bcool

Hugo Naya and Lucía Spangenberg

16 de enero de 2015

Índice

1. Introduction

The method presented here identifies associations between amino acid changes in "interesting"positions in an alignment (taking into account several amino acid properties) with some meta information (e. g. phenotypic data). The proposed method has general applicability to other organisms, different amino acid properties and different meta data. As a motivating example, we applied it to a set of 209 bacterial strains belonging to several genera (72 genera, 117 species) with the aim of finding amino acid changes that might be correlated with the pathogenicity of the bacteria. Several studies have shown that the pathogenicity character of different bacterial strains is determined by changes in amino acids causing changes in protein structure, and hence function (Sokurenko et al. (1998); Conenello et al. (2007); Marjuki et al. (2010)). Thus, the pathogenicity character can also be conferred by specific genetic variations having an effect on protein function and not solely by the presence or absence of virulence factor genes as previously assumed (Falkow (1997)). The proposed screening method identify interesting sites in an alignment (which might confer the pathogenicity character to some bacteria) through the application of linear mixed models on different amino acid properties in each of those columns. Amino acid properties can be grouped according to an enourmous number of different characteristics, such as size, polarity, alpha helix or beta sheet propensity. However, a big number of different properties are highly correlated (Kawashima et al., 2008), clustering in only six groups: α and turn propensities (A), β propensity (B), composition (C), hydrophobicity (H), physicochemical (P), and other properties (O) (Tomii and Kanehisa, 1996). Substitutions severely changing the value of some key properties (e. g. from polar to non-polar) tend to have a stronger effect on the tertiary structure, and probably in the function of the protein. If those substitutions are associated with the label (meta data) they define an interesting alignment column, which might be responsible for pathogenicity. The amino acid properties considered depend on the specific problem as will be discussed later.
Assuming that after a first fast filtering (to reduce the computational time) we keep only the columns of potential interest, we will apply a linear mixed model on each of these columns. The phylogenetic mixed model of Lynch (PMM) partitions each phenotypic values into three components:

$$\vec{Y} = \mathbf{X}\vec{\beta} + \mathbf{Z}\vec{a} + \vec{e}, \tag{1}$$

where $\vec{Y}$ is the vector of observations (the dependent variable), $\vec{\beta}$ is the vector of fixed effects, $\vec{a}$ is the vector of phylogenetic heritable additive effects and $\vec{e}$ is the vector of independent and identically distributed residual errors. $\mathbf{X}$ is the incidence matrix that associates effects with observations. The number of columns of $\mathbf{X}$ are the number of fixed effect levels one wants to consider. $\mathbf{Z}$ is the matrix that associates additive effects with observations. Both of them, $\mathbf{X}$ and $\mathbf{Z}$, are matrices relating the observations $\vec{Y}$ to regressors the $\vec{\beta}$ and $\vec{a}$.
Equation (1) is applicable to very general cases, especially $\vec{\beta}$ could be a vector holding the regressors for many different fixed effects and several link functions can be used for Y, extending the theory to generalized linear mixed models. In the case of a binary labeling (such as pathogen/non-pathogen), $\mathbf{X}$ contains the labels of the organisms, hence it has a dimension of $n \times 2$ (n corresponds to the number of organisms considered). Vector $\vec{X}_{i1}$ corresponds to the pathogens and it holds $x_{i1} = 1$ for pathogens, and $x_{i1} = 0$ for non-pathogens. Vector $\vec{X}_{i2}$ stands for non-pathogens and it holds $x_{i2} = 0$ for pathogens, and $x_{i2} = 1$ for non-pathogens. Each y_i is the value of the amino acid property considered in the organism i. $\mathbf{Z}$ is the matrix relating species to observations and in our case corresponds to a diagonal matrix of dimensions

