Multivariate Analysis

II: Constrained Ordination

Jari Oksanen

Oulu

January 2009

- Basic ordination methods to simplify multivariate data into low dimensional graphics
- Analysis of multivariate dependence and hypotheses
- Analyses can be performed in **R** statistical software using **vegan** package and allies
- Course homepage `http://cc.oulu.fi/~jarioksa/opetus/metodi/`
- **Vegan** homepage `http://vegan.r-forge.r-project.org/`

Outline

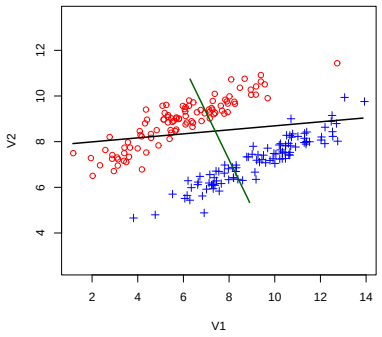
1. Constrained Ordination
   - Methods
   - Model Choice
   - Permutation Test
   - Partial Analysis
2. Analysis of Dissimilarities
   - Methods

Outline

1. Constrained Ordination
   - Methods
   - Model Choice
   - Permutation Test
   - Partial Analysis
2. Analysis of Dissimilarities
   - Methods

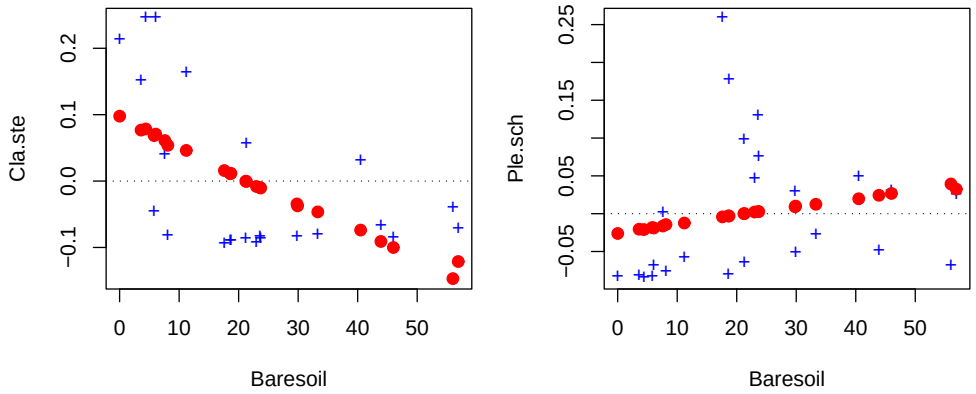
Constrained vs. Unconstrained

- Unconstrained ordination tries to display the variation in data.
- Constrained ordination tries to display only the variation that can be explained with constraining variables.
- You can observe only things that you have measured.

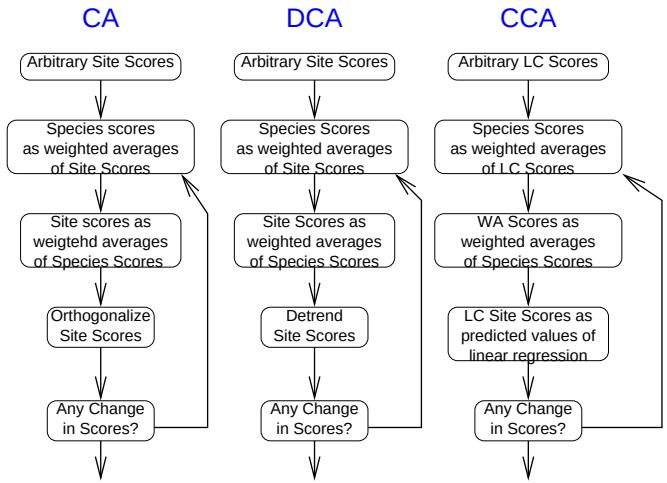


Constrained CA

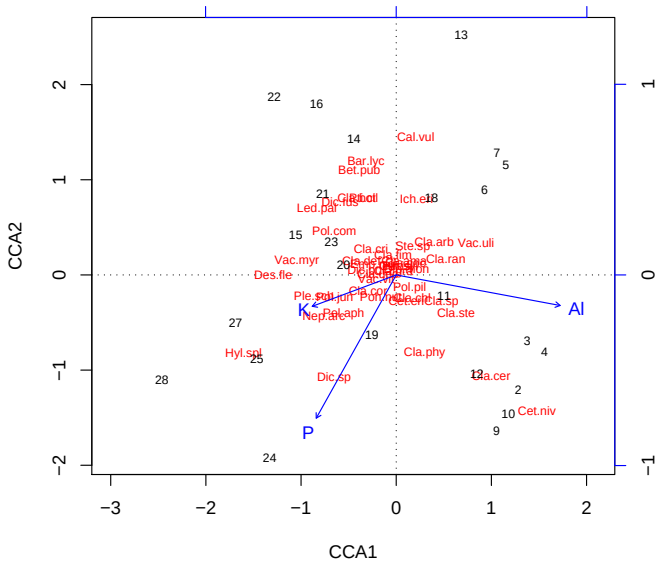
1. Fit weighted linear regression to all species individually using all constraints as explonatory variables
2. Analyse fitted values using CA



