

# **Ординация и классификация с использованием мер сходства-различия**

**Математические методы в зоологии с использованием R**

Марина Варфоломеева

# Меры сходства и различия, ординация, классификация

## Вы сможете

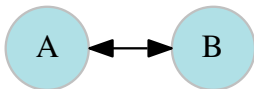
- Выбирать подходящий для данных коэффициент сходства/различия
- Представлять многомерные данные в меньшем числе измерений при помощи неметрического многомерного шкалирования
- Строить дендрограммы при помощи подходящего метода агрегации

# Коэффициенты сходства и различия

# Коэффициенты сходства и различия

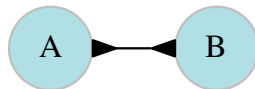
## Различия (dissimilarities)

$$d \geq 0$$



## Сходства (similarities)

$$0 \leq S \leq 1 \text{ или } -1 \leq S \leq 1$$



- Используются в качестве исходных данных для многих видов многомерных анализов, в т.ч. для неметрического многомерного шкалирования и некоторых видов кластерного анализа
- Из сходств можно получить расстояния и наоборот
- Свои коэффициенты для количественных и качественных признаков

# Свойства коэффициентов сходства-различия

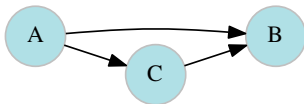
## Метрики и полуметрики

Адекватность:  $d_{A,A} = 0$



## Только метрики

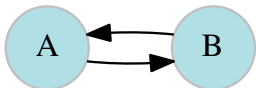
Триангулярность:  $d_{A,B} \leq d_{A,C} + d_{C,B}$



## Неметрики

Все остальное

Симметричность:  $d_{A,B} = d_{B,A}$



# Свойства коэффициентов сходства-различия

## Нестандартные

$$-\infty \leq d \leq \infty$$

## Стандартные

$$d_{min} \leq d \leq d_{max}$$

- частный случай стандартных коэффициентов - коррелятивные коэффициенты сходства

$$-1 \leq S \leq 1$$

# Примеры коэффициентов сходства-различия

## Метрики (расстояния, distances):

- без стандартизации:
  - Евклидово расстояние
  - Манхеттен (расстояние городских кварталов)
- со стандартизацией:
  - Канберра
  - хи-квадрат
  - Евклидово расстояние, рассчитанное по стандартизованным данным

## Полуметрики:

- расстояние Махаланобиса

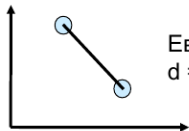
## Неметрики:

- со стандартизацией:
  - коррелятивные:
    - корреляция Пирсона
  - некоррелятивные:
    - коэффициент Брея-Куртиса

# Если количественные признаки измерены в одинаковых шкалах

## Метрики без стандартизации

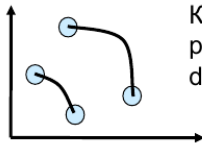
- Евклидово расстояние



Евклидово расстояние  
 $d = [\sum (x_{ik} - x_{jk})^2]^{-1/2}$

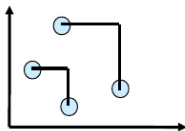
## Неевклидовы метрики

- Квадрат Евклидова расстояния



Квадрат Евклидова расстояния  
 $d = (1/k) \cdot \sum (x_{ik} - x_{jk})^2$

- Манхэттенское расстояние



Манхэттенское расстояние  
(городские кварталы)  
 $d = 1/k \sum |x_{ik} - x_{jk}|$



# Если количественные признаки измерены в разных шкалах

## Можно стандартизовать исходные данные

- Евклидово (или другое) расстояние, рассчитанное по стандартизованным данным

## Можно использовать коэффициенты со стандартизацией

- Канберра (метрика)  $d = \sum \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|}$
- хи-квадрат (метрика)  $\chi^2 = \sqrt{\sum \frac{1}{c_k} (x_{ik} - x_{jk})^2}$
- Коэффициент Махаланобиса (неметрика)  $d = \frac{\sum (x_{ik} - x_{jk})}{\sigma^2}$
- Корреляция Браве-Пирсона (коррелятивный)  $S = \frac{\sum (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{n\sigma_i^2\sigma_j^2}$
- Коэффициент Брея-Куртиса (не метрика)  $BC_{ij} = \frac{2C_{ij}}{S_i + S_j}$ , где  $C_{ij}$  - сумма минимальных значений из тех, которые не равны нулю для обоих объектов,  $S_i$  и  $S_j$  - общее число ненулевых значений признаков для обоих объектов.