$n \times n$. Random effects are normally distributed with mean 0 and variance matrices $\mathbf{R}$ and $\mathbf{G}$, corresponding to residual and additive effects, respectively. In the univariate case $\mathbf{R} = \mathbf{I_n} \times \sigma_e^2$ and $\mathbf{G} = \mathbf{A} \times \sigma_a^2$, σ_e^2 and σ_a^2 standing for residual and additive variances. The $\mathbf{A}$ matrix represents the phylogenetic relations between the n organisms. It holds evolutionary "time"values t_{ij} representing the time that organism i shared with organism j before speciation. The a_i and e_i values are the random organism effects and the error term for each organism, respectively. These two vectors, and the fixed effects, $\vec{\beta}$, are the ones to be estimated. $\vec{\beta}$ has dimension 2 (β_p: pathogen, β_{np}: non-pathogen) in our binary case, since we are calculating the fixed effects of the pathogenicity. In this work, a Bayesian approach similar to the one presented by Naya *et al.* (2006) is chosen, hence not just a single value for the difference between $\vec{\beta}_p$ and $\vec{\beta}_{np}$ is determined, but a posterior probability distribution. Our package makes extensive usage of the main function implemented in the MCMCglmm package (Hadfield (2010)) and we strongly encourage users to read the corresponding documentation.

2. An example dataset

This document briefly describes an introduction to the usage of the **bcool** package, basically analyzing the same information that used Spangenberg et al. (2011), which is available in the package (rpoS").

```
> library("bcool")
> data("rpoS")
> env <- new.env()
> utils::data("aaindex", package = "seqinr", envir = env)
> aaindex <- env$aaindex
>
>
```

In Spangenberg et al. (2011) an RpoS (σ^{38}) alignment was scanned searching for relevant columns, sites probably associated with pathogenicity. The labels (pathogenicity "YES" or "No") were obtained from the NCBI. The phylogenetic tree was reconstructed from the concatenated alignment of 7 groups of orthologous genes obtained from KEGG (K03070, K03073, K03076, K03087, K030106, K030110, K03217). The dataset includes a table with the pathogenicity labels ("labels"), the phylogenetic tree ("tree7") and the alignment (rpoSalign") of the RpoS protein.

3. Analysis of the dataset

Before initiate our analysis we need to define which amino acid properties are relevant for us. That is, which properties we consider that could provoke important changes in the function of the protein. While the properties to be used only depend on the users knowledge, a good starting point would be to consider out of the set of 500 properties included in the "aaindex", one representative of each of the six groups mentioned above (α and turn propensities, β propensity, composition, hydrophobicity, physicochemical, and other properties). As we only wish

to demonstrate the basical usage of the package and the time involved is directly proportional to the number of selected properties we will use only two properties here.

```
> options(width=50)
> prop<-c("CHOC760102","KYTJ820101")
> aaindex[prop]
```

```
$CHOC760102
$CHOC760102$H
[1] "CHOC760102"

$CHOC760102$D
[1] "Residue accessible surface area in folded protein (Chothia, 1976)"

$CHOC760102$R
[1] "LIT:2004094b PMID:994183"

$CHOC760102$A
[1] "Chothia, C."

$CHOC760102$T
[1] "The nature of the accessible and buried surfaces in proteins"

$CHOC760102$J
[1] "J. Mol. Biol. 105, 1-14 (1976)"

$CHOC760102$C
[1] "JANJ780101    0.973  GUYH850104    0.970  JANJ780103    0.959GUYH850105    0.946

$CHOC760102$I
Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys
 25  90  63  50  19  71  49  23  43  18  23  97
Met Phe Pro Ser Thr Trp Tyr Val
 31  24  50  44  47  32  60  18


$KYTJ820101
$KYTJ820101$H
[1] "KYTJ820101"

$KYTJ820101$D
[1] "Hydropathy index (Kyte-Doolittle, 1982)"

$KYTJ820101$R
[1] "LIT:0807099 PMID:7108955"

$KYTJ820101$A
```

```
[1] "Kyte, J. and Doolittle, R.F."

$KYTJ820101$T
[1] "A simple method for displaying the hydropathic character of a protein"

$KYTJ820101$J
[1] "J. Mol. Biol. 157, 105-132 (1982)"

$KYTJ820101$C
[1] "JURD980101    0.996  CHOC760103    0.964  OLSK800101    0.942JANJ780102    0.922

$KYTJ820101$I
 Ala  Arg  Asn  Asp  Cys  Gln  Glu  Gly  His  Ile
 1.8 -4.5 -3.5 -3.5  2.5 -3.5 -3.5 -0.4 -3.2  4.5
 Leu  Lys  Met  Phe  Pro  Ser  Thr  Trp  Tyr  Val
 3.8 -3.9  1.9  2.8 -1.6 -0.8 -0.7 -0.9 -1.3  4.2
```