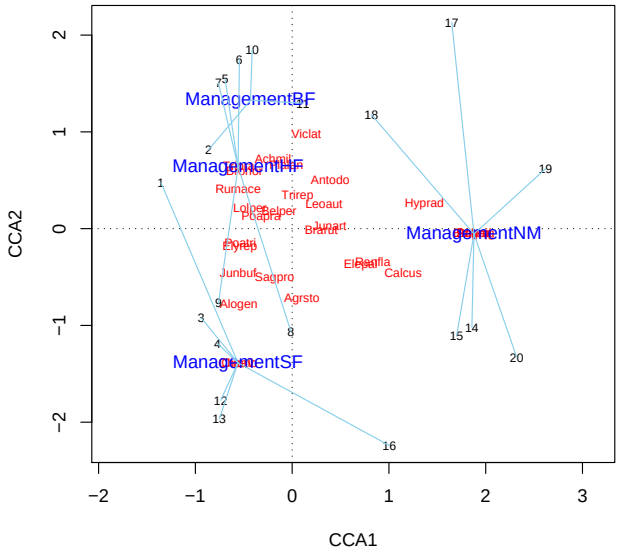
Alternative Algoritm: Alternate Regression and WA



Example: Continuous Constraints



Example: Class Constraints



Constrained Ordination

- Constrained analysis of proximities (CAP) in function `capscale` is related to metric multidimensional scaling (`cmdscale`). It can handle any dissimilarity measures and performs a linear mapping.
- Redundancy analysis (RDA) in function `rda` is related to principal components analysis. It is based on Euclidean distances and performs linear mapping.
- Constrained correspondence analysis (CCA) in function `cca` is related to correspondence analysis. It is based on Chi-squared distances and performs weighted linear mapping.

Running CCA I

```
> (ord <- cca(varespec, varechem))

Call: cca(X = varespec, Y = varechem)

              Inertia Rank
Total           2.083
Constrained     1.441   14
Unconstrained   0.642    9
Inertia is mean squared contingency coefficient

Eigenvalues for constrained axes:
   CCA1    CCA2    CCA3    CCA4    CCA5    CCA6    CCA7
0.43887 0.29178 0.16285 0.14213 0.11795 0.08903 0.07029
   CCA8    CCA9   CCA10   CCA11   CCA12   CCA13   CCA14
0.05836 0.03114 0.01329 0.00836 0.00654 0.00616 0.00473

Eigenvalues for unconstrained axes:
    CA1     CA2     CA3     CA4     CA5     CA6     CA7
0.19776 0.14193 0.10117 0.07079 0.05330 0.03330 0.01887
```


Running CCA II

```
    CA8     CA9
0.01510 0.00949

> head(summary(ord), 3)

Call:
cca(X = varespec, Y = varechem)

Partitioning of mean squared contingency coefficient:
              Inertia Proportion
Total           2.083      1.000
Constrained     1.441      0.692
Unconstrained   0.642      0.308

Eigenvalues, and their contribution to the mean squared contingency coefficie

            CCA1  CCA2  CCA3  CCA4  CCA5  CCA6   CCA7   CCA8
Eig.value  0.439 0.292 0.163 0.142 0.118 0.089 0.0703 0.0584
Accounted  0.211 0.351 0.429 0.497 0.554 0.596 0.6302 0.6582
             CCA9  CCA10   CCA11   CCA12   CCA13   CCA14    CA1
```

Running CCA III

```
Eig.value 0.0311 0.0133 0.00836 0.00654 0.00616 0.00473 0.198
Accounted 0.6732 0.6796 0.68359 0.68673 0.68969 0.69196 0.787
           CA2   CA3    CA4    CA5    CA6    CA7    CA8
Eig.value 0.142 0.101 0.0708 0.0533 0.0333 0.0189 0.0151
Accounted 0.855 0.904 0.9376 0.9632 0.9791 0.9882 0.9954
            CA9
Eig.value 0.00949
Accounted 1.00000

Accumulated constrained eigenvalues
 CCA1  CCA2  CCA3  CCA4  CCA5  CCA6  CCA7  CCA8  CCA9 CCA10
0.304 0.507 0.620 0.718 0.800 0.862 0.911 0.951 0.973 0.982
CCA11 CCA12 CCA13 CCA14
0.988 0.992 0.997 1.000

Scaling 2 for species and site scores
* Species are scaled proportional to eigenvalues
* Sites are unscaled: weighted dispersion equal on all dimensions
```


Running CCA IV

```
Species scores

          CCA1    CCA2   CCA3   CCA4   CCA5    CCA6
Cal.vul  0.0753 -0.9358 1.6777  0.696  1.078 -0.3450
Emp.nig -0.1813  0.0761 0.0365 -0.428 -0.138  0.0105
Led.pal -1.0535 -0.0603 0.0774 -0.939 -0.214 -0.5180
....


Site scores (weighted averages of species scores)

       CCA1   CCA2   CCA3   CCA4   CCA5  CCA6
18    0.178 -1.060 -0.409 -0.607 -0.565 0.242
15   -0.970 -0.197  0.421  0.303  0.152 0.804
24   -1.280  0.476 -2.947  0.393  3.954 0.766
....
```


Running CCA V

```
Site constraints (linear combinations of constraining variables)

       CCA1   CCA2   CCA3   CCA4    CCA5   CCA6
18   -0.423 -1.325 -0.492 -0.945 -0.0485  0.940
15   -0.190  0.497  0.455 -0.530 -0.0766 -0.790
24   -0.863  0.252 -2.760  0.570  3.2927  0.263
....


Biplot scores for constraining variables

          CCA1    CCA2     CCA3    CCA4     CCA5     CCA6
N       -0.223 -0.5287  0.00685  0.1778 -0.25359  0.10258
P       -0.319  0.5790 -0.16203  0.4795  0.18418 -0.12198
K       -0.366  0.3080  0.35983  0.4795  0.32551 -0.19676
Ca      -0.448  0.4218 -0.03779  0.0982  0.30808  0.04346
Mg      -0.435  0.3407 -0.14216  0.1080  0.49788 -0.00570
S       -0.024  0.4159  0.14840  0.4446  0.59712 -0.16631
Al       0.770 -0.0477  0.03755  0.3909  0.16111 -0.33702
```