# Если признаки — подсчеты численности

## Можно стандартизовать исходные данные

Простая стандартизация не подходит (счет, не может быть среднее 0)

Можно использовать трансформации:

- корень, корень 4-й степени
- логарифмирование со сдвигом ( $\log_{10}(x + 1)$ )

## Можно использовать коэффициенты со стандартизацией

- Канберра (метрика)  $d = \sum \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|}$
- хи-квадрат (метрика)  $\chi^2 = \sqrt{\sum \frac{1}{c_k} (x_{ik} - x_{jk})^2}$

## Если признаки — доли или проценты

- хи-квадрат (метрика)  $\chi^2 = \sqrt{\sum \frac{1}{c_k} (x_{ik} - x_{jk})^2}$
- коэффициент Брея-Куртиса (не метрика)  $BC_{ij} = \frac{2C_{ij}}{S_i + S_j}$
- Евклидово расстояние  $d = \sqrt{\sum (x_{ik} + x_{jk})^2}$

# Если используются бинарные данные (присутствие-отсутствие признака)

I \ J	+	-
+	a	b
-	c	d

I, J – множества

$$n_J = a + c \quad n_I = a + b$$

$$n = a + b + c + d$$

	I	J	
1	+	+	a – сходство по наличию
2	+	-	b – различие
3	-	+	c – различие
4	-	-	d – сходство по отсутствию

# Примеры коэффициентов для качественных данных

I \ J	+	-
+	a	b
-	c	d

I, J – множества

$$n_J = a + c \quad n_I = a + b$$

$$n = a + b + c + d$$

Коэффициент	Формула	Источник
Simple matching	$\frac{a+d}{a+b+c+d}$	Sokal and Michener, 1958
Rogers and Tanimoto	$\frac{a+d}{a+2b+2c+d}$	Rogers and Tanimoto, 1960
Anderberg	$\frac{a+d}{a+2(b+c)}$	Anderberg, 1973
Russel and Rao	$\frac{a}{a+b+c+d}$	Russel and Rao, 1940
Jaccard	$\frac{a}{a+b+c}$	Jaccard, 1901
Sorensen- <b>Dice</b>	$\frac{2a}{2a+b+c}$	Dice, 1945, Sorensen, 1948

И т.д.

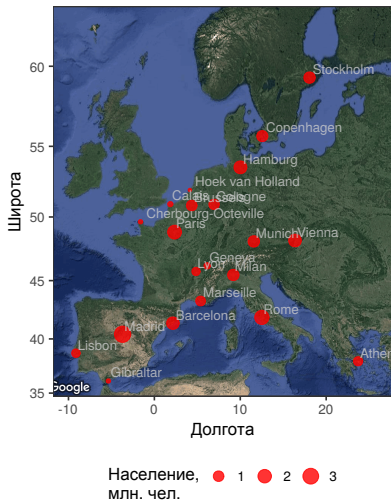
# Если данные смешанные (качественные и количественные)

## Коэффициенты для смешанных данных

- расстояние Говера

# Неметрическое многомерное шкалирование

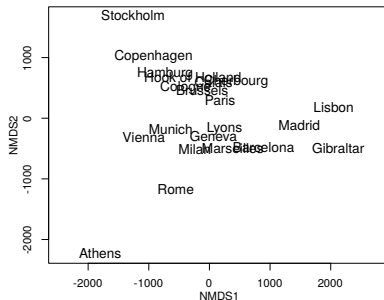
# Неметрическое многомерное шкалирование визуализирует отношения между объектами на основе расстояний между ними



Если бы мы знали расстояния по автодорогам между городами Европы

#	Athens	Barcelona	Brussels	Calais
# Barcelona	3313			
# Brussels	2963	1318		
# Calais	3175	1326	204	
# Cherbourg	3339	1294	583	460

мы бы смогли восстановить по ним карту



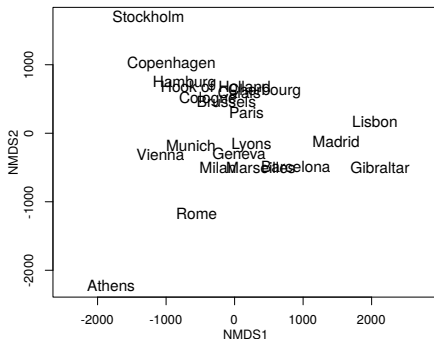


# Неметрическое многомерное шкалирование

Неметрическое многомерное шкалирование (nonmetric multidimensional scaling, nMDS) — метод визуализации отношений между объектами в пространстве с небольшим числом измерений.

Исходные данные — матрица расстояний между объектами в многомерном пространстве

- nMDS подбирает расстояния между объектами на графике так, чтобы сохранились соотношения исходных расстояний между ними. Т.е. если исходно А и В были ближе, чем В и С, то и в результате они должны быть ближе, чем В и С.
- Ординацию nMDS можно поворачивать, отражать, сдвигать - результат от этого не изменится.



