`.

```
MetStats <- MetStats(Data)
```

```
## [1] "MetDiv table must be store as Hj"
##           Species      Hj numb_peaks      HRj      Divj
## 1:      S.lycopersicum 7.660288      948 8.138887 0.4785987
## 2:      S.lycopersicum 7.634306      976 8.198506 0.5642006
## 3:      S.lycopersicum 7.597500      963 7.998197 0.4006972
## 4:      S.lycopersicum 7.505589      936 8.107272 0.6016825
## 5:      S.lycopersicum 8.006721      965 8.464697 0.4579765
## ---
## 398: S.pimpinellifolium 7.879622      980 8.458192 0.5785702
## 399: S.pimpinellifolium 7.866917      971 8.314790 0.4478731
## 400: S.pimpinellifolium 7.639788      980 8.130422 0.4906336
## 401: S.pimpinellifolium 7.881527      966 8.535707 0.6541799
## 402: S.pimpinellifolium 7.780209      979 8.332928 0.5527190
```

The obtained data fram includes two statistical parameters:

1. HRj measures the divergence with respect to the whole average metabolome. HRj will be equal to or larger than the corresponding Hj.
2. Divj is define as the Kullback–Leibler divergence of the sample j. Divj measures how much a given sample j departs from the corresponding metabolome distribution of the whole system.