Running CCA VI

```
Fe        0.649 -0.0886 -0.04218  0.2627 -0.06955 -0.11188
Mn       -0.722  0.2247  0.11306  0.2916 -0.13870  0.18055
Zn       -0.358  0.3352 -0.27789  0.3460  0.61920 -0.00103
Mo        0.205 -0.1028 -0.15689  0.3250  0.51625 -0.31305
Baresoil -0.537 -0.2538  0.13751 -0.5202  0.16592 -0.35143
Humdepth -0.697  0.2023  0.27184 -0.1353 -0.00363 -0.05074
pH        0.497  0.0744 -0.32666  0.0203 -0.14517 -0.05996
```

Numbers

- Eigenvalues and axis scores like in unconstrained ordination
- Eigenvalues should be lower than in unconstrained analysis, or constraints had no effect
- Components separately for constrained (explained) and unconstrained (residual) variation
- Four kind of scores
  1. Species scores derived from site (LC) scores
  2. Site scores which are linear components of constraints: **LC Scores**
  3. Site scores derived from species scores: **WA Scores**
  4. Scores for constraints: arrowheads for continuous variables (**biplot** scores) and centroids of factor levels
- Species–environment correlation: correlation between WA and LC scores

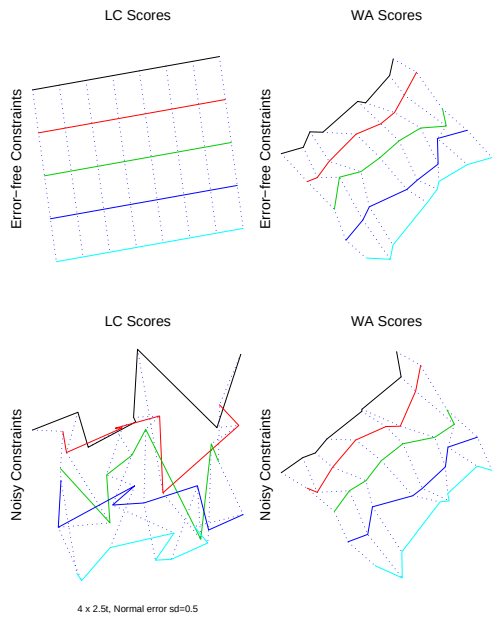
WA or LC Scores?

Mike Palmer:

- Use LC scores, because they give the best fit with the environment, and WA scores are a step from CCA towards CA.

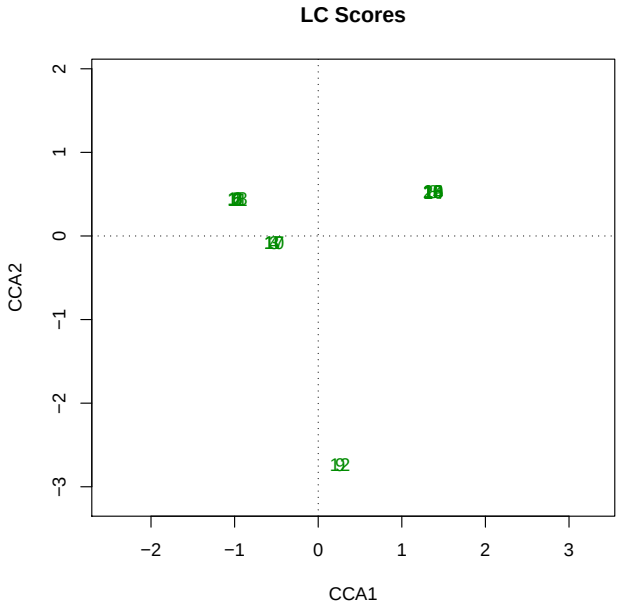
Bruce McCune:

- LC scores are excellent, if you have no error in constraining variables. Even with small error, LC scores become miserable, but WA scores are good even in noisy data.



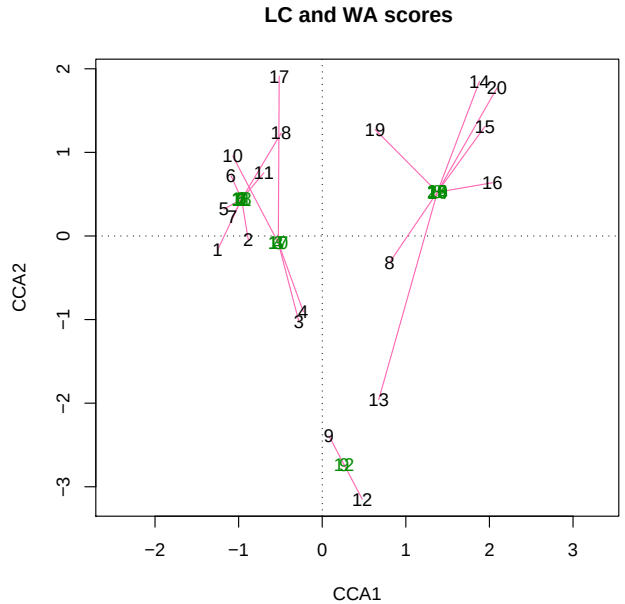
LC Scores are Constraints

Dune Meadows Constrained by Moisture Level



LC Scores are Constraints

Dune Meadows Constrained by Moisture Level



Outline

1. Constrained Ordination
   - Methods
   - Model Choice
   - Permutation Test
   - Partial Analysis
2. Analysis of Dissimilarities
   - Methods