## Пример: Морфометрия поссумов



© Hasitha Tudugalle | Photography

{Possum by Hasitha Tudugalle on Flickr

## Знакомимся с данными

```
library(DAAG)
data(possum)
colnames(possum)
```

```
# [1] "case"      "site"      "Pop"       "sex"       "age"       "hdlngh"
# [7] "skullw"    "totlngh"   "tail"     "footlgth"  "earconch"  "eye"
# [13] "chest"     "belly"
```

```
colSums(is.na(possum))
```

```
#      case      site      Pop      sex      age      hdlngh      skullw
#         0         0         0         0         2         0         0
# totlngh      tail footlgth earconch      eye      chest      belly
#         0         0         1         0         0         0         0
```

*# оставим только строки с полными наблюдениями*

```
pos <- possum[complete.cases(possum), ]
```

*# поссумы из разных сайтов из 2 популяций*

```
table(pos$Pop, pos$site)
```

```
#
#      1  2  3  4  5  6  7
```

# Неметрическое многомерное шкалирование

Построим ординацию поссумов на основе их сходства по морфометрии и возрасту.

Функция `metaMDS` много раз итеративно подбирает координаты поссумов в новом пространстве (двумерном по умолчанию) и сохраняет лучшую конфигурацию.

`autotransform` — если `TRUE`, то данные предварительно подвергаются двойной (“висконсинской”) стандартизации (см. `?metaMDS`). **Если у вас не данные о сообществах, то это нужно отключить**

```
library(vegan)
ord_euclid <- metaMDS(pos[, 6:14], distance = "euclid", autotransform = FALSE)
```

```
# Run 0 stress 0.1034941
# Run 1 stress 0.1034941
# ... New best solution
# ... Procrustes: rmse 0.000007013358   max resid 0.0000507957
# ... Similar to previous best
# Run 2 stress 0.1034941
# ... Procrustes: rmse 0.000006254376   max resid 0.00005470417
# ... Similar to previous best
# Run 3 stress 0.1034941
# ... Procrustes: rmse 0.000008171415   max resid 0.00007757897
# ... Similar to previous best
```

## Качество подгонки модели

**stress** - оценивает, насколько были искажены исходные расстояния между объектами при снижении размерности

```
ord_euclid$stress
```

```
# [1] 0.1034941
```

- Эмпирическое правило: хорошо  $< 0.25$  (или, иногда,  $0.20$ )  $<$  плохо

# Ординация

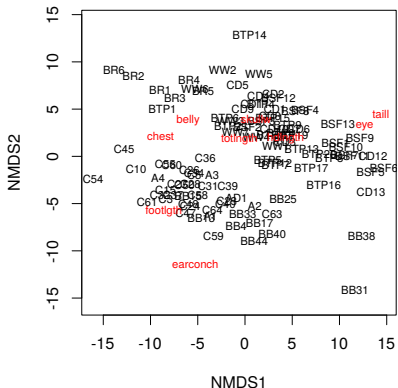
Координаты наблюдений:

```
head(ord_euclid$points, 10)
```

#		MDS1	MDS2
# C3		-8.367322	-4.596269
# C5		-5.348827	-2.057263
# C10		-11.487014	-1.414180
# C15		-8.360981	-3.624207
# C23		-1.886223	-4.743281
# C24		-5.778006	-5.346202
# C26		-5.837552	-1.542074
# C27		-7.101858	-2.983519
# C28		-5.743909	-2.972527
# C31		-3.875284	-3.236825

График ординации:

```
ordiplot(ord_euclid, type = "t")
```



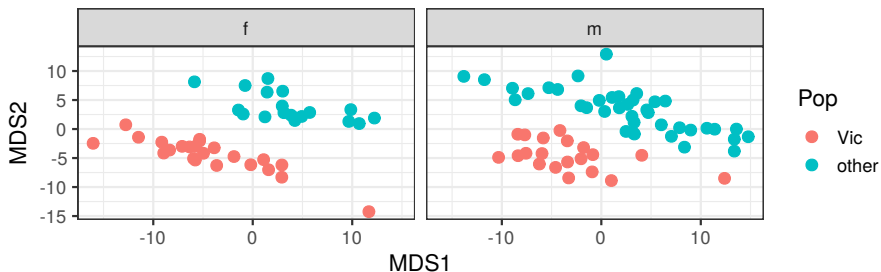
## Задание 1

При помощи `ggplot2` постройте график неметрического многомерного шкалирования.

Для графика используйте координаты точек `ord_euclid$points` и исходные данные.

Раскрасьте график по значениям переменной `Pop`.