3.1. Instantiation of the APPT.list object

We have now all the elements we require to instantiate the object of the main class `APPT.list` (Alignment, Phenotype, Properties, Tree):

```
> myAPPT.list<-new("APPT.list",alignment=rpoSalign,pheno=labels,
 properties=lapply(aaindex[prop],function(x) x$I),tree=tree7)
> head(columns(myAPPT.list))
[1] 1 2 3 4 5 6
> head(pheno(myAPPT.list))
  spKEGG
1    aae
2    aeh
3    afe
4    afr
5    alv
6    amc
                                  organism
1                          Aquifex aeolicus
2                 Alkalilimnicola ehrlichei
3 Acidithiobacillus ferrooxidans ATCC 53993
4 Acidithiobacillus ferrooxidans ATCC 23270
5                    Allochromatium vinosum
6                     Alteromonas macleodii
  pathogenicity
1            No
2            No
```

```
3            No
4            No
5            No
6            No
```

```
> properties(myAPPT.list)
```

```
$CHOC760102
Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys
 25  90  63  50  19  71  49  23  43  18  23  97
Met Phe Pro Ser Thr Trp Tyr Val
 31  24  50  44  47  32  60  18

$KYTJ820101
 Ala  Arg  Asn  Asp  Cys  Gln  Glu  Gly  His  Ile
 1.8 -4.5 -3.5 -3.5  2.5 -3.5 -3.5 -0.4 -3.2  4.5
 Leu  Lys  Met  Phe  Pro  Ser  Thr  Trp  Tyr  Val
 3.8 -3.9  1.9  2.8 -1.6 -0.8 -0.7 -0.9 -1.3  4.2
```

```
> tree(myAPPT.list)
```

```
Phylogenetic tree with 209 tips and 208 internal nodes.

Tip labels:
        ect, ecp, ecq, ecz, eci, ecc, ...

Rooted; includes branch lengths.
```

```
>
```

3.2. Fast approaches

IMPORTANT: the alignment should be a matrix of one letter code. The properties are grouped in a list of vectors, each with 20 values, which names are the amino acids in three letter codes. You can use **seqinr** package to convert between formats if needed.

Our object contains now the essential data to start with the analysis. Usually the complete analysis takes a while, then we can try to remove noninformative columns. In the paper of Spangenberg et al. (2011) they tried three different fast approaches to select columns. Unfortunately, these approaches were unsucessful. However, it is difficult to say that this occurs in general and that it is independent from data idiosyncracies. For this reason we implemented two of the three methods here (Conditional Entropy Reduction, CER and ANOVA) while the third is straightforward to implement (Entropy). Note that we can explicit two obvious restrictions that reduce the number of columns to analyze: the maximum number of gaps that we allow and the minimum number of different aminoacids in each column (below 2 has no sense).

```
> my.cer<-cer.APPT.list(myAPPT.list,class.var="pathogenicity",
 which.columns=NULL,nummin=2,maxngaps=10)
> head(sort(my.cer))
```

```
      509       322       295       549       508
0.6434340 0.8088848 1.0914037 1.1435014 1.3091885
      273
1.7166933

> tail(sort(my.cer))

     287      304      428      277      340
17.64696 19.26519 19.77923 20.88705 22.16580
     281
25.75515

> my.anova<-anovaAPPT.list(myAPPT.list,class.var=~pathogenicity,
 which.columns=NULL,nummin=2,maxngaps=10)
> sum(my.anova$hm.signif==4,na.rm=TRUE)

[1] 0

> which(my.anova$hm.signif==4)

integer(0)

> head(my.anova)

     CHOC760102   KYTJ820101 hm.signif    median
95  0.013004205 1.910214e-04         1 2.8024170
96  0.002077156 1.455420e-01         1 1.7597712
97  0.774784231 5.850704e-02         0 0.6718055
98  0.112565016 2.440715e-07         1 3.7805397
99  0.038802670 2.352290e-02         0 1.5198238
126 0.004935960 5.641298e-02         1 1.7776247
         mean
95  2.8024170
96  1.7597712
97  0.6718055
98  3.7805397
99  1.5198238
126 1.7776247

>
```