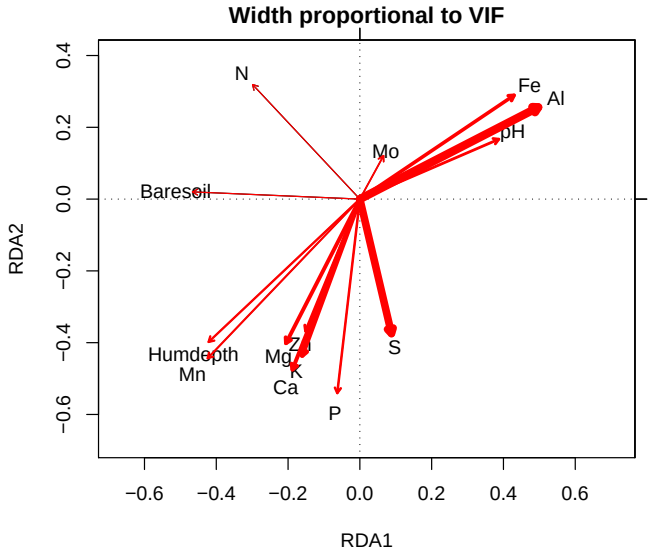
Model Choice

- Often people chunk in all evironmental variables they have — may be a patently bad idea
- Increasing the number of constraints means slacker constraint: analysis approaches unconstrained ordination and fitting environmental variables
- Does not allow hypothesis testing
- Many of the variables may be insignificant
- Multicollinearity between variables evident as *Variance Inflation*

```
> vif.cca(cca(varespec, varechem))
```

```
     N      P      K     Ca     Mg      S     Al
  1.98   6.03  12.01   9.93   9.81  18.38  21.19
    Fe     Mn     Zn     Mo Baresoil Humdepth     pH
  9.13   5.38   7.74   4.32   2.25    6.01   7.39
```


Variance Inflation Factor



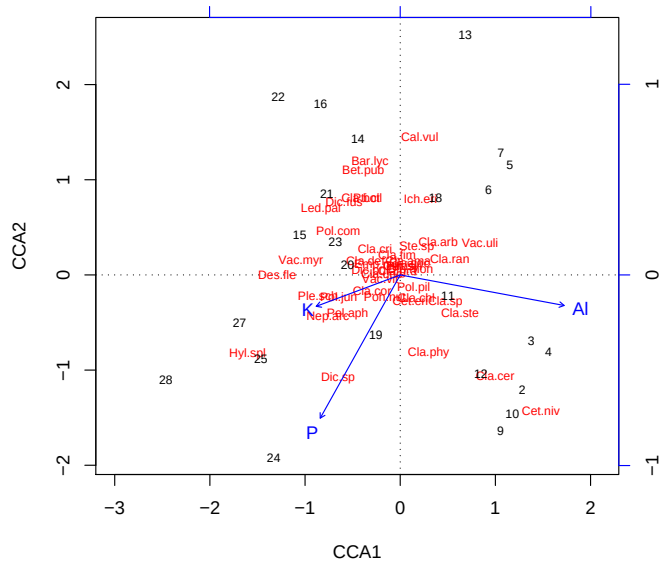
Model Specification: Formula Interface I

```
> (vare.cca <- cca(varespec ~ Al + P + K, varechem))

Call: cca(formula = varespec ~ Al + P + K, data =
varechem)

              Inertia Rank
Total           2.083
Constrained     0.644    3
Unconstrained   1.439   20
Inertia is mean squared contingency coefficient

Eigenvalues for constrained axes:
 CCA1  CCA2  CCA3
0.362 0.170 0.113

Eigenvalues for unconstrained axes:
   CA1    CA2    CA3    CA4    CA5    CA6    CA7    CA8
0.3500 0.2201 0.1851 0.1551 0.1351 0.1003 0.0773 0.0537
(Showed only 8 of all 20 unconstrained eigenvalues)
```

Model Specification: Formula Interface II

```
> vif.cca(vare.cca)

  Al    P    K
1.01 2.37 2.38
```


Plot



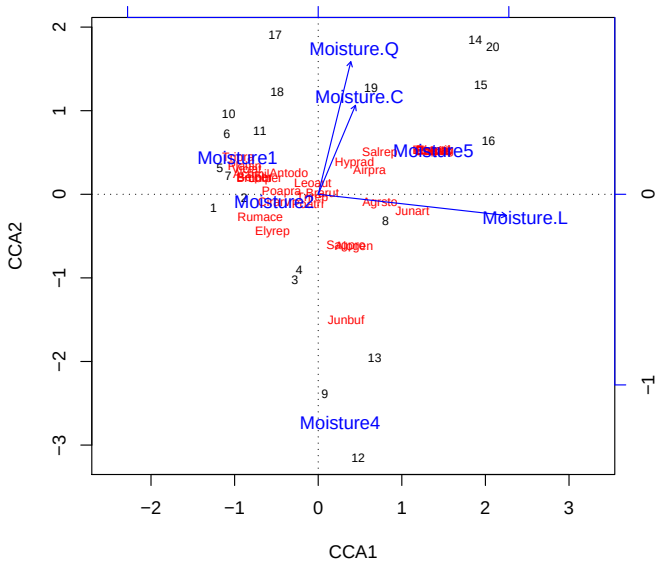
Coding Factors: Dummy variables

	ManagementHF	ManagementNM	ManagementSF
BF	0	0	0
SF	0	0	1
HF	1	0	0
NM	0	1	0

Ordered Factors

	Moisture.L	Moisture.Q	Moisture.C
1	-0.671	0.5	-0.224
2	-0.224	-0.5	0.671
4	0.224	-0.5	-0.671
5	0.671	0.5	0.224