Сделайте так, чтобы особи разного пола были изображены на разных панелях



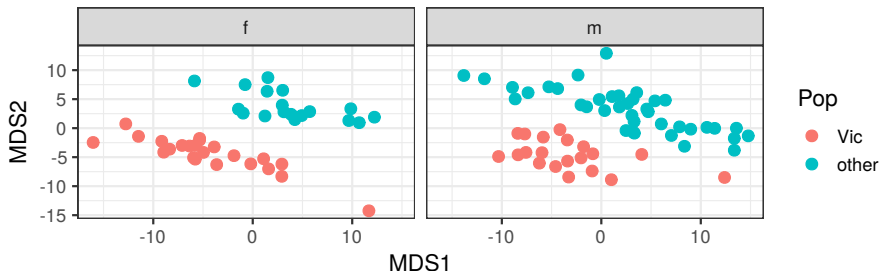
## Дополните код

```
library()  
# Данные для графика  
points_euclid <- data.frame( , )  
# График nMDS ординации  
gg_euclid <- ggplot(, aes(x = , y = )) +  
  geom_point() +  
  facet_wrap(~ )  
gg_euclid
```



## Решение: график ординации

```
library(ggplot2)
# Данные для графика
points_euclid <- data.frame(pos, ord_euclid$points)
# График nMDS ординации
gg_euclid <- ggplot(points_euclid, aes(x = MDS1, y = MDS2)) +
  geom_point(aes(colour = Pop)) +
  facet_wrap(~sex)
gg_euclid
```



## Задание 2

Постройте nMDS ординацию при помощи евклидова расстояния, по стандартизованным данным

Дополните код

```
# Ординация  
ord_scaled <- metaMDS( (pos), distance = , autotransform = )  
# Качество ординации
```

## Решение:

*# Ординация*

```
ord_scaled <- metaMDS(scale(pos[, 6:14]), distance = "euclidean", autotransform = FALSE)
```

```
# Run 0 stress 0.1471067
# Run 1 stress 0.1471067
# ... New best solution
# ... Procrustes: rmse 0.00006226554  max resid 0.0003598938
# ... Similar to previous best
# Run 2 stress 0.1509339
# Run 3 stress 0.1471097
# ... Procrustes: rmse 0.001068262  max resid 0.005866718
# ... Similar to previous best
# Run 4 stress 0.1471096
# ... Procrustes: rmse 0.001065267  max resid 0.005862186
# ... Similar to previous best
# Run 5 stress 0.1509505
# Run 6 stress 0.1471096
# ... Procrustes: rmse 0.001065057  max resid 0.005861617
# ... Similar to previous best
# Run 7 stress 0.1509339
# Run 8 stress 0.1509504
# Run 9 stress 0.1509341
# Run 10 stress 0.1471028
# ... New best solution
# ... Procrustes: rmse 0.0008156886  max resid 0.005886155
```

# График ординации по матрице евклидовых расстояний, рассчитанных по стандартизованным данным

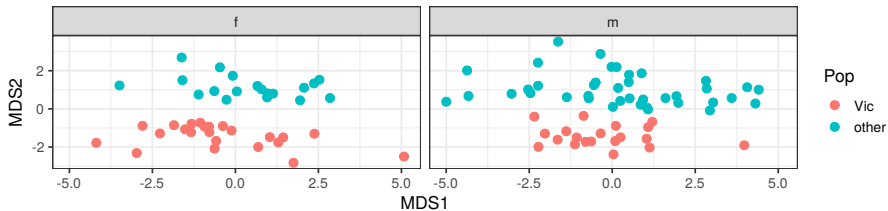
# Данные для графика

```
points_scaled <- data.frame(ord_scaled$points, pos)
```

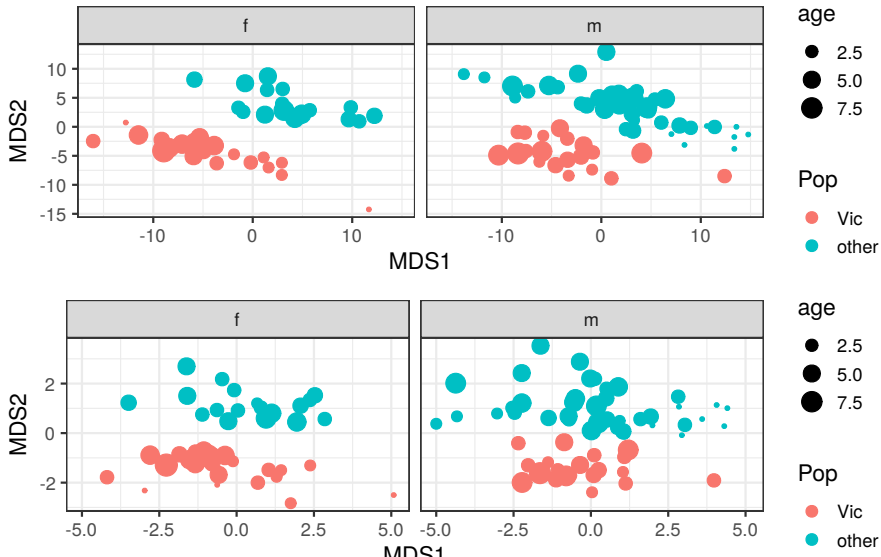
# График nMDS-ординации

```
gg_scaled <- gg_euclid %>% points_scaled
```

```
gg_scaled
```



## Видно, что графики ординации, полученные разными методами, различаются



## Код для графиков

```
library(gridExtra)
grid.arrange(gg_euclid + aes(size = age),
              gg_scaled + aes(size = age),
              ncol = 1)
```