3.3. Setting the model

As the computation time is proportional to the number of columns our example will be run in only 3 columns (arbitrarily chosen):

```
> colu<-c(374:376)
> # define the priors and run the model
> # here the number of processors is one (count=1).
> # increase this number if possible
> prior<-list(list(R=list(V=40, nu=1), G=list(G1=list(V=40, nu=1))),
```

```
 list(R=list(V=3, nu=1), G=list(G1=list(V=3, nu=1))))
> myAPPT.list<-MCMCglmm.APPT.list(myAPPT.list, ~ -1+pathogenicity,
         random.eff="spKEGG",nitt=1.5e3,burnin=5e2,prior,scale=FALSE,
         parallel=TRUE,which.columns=colu,maxngaps=10,nummin=2,
         count=3,pr=FALSE)

        3 slaves are spawned successfully. 0 failed.
        3 slaves are spawned successfully. 0 failed.

>
>
```

Now

3.4. Monitoring convergence

After running the PMM we want to know in which properties and sites the model converged. For this task we can simply realize a Geweke diagnostic test for each site and property. The values obtained are the Z-scores for a test of equality of means between the first and last parts of the chain.

```
> matGwk<-matrix(0,length(columns(myAPPT.list)),
 length(properties(myAPPT.list)))
> for (i in 1:length(properties(myAPPT.list))){
         for (j in 1:length(columns(myAPPT.list))){
                 matGwk[j,i]<-geweke.diag(as.mcmc(
                 multiMCMCglmm(myAPPT.list)[[i]][[j]]$Sol[,2]
                 -multiMCMCglmm(myAPPT.list)[[i]][[j]]$Sol[,1]
                 ))$z
         }
 }
> colnames(matGwk)<-names(properties(myAPPT.list))
> rownames(matGwk)<-columns(myAPPT.list)
> matGwk

    CHOC760102 KYTJ820101
374  0.7670528  1.3731115
375 -4.9478687 -0.4026659
376  1.3568568 -1.0811796

> which(abs(matGwk)>2,TRUE)

    row col
375   2   1

>
```

3.5. Summarizing the results

We finally arrived at the point where we can summarize the results. The method `levelsMCMCglmm` report all levels available for contrast in the model. The output of the method `summaryAPPT.list` is a list with 6 elements. The first is a table with the gt0 values transformed via 2*(gt0-0.5). The second element is the table with the median effect sizes. The third report the summary statistic S_{T_j}. The fourth is the χ^2_m statistic, calculated from the number of samples taken from the MCMC sampling, with m corresponding to the number of properties. The fifth is the χ^2_m statistic but now calculated from the effective sample size from each MCMC run. The last element report a table of amino acid frequencies for each site.

```
> levelsMCMCglmm(myAPPT.list)

[1] "pathogenicityNo"  "pathogenicityYes"

> my.summary<-summaryAPPT.list(myAPPT.list,
 contrast=c("pathogenicityYes","pathogenicityNo"),
 class.var="pathogenicity",what.prop=NULL)
> attributes(my.summary)

$names
[1] "tabcor"    "tabSize"   "SumTr"
[4] "ChiSq"     "ChiSq.eff" "AAlist"

> my.summary$tabcor

    CHOC760102 KYTJ820101
374      -0.42       0.68
375      -0.66       0.12
376       0.02      -0.74

> my.summary$tabSize

       CHOC760102   KYTJ820101
374 -0.0104176605  0.023066887
375 -0.0183836048  0.002633052
376  0.0005554163 -0.012892218

> my.summary$SumTr

   374    375    376
0.3194 0.2250 0.2740

> my.summary$ChiSq

  374   375   376
63.88 45.00 54.80

> my.summary$ChiSq.eff

     374      375      376
1.879009 6.577161 1.172929

> my.summary$AAlist
```

```
$374

               A          D          E          H
  No  0.07368421 0.05263158 0.37894737 0.05263158
  Yes 0.14912281 0.00000000 0.53508772 0.06140351

               K          L          M          N
  No  0.08421053 0.03157895 0.01052632 0.06315789
  Yes 0.01754386 0.03508772 0.00000000 0.05263158

               Q          R          S          T
  No  0.05263158 0.07368421 0.03157895 0.06315789
  Yes 0.01754386 0.00000000 0.00000000 0.00000000

               Y
  No  0.03157895
  Yes 0.13157895

$375

               A          C          E          F
  No  0.08421053 0.01052632 0.28421053 0.02105263
  Yes 0.00000000 0.00000000 0.20175439 0.00000000

               H          K          L          M
  No  0.05263158 0.12631579 0.25263158 0.02105263
  Yes 0.00000000 0.02631579 0.23684211 0.00877193

               Q          R          T          V
  No  0.04210526 0.03157895 0.00000000 0.02105263
  Yes 0.46491228 0.03508772 0.02631579 0.00000000

               Y
  No  0.05263158
  Yes 0.00000000

$376

               A          C          F          I
  No  0.01052632 0.01052632 0.00000000 0.02105263
  Yes 0.00000000 0.00000000 0.02631579 0.04385965

               L          M          S          T
  No  0.75789474 0.01052632 0.08421053 0.04210526
  Yes 0.58771930 0.01754386 0.16666667 0.13157895

               V
```

```
  No  0.06315789
  Yes 0.02631579
>
```