Plotting Ordered Factors



Goodness of Model and its Costs

- Eigenvalue is the measure of goodness of fit
- Eigenvalue is maximized: even random constraints will have $\lambda > 0$
- AIC: balance maximization of eigenvalue by a penalty for each used constraint
- AIC does not exist for constrained ordination: AIC is based on Likelihood of the fitted model, and ordination models do not have Likelihood
- Toy-AIC may sometimes work, and can be used in automated model building
- Permutation tests can be used to check the approximate validity of automated model building

Shortcut to a Maximal Model I

```
> mod1 <- cca(varespec ~ ., varechem)
> mod1

Call: cca(formula = varespec ~ N + P + K + Ca + Mg + S
+ Al + Fe + Mn + Zn + Mo + Baresoil + Humdepth + pH,
data = varechem)

              Inertia Rank
Total           2.083
Constrained     1.441   14
Unconstrained   0.642    9
Inertia is mean squared contingency coefficient

Eigenvalues for constrained axes:
   CCA1    CCA2    CCA3    CCA4    CCA5    CCA6    CCA7
0.43887 0.29178 0.16285 0.14213 0.11795 0.08903 0.07029
   CCA8    CCA9   CCA10   CCA11   CCA12   CCA13   CCA14
0.05836 0.03114 0.01329 0.00836 0.00654 0.00616 0.00473
```


Shortcut to a Maximal Model II

```
Eigenvalues for unconstrained axes:
    CA1     CA2     CA3     CA4     CA5     CA6     CA7
0.19776 0.14193 0.10117 0.07079 0.05330 0.03330 0.01887
    CA8     CA9
0.01510 0.00949
```

Stepping to a Good Model I

```
> mod0 <- cca(varespec ~ 1, varechem)
> mod <- step(mod0, scope = formula(mod1), test = "perm",
+     perm.max = 100)

Start:  AIC=130
varespec ~ 1

           Df AIC    F N.Perm Pr(>F)
+ Al        1 129 3.67     99   0.01 **
+ Mn        1 129 3.31     99   0.01 **
+ Humdepth  1 129 3.01     99   0.01 **
+ Baresoil  1 130 2.46     99   0.02 *
+ Fe        1 130 2.44     99   0.01 **
+ P         1 130 2.19     99   0.05 *
+ Zn        1 130 1.93     99   0.04 *
<none>        130
+ Mg        1 130 1.87     99   0.10 .
+ K         1 130 1.86     99   0.06 .
+ Ca        1 130 1.80     99   0.10 .
```


Stepping to a Good Model II

```
+ pH        1 131 1.66     99   0.09 .
+ S         1 131 1.51     99   0.15
+ N         1 131 1.46     99   0.13
+ Mo        1 131 1.06     99   0.41
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Step:  AIC=129
varespec ~ Al

           Df AIC    F N.Perm Pr(>F)
+ P         1 128 2.50     99   0.01 **
+ K         1 128 2.32     99   0.01 **
+ S         1 128 2.16     99   0.02 *
+ Zn        1 128 1.99     99   0.06 .
+ Mn        1 128 1.89     99   0.03 *
<none>        129
+ Mg        1 129 1.74     99   0.08 .
+ N         1 129 1.59     99   0.09 .
```


Stepping to a Good Model III

```
+ Baresoil  1 129 1.57     99   0.10 .
+ Ca        1 129 1.42     99   0.08 .
+ Humdepth  1 129 1.38     99   0.24
+ Mo        1 130 0.99     99   0.51
+ pH        1 130 0.88     99   0.55
+ Fe        1 130 0.52     99   0.88
- Al        1 130 3.67     99   0.01 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Step:  AIC=128
varespec ~ Al + P

           Df AIC    F N.Perm Pr(>F)
+ K         1 127 2.17     99   0.07 .
<none>        128
+ Baresoil  1 128 1.66     99   0.08 .
+ N         1 128 1.55     99   0.09 .
+ S         1 128 1.34     99   0.22
```


Stepping to a Good Model IV

```
+ Mn        1 128 1.26     99   0.27
+ Zn        1 128 1.20     99   0.27
+ Humdepth  1 129 1.15     99   0.37
- P         1 129 2.50     99   0.01 **
+ Mo        1 129 0.98     99   0.61
+ Mg        1 129 0.96     99   0.48
+ pH        1 129 0.92     99   0.52
+ Fe        1 129 0.53     99   0.91
+ Ca        1 129 0.46     99   0.97
- Al        1 130 3.94     99   0.01 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Step:  AIC=127
varespec ~ Al + P + K

           Df AIC    F N.Perm Pr(>F)
<none>        127
+ N         1 128 1.51     99   0.14
```