# Кластерный анализ

## Пример: Морфометрия самок поссумов

```
# library(DAAG)
data(fossum)
```

Данные для кластерного анализа нужно подготовить:

- создать осмысленные имена строк
- выбрать только переменные, нужные для построения матрицы сходств-различий
- выбрать только строки без пропусков



## Создаем “говорящие” названия строк

Сейчас в названиях строк записано из какой точки каждый поссум

```
rownames(fossum) # было
```

```
# [1] "C5"      "C10"     "C15"     "C23"     "C24"     "C27"     "C28"     "C31"
# [9] "C32"     "C34"     "C45"     "C48"     "C50"     "C54"     "C58"     "C63"
# [17] "A1"      "A2"      "A4"      "BB17"    "BB31"    "BB33"    "BB36"    "BB40"
# [25] "WW4"     "WW5"     "WW7"     "BR4"     "BR7"     "CD2"     "CD3"     "CD4"
# [33] "CD5"     "CD6"     "CD10"    "BSF1"    "BSF9"    "BSF10"   "BSF13"   "BTP3"
# [41] "BTP15"   "BTP19"   "BTP21"
```

Чтобы имена строк были более информативны, добавим к ним название популяции

```
rownames(fossum) <- paste(fossum$Pop,
                           rownames(fossum),
                           sep = "_")
```

```
rownames(fossum) # стало
```

```
# [1] "Vic_C5"      "Vic_C10"     "Vic_C15"     "Vic_C23"
# [5] "Vic_C24"     "Vic_C27"     "Vic_C28"     "Vic_C31"
# [9] "Vic_C32"     "Vic_C34"     "Vic_C45"     "Vic_C48"
# [13] "Vic_C50"     "Vic_C54"     "Vic_C58"     "Vic_C63"
# [17] "Vic_A1"      "Vic_A2"      "Vic_A4"      "Vic_BB17"
# [21] "Vic_BB31"    "Vic_BB33"    "Vic_BB36"    "Vic_BB40"
# [25] "Vic_WW4"     "Vic_WW5"     "Vic_WW7"     "Vic_BR4"
# [29] "Vic_BR7"     "Vic_CD2"     "Vic_CD3"     "Vic_CD4"
# [33] "Vic_CD5"     "Vic_CD6"     "Vic_CD10"    "Vic_BSF1"
# [37] "Vic_BSF9"    "Vic_BSF10"   "Vic_BSF13"   "Vic_BTP3"
# [41] "Vic_BTP15"   "Vic_BTP19"   "Vic_BTP21"
```

## Отбираем только то, что понадобится для кластеризации

Отбираем

только строки без пропущенных значений, и только столбцы с морфометрическими данными

```
fos <- fossum[complete.cases(fossum), 6:14]
```

# Какие бывают методы построения деревьев?

## Методы кластеризации на основании расстояний (о них сегодня пойдет речь)

- Метод ближайшего соседа (single linkage)
- Метод отдаленного соседа (complete linkage)
- Метод среднегруппового расстояния (average linkage, UPGMA)
- Метод Варда (Ward's method)
- Метод присоединения соседей (Neighbour Joining)

Эти методы есть в базовом пакете `stats`, и в пакете `ape`. Разные полезные функции есть в `ade4` и `adeigenet`

## Методы кластеризации на основании признаков

- Метод максимальной бережливости
- Метод максимального правдоподобия

Эти методы реализованы в пакете `phangorn`

Со списком пакетов для филогенетического анализа в R можно познакомиться здесь:

<https://cran.r-project.org/web/views/Phylogenetics.html>

# Методы кластеризации на основании расстояний

# От чего зависит результат кластеризации

Результат кластеризации зависит от

- коэффициента сходства-различия
- от алгоритма кластеризации

## Кластерный анализ начинается с расчета матрицы расстояний между объектами

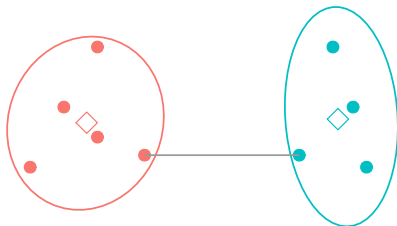
Далее мы будем использовать матрицу евклидовых расстояний между поссумами.

```
d <- dist(x = fos, method = "euclidean")
```

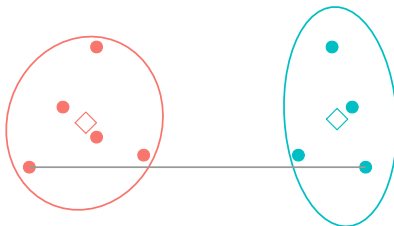
Давайте построим деревья при помощи нескольких алгоритмов кластеризации и сравним их.