Relevant columns can be selected simply ranking the columns by S_{T_j}, that is, by the statistical significance of the aggregated properties, or alternatively by aggregating effect sizes via the l^2 norm (see Spangenberg et al. 2011).

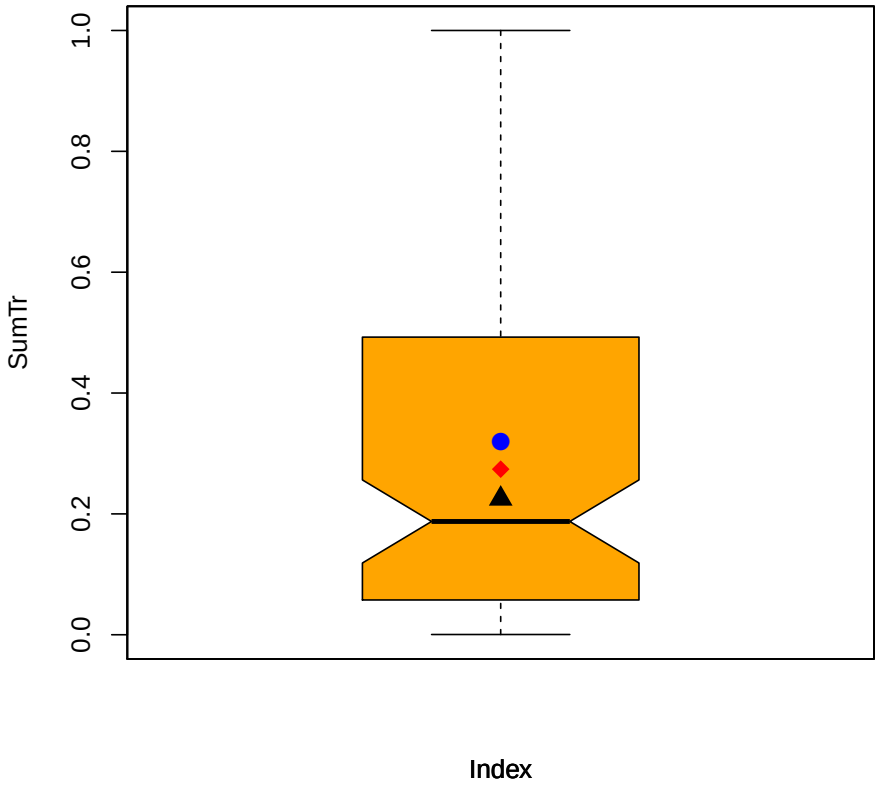
3.6. Bootstrapping relevant columns

To check the relevance of the scores obtained for the selected columns we can perform a bootstrap in few of them (one each time). The distributions of bootstrap scores obtained is very similar for an ample range of S_{T_j} scores and then you only need to do the bootstrap in few scores (for details see Spangenberg et al. (2011)).

```
> myAPPT.list.boot<-bootMCMCglmm.APPT.list(
        myAPPT.list, ~ -1+pathogenicity,boot=100,
        contrast=c("pathogenicityYes","pathogenicityNo"),
        random.eff="spKEGG",nitt=1.5e4,burnin=5e3,prior,
        scale=FALSE,parallel=TRUE,what.prop=c(1,2),
        which.column=c(376),maxngaps=10,nummin=2,count=3)

       3 slaves are spawned successfully. 0 failed.
       3 slaves are spawned successfully. 0 failed.

> boxplot(myAPPT.list.boot$SumTr,col="orange",notch=TRUE,ylim=c(0,1))
> par(new=TRUE)
> plot(my.summary$SumTr["374"],ylim=c(0,1),pch=16,col="blue",xaxt="n",
 yaxt="n",cex=1.5,ylab="")
> par(new=TRUE)
> plot(my.summary$SumTr["375"],ylim=c(0,1),pch=17,col="black",xaxt="n",
 yaxt="n",cex=1.5,ylab="")
> par(new=TRUE)
> plot(my.summary$SumTr["376"],ylim=c(0,1),pch=18,col="red",xaxt="n",
 yaxt="n",cex=1.5,ylab="SumTr")
>
```