Stepping to a Good Model V

```
+ Baresoil  1 128 1.45     99   0.10 .
+ Zn        1 128 1.31     99   0.25
+ S         1 128 1.26     99   0.25
- K         1 128 2.17     99   0.02 *
+ Mo        1 128 1.24     99   0.23
- P         1 128 2.34     99   0.02 *
+ Mg        1 128 1.03     99   0.43
+ Mn        1 128 0.89     99   0.65
+ Humdepth  1 128 0.81     99   0.57
+ Fe        1 129 0.52     99   0.84
+ pH        1 129 0.51     99   0.81
+ Ca        1 129 0.44     99   0.88
- Al        1 130 4.33     99   0.01 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

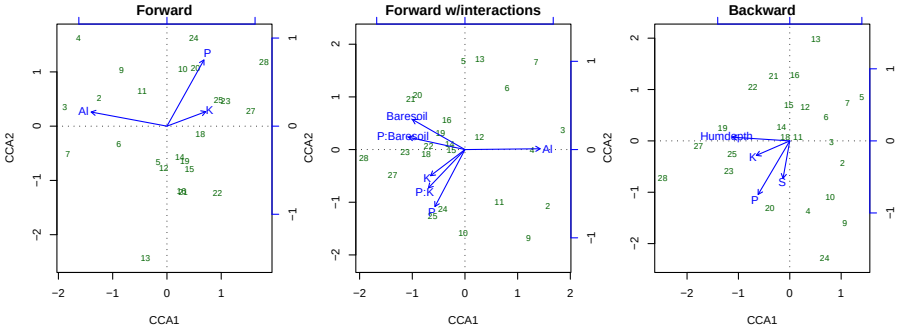
> mod

Stepping to a Good Model VI

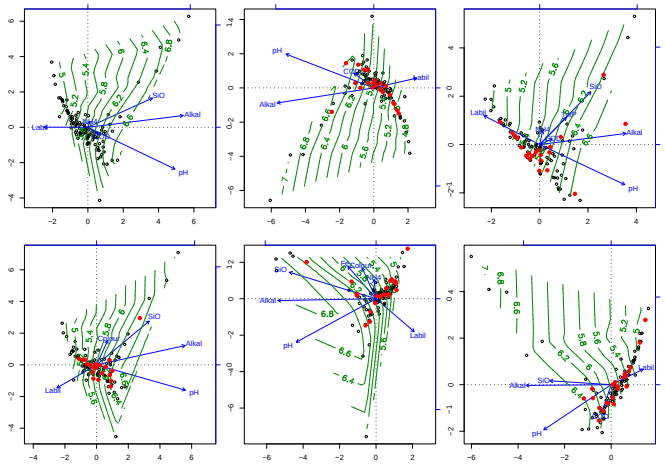
```
Call: cca(formula = varespec ~ Al + P + K, data =
varechem)

              Inertia Rank
Total           2.083
Constrained     0.644    3
Unconstrained   1.439   20
Inertia is mean squared contingency coefficient

Eigenvalues for constrained axes:
 CCA1  CCA2  CCA3
0.362 0.170 0.113

Eigenvalues for unconstrained axes:
   CA1    CA2    CA3    CA4    CA5    CA6    CA7    CA8
0.3500 0.2201 0.1851 0.1551 0.1351 0.1003 0.0773 0.0537
(Showed only 8 of all 20 unconstrained eigenvalues)
```


Stepping is Dangerous

Automatic model selection may give different results depending on stepping direction, scope or small changes in the data set



5-fold Cross-validation and stepping



Outline

1. Constrained Ordination
   - Methods
   - Model Choice
   - Permutation Test
   - Partial Analysis
2. Analysis of Dissimilarities
   - Methods

Permutation Test

- The signficance of constraints cannot be directly evaluated, but we can use permutation tests
- Shuffle community data into random order and refit the model: gives goodness of fit of a random model
- If observed goodness of fit is better than (most) random models, then the constraints are significant
- The criterion of the goodness of fit is pseudo-F:

$$F = \frac{\Lambda_c/p}{\Lambda_r/(n-p-1)},$$

  where Λ_c and Λ_r are constrained and residual inertia, p is the rank of constraints, and n is the number of observations
- Definition similar to F-statitistic in ANOVA, but does not follow its distribution (except for single variable in RDA)

Overall Test

```
> anova(mod)

Permutation test for cca under reduced model

Model: cca(formula = varespec ~ Al + P + K, data = varechem)
         Df Chisq    F N.Perm Pr(>F)
Model     3  0.64 2.98    199  0.005 **
Residual 20  1.44
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```


ANOVA by Terms

```
> anova(mod, by = "terms")

Permutation test for cca under reduced model
Terms added sequentially (first to last)

Model: cca(formula = varespec ~ Al + P + K, data = varechem)
         Df Chisq    F N.Perm Pr(>F)
Al        1  0.30 4.14     99   0.01 **
P         1  0.19 2.64     99   0.03 *
K         1  0.16 2.17     99   0.04 *
Residual 20  1.44
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

ANOVA by Margins

Type III Sums of Squares

```
> anova(mod, by = "mar")

Permutation test for cca under reduced model
Marginal effects of terms

Model: cca(formula = varespec ~ Al + P + K, data = varechem)
         Df Chisq    F N.Perm Pr(>F)
Al        1  0.31 4.33    199  0.005 **
P         1  0.17 2.34    199  0.015 *
K         1  0.16 2.17    699  0.030 *
Residual 20  1.44
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