# Методы кластеризации

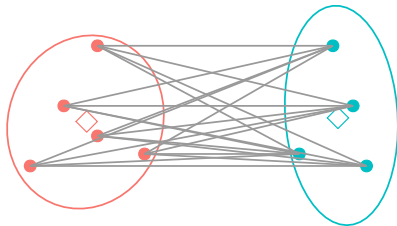
Метод ближайшего соседа



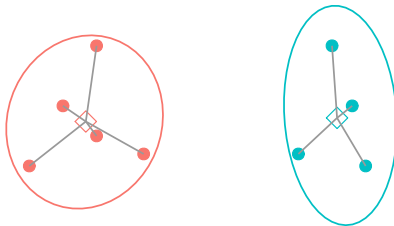
Метод отдаленного соседа



Метод среднegrupпового расстояния

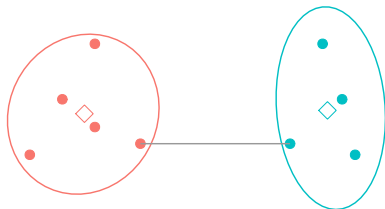


Метод Варда



# Метод ближайшего соседа

- = nearest neighbour = single linkage
- к кластеру присоединяется ближайший к нему кластер/объект
- кластеры объединяются в один на расстоянии, которое равно расстоянию между ближайшими объектами этих кластеров



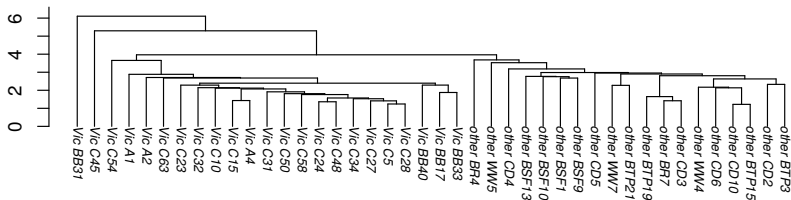
## Особенности:

- Может быть сложно интерпретировать, если нужны группы
  - объекты на дендрограмме часто не образуют четко разделенных групп
  - часто получаются цепочки кластеров (объекты присоединяются как бы по-одному)
- Хорош для выявления градиентов



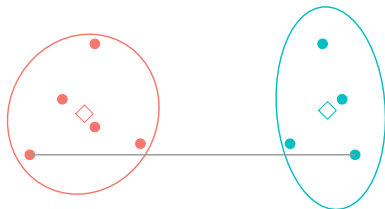
# Метод ближайшего соседа в R

```
hc_single <- hclust(d, method = "single")
library(ape)
ph_single <- as.phylo(hc_single)
# cex - относительный размер шрифта
plot(ph_single, type = "phylogram", direction = "downwards", cex = 0.7)
axisPhylo(side = 2)
```



# Метод отдаленного соседа

- = furthest neighbour = complete linkage
- к кластеру присоединяется отдаленный кластер/объект
- кластеры объединяются в один на расстоянии, которое равно расстоянию между самыми отдаленными объектами этих кластеров (следствие - чем более крупная группа, тем сложнее к ней присоединиться)

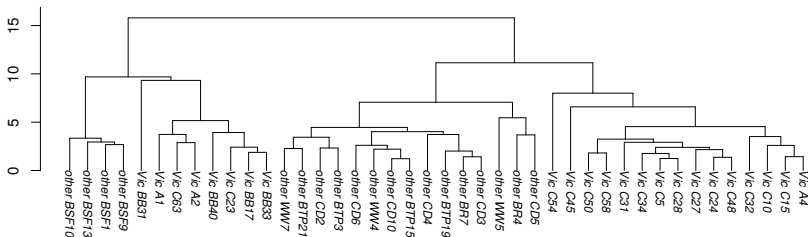


## Особенности:

- На дендрограмме образуется много отдельных не крупных групп
- Хорош для поиска дискретных групп в данных

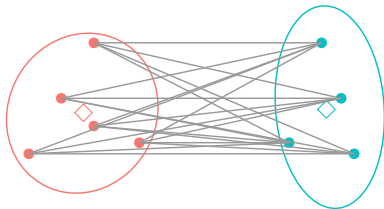
## Метод отдаленного соседа в R

```
ph_compl <- as.phylo(hclust(d, method = "complete"))  
plot(ph_compl, type = "phylogram", direction = "downwards", cex = 0.8)  
axisPhylo(side = 2)
```

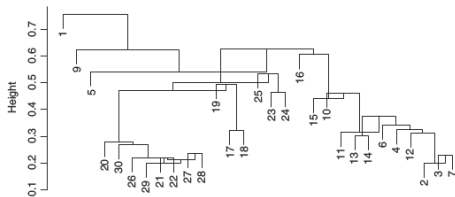


# Метод невзвешенного попарного среднего

- = UPGMA = Unweighted Pair Group Method with Arithmetic mean
- кластеры объединяются в один на расстоянии, которое равно среднему значению всех возможных расстояний между объектами из разных кластеров.