3.7. Estimating heritabilities

Finally, as usual in the context of Bayesian MCMC, it is very easy to calculate heritability for each site and property. For the meaning of these estimates in the comparative method context see Lynch (1991), Naya et al. (2006) and Hadfield (2010).

```
> matH2<-matrix(0,length(columns(myAPPT.list)),
 length(properties(myAPPT.list)))
> for (i in 1:length(properties(myAPPT.list))){
         for (j in 1:length(columns(myAPPT.list))){
                 matH2[j,i]<-median(
                 multiMCMCglmm(myAPPT.list)[[i]][[j]]$VCV[,1]
                 /(multiMCMCglmm(myAPPT.list)[[i]][[j]]$VCV[,1]+
                 multiMCMCglmm(myAPPT.list)[[i]][[j]]$VCV[,2]))
         }
 }
> colnames(matH2)<-names(properties(myAPPT.list))
> rownames(matH2)<-columns(myAPPT.list)
> matH2
```

```
    CHOC760102 KYTJ820101
374  0.9976138  0.9935374
375  0.9975296  0.9932501
376  0.9890726  0.9779746
>
```

4. Conclusions

bcool is a simple package implemented in *S4*, which applies the PMM of Lynch (1991) in a Bayesian framework as proposed by Naya et al. (2006). The package allow to identify potentially interesting sites based on the statistical significance of the difference between classes of organisms, or directly by the effect sizes of the difference in relevant properties.

5. Acknowledgments

The authors are indebted to Florian Battke, Kay Nieselt, Héctor Romero, Natalia Rego and Martín Graña for helpful suggestions, software testing and careful revision of the documentation.

Referencias

G. M. Conenello, D. Zamarin, L. A. Perrone, T. Tumpey and P. Palese (2007) A single mutation in the PB1-F2 of H5N1 (HK/97) and 1918 influenza A viruses contributes to increased virulence. *PLoS Pathog.*, **3**, 1414-1421.

S. Falkow (1997) What is a Pathogen? Developing a definition of a pathogen requires looking closely at the many complicated relationships that exist among organisms. *ASM News*, **63**, 359-365.

J.D. Hadfield (2010) MCMC Methods for Multi-Response Generalized Linear Mixed Models: The MCMCglmm R Package. *Journal of Statistical Software* **33** 2:1-22

M. Lynch (1991) Methods for the analysis of comparative data in evolutionary biology. *Evolution* **45**:1065-1080

H. Marjuki, C. Scholtissek, J. Franks, N. J. Negovetich, J. R. Aldridge, R. Salomon, D. Finkelstein and R. G. Webster (2010) Three amino acid changes in PB1-F2 of highly pathogenic H5N1 avian influenza virus affect pathogenicity in mallard ducks. *Arch Virol.*, **155**, 925-934.

H. Naya, D. Gianola, H. Romero, J.I. Urioste and H. Musto (2006) Inferring Parameters Shaping Amino Acid Usage in Prokaryotic Genomes via Bayesian MCMC Methods. *Mol Biol Evol* **23**:203-211

E. V. Sokurenko, V. Chesnokova, D. E. Dykhuizen, I. Ofek, X. WU, K. A. Krogfelt, C. Struve, M. A. Schembri and D. L. Hasty (1998) Pathogenic adaptation of Escherichia coli by natural variation of the FimH adhesin. *Proc. Natl. Acad. Sci. U. S. A.*, **95**, 8922-8926.

L. Spangenberg, F. Battke, M. Graña, K. Nieselt and H. Naya (2011) Identifying associations between amino acid changes and meta information in alignments. *Bioinformatics* **27**(20):2782-9