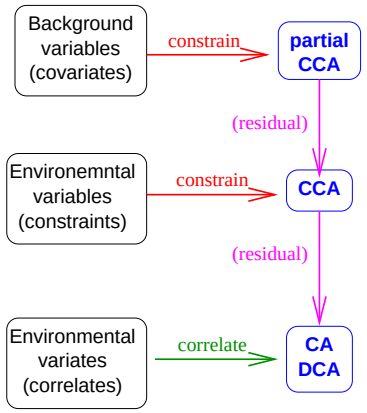

ANOVA by Axis

```
> anova(vare.cca, by = "axis", perm = 1000)

Permutation test for cca under reduced model

Model: cca(formula = dune ~ Moisture, data = dune.env)
         Df Chisq    F N.Perm Pr(>F)
CCA1      1  0.42 4.51    199  0.005 **
CCA2      1  0.13 1.43     99  0.280
CCA3      1  0.08 0.82     99  0.570
Residual 16  1.49
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```


Outline

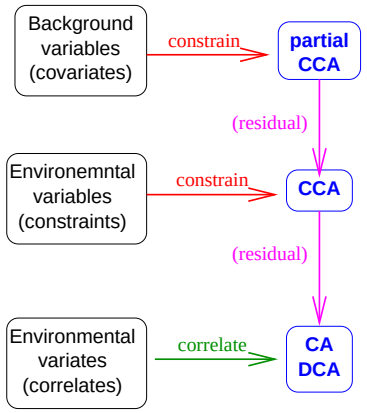
1. Constrained Ordination
   - Methods
   - Model Choice
   - Permutation Test
   - Partial Analysis
2. Analysis of Dissimilarities
   - Methods

Levels of Intervention



- **Partial CCA** removes the effect of background variables before proper (C)CA: ‘random’ or ‘nuisance’ variables.
- Residual ordinations: Partitioning of variation.

Levels of Intervention



- **Partial CCA** removes the effect of background variables before proper (C)CA: 'random' or 'nuisance' variables.
- Residual ordinations: Partitioning of variation.
- Constraints are linear: Non-orthogonal environmental variables may give 'negative components of variation'
- Information of lower levels mixed with upper.

Why Partial Ordination?

- Remove the effect of background (or "random") variables before analysing the effect of interesting variables
- Allows analysis of experimental design (constraints) with confounding variables (conditions)
- Allows split-plot and other hierarchical designs
- Decomposition of variation due to different sources, like spatial and environmental components

Treatment with Confounding Natural Variation I

```
> (ord <- rda(dune ~ Management + Condition(A1 + Moisture),
+     dune.env))

Call: rda(formula = dune ~ Management + Condition(A1 +
Moisture), data = dune.env)

              Inertia Rank
Total            84.1
Conditional      29.8    4
Constrained      19.1    3
Unconstrained    35.2   12
Inertia is variance

Eigenvalues for constrained axes:
 RDA1  RDA2  RDA3
11.26  4.88  2.97

Eigenvalues for unconstrained axes:
  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10
```


Treatment with Confounding Natural Variation II

```
8.211 7.138 4.613 4.027 3.023 2.659 1.867 1.499 0.910 0.641
 PC11  PC12
0.387 0.268

> anova(ord)

Permutation test for rda under reduced model

Model: rda(formula = dune ~ Management + Condition(A1 + Moisture), data = dun
         Df   Var    F N.Perm Pr(>F)
Model     3  19.1 2.17  199.0  0.005 **
Residual 12  35.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What Actually is Permuted in Tests?

- **Direct Model**: Always permutes community data
- **Reduced Model**: Permutes community data in non-partial models, and residuals after conditions in partial model
- **Full Model**: Permutes residuals after constraints and possible conditions
- When residuals are permuted in reduced and full model, the permuted residuals are added to the unpermuted fitted values
- Theory assume that residuals are *exchangeable*, and hypothesis of randomness concern residuals
- Assumes *independent and identically distributed* residuals: these can be added to fitted values

Components of Variation

- There can be several groups of source of variation, and we may be interested in quantifying these components
- Typical example: decomposition of variation into pure spatial, pure environmental and spatially structured environmental variation
- We expect that usual $R^2 > 0$, because the goodness of fit is maximized, but adjusted R^2 takes into account the number of constraints and has expectation 0 with random constraints
- Spatial structure can be described by Principal Components of Neighbourhood Matrix (PCNM)

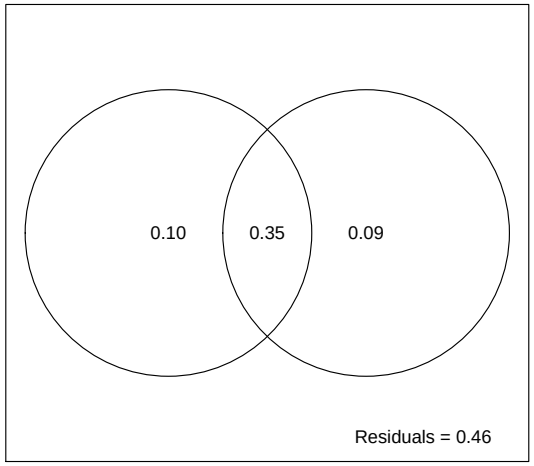
Example: Spatial and Environmental Variation I

```
> (mod <- varpart(mite, mite.pcnm, ~., data = mite.env,
+     transfo = "hellinger"))

Partition of variation in RDA

Call:
varpart(Y = mite, X = mite.pcnm, ~., data = mite.env, transfo = "hellinger")
Species transformation:  hellinger
Explanatory tables:
X1:  mite.pcnm
X2:  ~.

No. of explanatory tables: 2
Total variation (SS): 27.205
            Variance: 0.39428
No. of observations: 70

Partition table:
                    Df R.squared Adj.R.squared Testable
```


Example: Spatial and Environmental Variation II

```
[a+b] = X1            22    0.62300       0.44653     TRUE
[b+c] = X2            11    0.52650       0.43670     TRUE
[a+b+c] = X1+X2       33    0.75893       0.53794     TRUE
Individual fractions
[a] = X1|X2           22                  0.10124     TRUE
[b]                    0                  0.34530    FALSE
[c] = X2|X1           11                  0.09141     TRUE
[d] = Residuals                           0.46206    FALSE
---
Use function 'rda' to test significance of fractions of interest
```

Components of Variance



Outline

1 Constrained Ordination
- Methods
- Model Choice
- Permutation Test
- Partial Analysis

2 Analysis of Dissimilarities
- Methods

Direct Analysis of Dissimilarities

- Analyse dissimilarities instead of mapping them into reduced number of dimensions of ordination
- Distance-based Redundancy Analysis (`capscale` in vegan) can perform the reduction
- Want to have non-Euclidean metric?
- Want to study the effect of geographic (spatial) distance?
- Do you have huge number of variables, but a modest number of observations (like in genetic data)