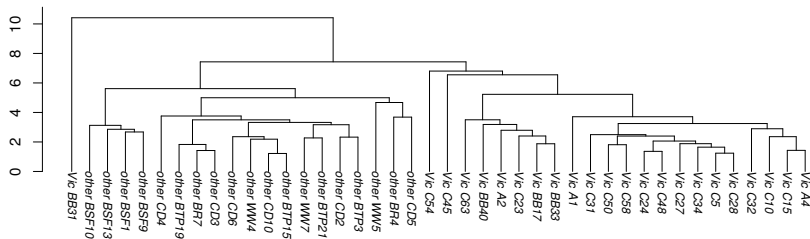
**Особенности:** UPGMA и WUPGMC иногда могут приводить к инверсиям на дендрограммах



{из Borcard et al., 2011}

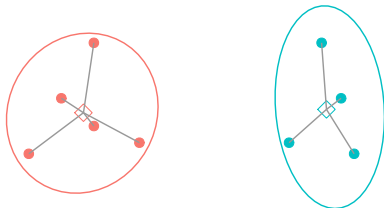
## Метод невзвешенного попарного среднего в R

```
ph_avg <- as.phylo(hclust(d, method = "average"))  
plot(ph_avg, type = "phylogram", direction = "downwards", cex = 0.8)  
axisPhylo(side = 2)
```



# Метод Варда

- = Ward's Minimum Variance Clustering
- объекты объединяются в кластеры так, чтобы внутригрупповая дисперсия расстояний была минимальной

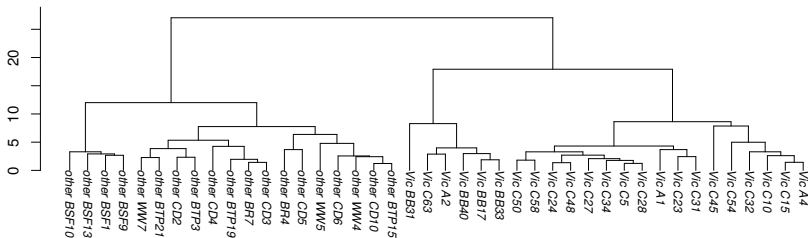


## Особенности:

- метод годится и для неевклидовых расстояний несмотря на то, что внутригрупповая дисперсия расстояний рассчитывается так, как будто это евклидовы расстояния

## Метод Варда в R

```
ph_w2 <- as.phylo(hclust(d, method = "ward.D2"))  
plot(ph_w2, type = "phylogram", direction = "downwards", cex = 0.8)  
axisPhylo(side = 2)
```



# Сравнение и интерпретация результатов кластеризации



## Кофенетическая корреляция

Кофенетическое расстояние - расстояние между объектами на дендрограмме

Кофенетическую корреляцию можно рассчитать как пирсоновскую корреляцию (обычную) между матрицами исходных и кофенетических расстояний между всеми парами объектов

Метод, который дает наибольшую кофенетическую корреляцию дает кластеры лучше всего отражающие исходные данные

## Кофенетическая корреляция в R

```
# Кофенетические расстояния  
c_single <- as.dist(cophenetic(ph_single))  
c_compl <- as.dist(cophenetic(ph_compl))  
c_avg <- as.dist(cophenetic(ph_avg))  
c_w2 <- as.dist(cophenetic(ph_w2))
```

```
# Кофенетические корреляции  
cor(d, c_single)
```

```
# [1] 0.7482612
```

```
cor(d, c_compl)
```

```
# [1] 0.5988573
```

```
cor(d, c_avg)
```

```
# [1] 0.8084252
```

```
cor(d, c_w2)
```

```
# [1] 0.6926997
```

## На каком уровне нужно делить дендрограмму на кластеры?

- Можно субъективно, на любом выбранном уровне. Главное, чтобы кластеры были осмысленными и интерпретируемыми.
- Можно выбрать, глядя на распределение расстояний ветвления
- Можно оценить вероятность разделения на кластеры при помощи бутстрепа

# Бутстреп-поддержка ветвей

Функция `system.time` - покажет, сколько времени заняли расчеты

Аргументы `pvclust`:

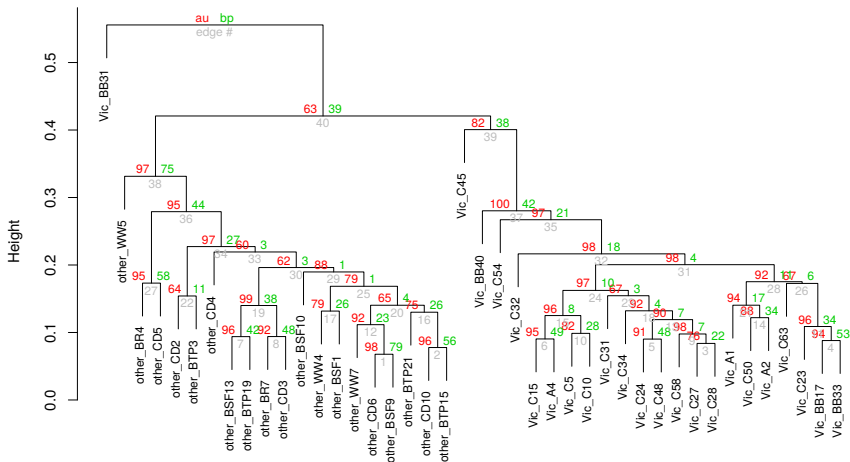
- `nboot` — число итераций должно быть больше 10000. В примере мы используем мало для скорости
- `parallel = TRUE` — проводить параллельные вычисления на нескольких ядрах процессора. Ускоряет расчеты
- `iseed` — зерно генератора случайных чисел для вычислений. Обязательно задавайте этот аргумент, если хотите, чтобы вычисления воспроизводились при повторных запусках

`library(pvclust)`

```
system.time({  
  cl_boot <- pvclust(scale(t(fos)),  
                    method.hclust = "average",  
                    method.dist = "euclidean",  
                    nboot = 5000,  
                    parallel = TRUE,  
                    iseed = 42)  
})
```

```
plot(cl_boot, cex.pv = 0.8, cex = 0.8)
```

Cluster dendrogram with p-values (%)



# Построение деревьев по генетическим данным

## Teaser

В этом курсе нет возможности рассказать даже о малой доле возможностей R для работы с генетическими данными, поэтому давайте сделаем небольшую демонстрацию.

## Пример: Митохондриальная ДНК приматов.

В файле `primates.dna` содержатся последовательности участка митохондриальной ДНК. для 12 видов приматов. Последовательности для мыши и коровы — в качестве аутгруппы. (232bp в контрольном участке плюс третий кодон в близлежащих белок-кодирующих митохондриальных генах — 1-2 кодона исключены в попытке получить сходную скорость эволюции во всех сайтах)

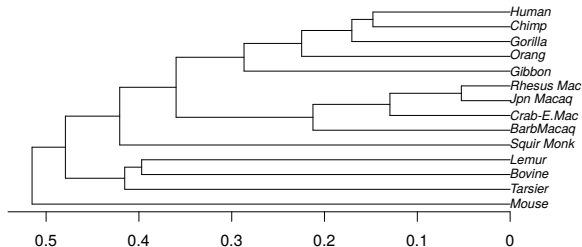
Датасет собан Dr. Masami Hasegawa (Institute of Statistical Mathematics, Tokyo), по данным сиквенирования Kenji Hayasaka, Takashi Gojobori, Satoshi Horai (Molecular Biology and Evolution 5: 626-644, 1988).

Исходный файл в формате PHYLIP можно загрузить по ссылке:  
<http://evolution.genetics.washington.edu/book/primates.dna>



## Дерево по генетическим данным

```
webpage <- "http://evolution.genetics.washington.edu/book/primates.dna"
primates.dna <- read.dna(webpage)
d_pri <- dist.dna(primates.dna, model = "K80")
hc_pri <- hclust(d_pri, method = "average")
ph_pri <- as.phylo(hc_pri)
plot(ph_pri, cex = 0.8)
axisPhylo()
```



# Take home messages

- Неметрическое многомерное шкалирование (nMDS):
  - nMDS — способ снижения размерности, сохраняющий ранги расстояний между объектами
  - Направления на графике многомерного шкалирования можно интерпретировать произвольным образом в зависимости от изменения других переменных (не обязательно вдоль осей)
  - Результат многомерного шкалирования зависит от выбора коэффициента различия
  - Стресс — мера оценки качества ординации nMDS
- Кластерный анализ:
  - Результат кластеризации зависит не только от выбора коэффициента, но и от выбора алгоритма кластеризации
  - Кофенетическая корреляция — мера оценки соответствия расстояний на дендрограмме и коэффициентов сходства/различия в исходной матрице

## Дополнительные ресурсы

- Borcard, D., Gillet, F., Legendre, P., 2011. Numerical ecology with R. Springer.
- Legendre, P., Legendre, L., 2012. Numerical ecology. Elsevier.
- Oksanen, J., 2011. Multivariate analysis of ecological communities in R: vegan tutorial. R package version 2-0.
- Quinn, G.G.P., Keough, M.J., 2002. Experimental design and data analysis for biologists. Cambridge University Press.
- Как работает UPGMA:  
<http://www.southampton.ac.uk/~re1u06/teaching/upgma/>
- pvclust: An R package for hierarchical clustering with p-values [WWW Document], n.d. URL  
<http://www.sigmath.es.osaka-u.ac.jp/shimo-lab/prog/pvclust/> (accessed 11.7.14).

Для анализа филогенетических данных:

- Paradis, E., 2011. Analysis of Phylogenetics and Evolution with R. Springer.
- Список пакетов для филогенетического анализа в R:  
<https://cran.r-project.org/web/views/Phylogenetics.html>