Distance-based RDA I

```
> pcnmmat <- as.matrix(mite.pcnm)
> (ord <- capscale(vegdist(mite) ~ . + pcnmmat, mite.env))

Call: capscale(formula = vegdist(mite) ~ SubsDens +
WatrCont + Substrate + Shrub + Topo + pcnmmat, data =
mite.env)

              Inertia Rank
Total            1155
Constrained       821   33
Unconstrained     334   36
Inertia is squared Bray distance

Eigenvalues for constrained axes:
    CAP1     CAP2     CAP3     CAP4     CAP5     CAP6     CAP7
361.3613 100.9934  77.2955  51.5555  38.6290  30.5674  24.6378
    CAP8     CAP9    CAP10    CAP11    CAP12    CAP13    CAP14
 18.0338  17.4683  15.7685  13.0950  12.2059   9.4126   8.2788
   CAP15    CAP16    CAP17    CAP18    CAP19    CAP20    CAP21
```

Distance-based RDA II

```
  6.5805   6.2683   5.0483   4.6042   3.5926   3.2217   3.0243
   CAP22    CAP23    CAP24    CAP25    CAP26    CAP27    CAP28
  2.1762   1.9741   1.3669   1.1282   0.8464   0.7629   0.3836
   CAP29    CAP30    CAP31    CAP32    CAP33
  0.3671   0.2670   0.2117   0.1191   0.0232

Eigenvalues for unconstrained axes:
MDS1 MDS2 MDS3 MDS4 MDS5 MDS6 MDS7 MDS8
73.4 41.2 25.7 24.4 22.6 20.0 18.9 14.0
(Showed only 8 of all 36 unconstrained eigenvalues)
```

```
> anova(ord, by = "margin", perm.max = 1000)
```


Distance-based RDA III

```
Permutation test for capscale under reduced model
Marginal effects of terms

Model: capscale(formula = vegdist(mite) ~ SubsDens + WatrCont + Substrate + Sh
          Df Var    F N.Perm Pr(>F)
SubsDens   1   9 0.97     99  0.490
WatrCont   1  21 2.24    299  0.020 *
Substrate  6  76 1.36    999  0.034 *
Shrub      2  11 0.58     99  0.950
Topo       1  13 1.35     99  0.180
pcnmmat   22 291 1.43    199  0.005 **
Residual  36 334
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```


Mantel and Partial Mantel Tests

- Mantel correlation (a.k.a. matrix correlation) is the correlation between two sets of dissimilarities or distances
- $n(n-1)/2$ dissimilarities among n points: ordinary statistical tests of correlations do not apply
- Significance can be assessed by permutation, or asymptotically using equations of Mantel
- Partial Mantel test: use partial correlations
- Analagous to conditioned db-RDA: partial out variation by background distances
- Residuals of distances are not equivalent to residuals of raw data: decomposition of variation dubious

Example: Community Structure and Environment

```
> library(cluster)
> envdis <- daisy(mite.env)
> mantel(vegdist(mite), envdis)

Mantel statistic based on Pearson's product-moment correlation

Call:
mantel(xdis = vegdist(mite), ydis = envdis)

Mantel statistic r: 0.422
      Significance: <0.001

Empirical upper confidence limits of r:
   90%    95%  97.5%    99%
0.0355 0.0473 0.0590 0.0714

Based on 1000 permutations
```

Controlling for Spatial Distance

```
> mantel.partial(vegdist(mite), envdis, dist(mite.xy))

Partial Mantel statistic based on Pearson's product-moment correlation

Call:
mantel.partial(xdis = vegdist(mite), ydis = envdis, zdis = dist(mite.xy))

Mantel statistic r: 0.292
      Significance: <0.001

Empirical upper confidence limits of r:
   90%    95%  97.5%    99%
0.0437 0.0567 0.0766 0.0983

Based on 1000 permutations
```


Linear Analysis of Dissimilarities

- Function `adonis` in vegan
- Permutational MANOVA or non-parametric MANOVA
- Uses "outer products" in MANOVA instead of usual "inner products": dissimilarities among points instead of distances of variables to their centroids
- Practical if the number of variables is huge: related to AMOVA of gene expression data
- With Euclidean distances equal to MANOVA, but uses permutation tests
- Can be used with any adequate dissimilarity measure
- Test sequential: order of variables does matter

Example: Environment after Spatial Variation

```
> adonis(vegdist(mite) ~ pcnmmat + ., mite.env, perm = 500)

Call:
adonis(formula = vegdist(mite) ~ pcnmmat + ., data = mite.env,      permutations

              Df SumsOfSqs MeanSqs F.Model   R2 Pr(>F)
pcnmmat   22.0000    8.8390  0.4018  3.8794 0.60 <0.002 ***
SubsDens   1.0000    0.4102  0.4102  3.9605 0.03  0.006 **
WatrCont   1.0000    0.3239  0.3239  3.1278 0.02  0.014 *
Substrate  6.0000    1.0719  0.1787  1.7250 0.07  0.022 *
Shrub      2.0000    0.1592  0.0796  0.7687 0.01  0.690
Topo       1.0000    0.1637  0.1637  1.5806 0.01  0.138
Residuals 36.0000    3.7284  0.1036         0.25
Total     69.0000   14.6963                 1.00
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Other Dissimilarity-based Methods

- MRPP (Multi-Response Permutation Procedure) and ANOSIM (Analysis of Dissimilarities) compare differences among groups
  - Both are sensitive to differences in the dispersions within groups: **not recommended**
- Multivariate analysis of homogeneity (`betadisper` in vegan)
  - With Euclidean distances equal to Levene's test on the homegeneity of variances
  - Also works exactly on non-Euclidean dissimilarities
  - Can be used to study beta diversity within groups
  - Either parametric ANOVA or permutation tests available
  - Pairwise *post hoc* comparison available (Tukey)
  - PERMDISP2 by another name