

# Vignette for the **anoaint** package

Stephanie Kovalchik and Ravi Varadhan

March 6, 2013

This document gives a brief introduction to the main functions of the **anoaint** package. The package provides a set of tools for the “analysis of interactions” to investigate consistency of treatment effect with data from a parallel-group clinical trial.

The examples will make use of a simulated clinical trial data set, motivated by the structure of the Studies of Left Ventricular Dysfunction Trial (SOLVD-T), a placebo-controlled trial of the angiotensin-converting-enzyme inhibitor enalapril for patients with congestive heart failure (Yusuf 1991). The simulated data set is included with the package.

## Creating an **anoaint** object

### Specifying the prognostic factors

The fundamental object of the package is an **anoaint** object. This is what specifies family of model to be fit in all regression methods, the candidate prognostic factors for investigating effect modification, and whether a selection procedure is used for identifying prognostic factors.

In the following, we create an **anoaint** object for the mock SOLVD-T data, specifying age, pulse, lower ejection fraction, cardiothoracic ratio, and sodium as prognostic factors. The regression model for all the **anoaint** regression methods will be a logistic model on the outcome of time to hospitalization or death within two years of study entry.

```
set.seed(10101)
library(anoaint)
data(simsolvd)

simsolvd$event <- 1 - simsolvd$censor

obj <- anoaint(event ~ (age + beat + lvef + cardratio + sodium) *
  trt, family = "binomial", data = simsolvd)

obj

## event ~ (age + beat + lvef + cardratio + sodium) * trt
```

The option family can be any one of the canonical models in the GLM family or a Cox regression model which is specified by setting **family** to ‘coxph’.

### Model-based prognostic selection

If we were uncertain whether all of the factors were prognostic, we could perform a prognostic selection step prior to any interaction testing. When the **anoaint** option **select** is set to either

‘glmnet’ or ‘stepAIC’. When the `glmnet` method is selected, a lasso variable selection is performed on the control data in order to identify factors that have a marginal influence on the outcome. The default settings use a ten-fold cross-validation to estimate the penalty parameter  $\lambda$ . When `stepAIC` is selected, a stepwise selection procedure is used and the best model is chosen so as to minimize the AIC.

As an illustration of the use of these methods, suppose we were uncertain about the importance of the variables `noise`, a randomly generated variable, `sodium`, `cardratio`, and `lvef`. Using the lasso, we can force the known prognostic factors `age` and `beat` into the model with the `keep.vars` argument. This sets the `penalty.factor` in the lasso `glmnet` so that these variables must be retained.

```
simsolv$d$noise <- runif(nrow(simsolv))

obj <- anoint(event ~ (age + beat + lvef + cardratio + sodium + noise) *
  trt, family = "binomial", data = simsolv, select = "glmnet", keep.vars =
  c("age",
    "beat"))

## Performing selection procedure for prognostic model...
## 7 x 1 sparse Matrix of class "dgCMatrix"
##              1
## (Intercept) -0.68262
## age          0.23843
## beat         0.03343
## lvef         -0.31245
## cardratio    0.28266
## sodium       -0.22621
## noise        .
## Selected MIM:
## event ~ (age + beat + lvef + cardratio + sodium) * trt

obj

## event ~ (age + beat + lvef + cardratio + sodium) * trt
```

Note that an intercept variable is always retained in GLM models. In this example, `noise` was dropped from the set of prognostic factors based on the empirical evidence in the control group.

To place the same restriction with the ‘stepAIC’ selection method, we can specify the lower and upper models with the `scope` argument of `stepAIC`.

```
thescope <- list(lower = event ~ (age + beat), event ~ (age + beat +
  lvef + cardratio + sodium + noise))

obj <- anoint(event ~ (age + beat + lvef + cardratio + sodium + noise) *
  trt, family = "binomial", data = simsolv, select = "stepAIC", scope =
  thescope)

## Performing selection procedure for prognostic model...
## Start:  AIC=1651
```

```
## event ~ age + beat + lvef + cardratio + sodium + noise
##
##           Df Deviance  AIC
## - noise      1      1637 1649
## <none>         1637 1651
## - cardratio   1      1644 1656
## - sodium      1      1655 1667
## - lvef        1      1669 1681
##
## Step:  AIC=1649
## event ~ age + beat + lvef + cardratio + sodium
##
##           Df Deviance  AIC
## <none>         1637 1649
## - cardratio   1      1644 1654
## - sodium      1      1655 1665
## - lvef        1      1669 1679
## Selected MIM:
## event ~ (age + beat + lvef + cardratio + sodium) * trt

obj

## event ~ (age + beat + lvef + cardratio + sodium) * trt
```

## PR-plot

Before formal testing of multiple interactions, we might be interested to see whether a proportional interaction model is reasonable. The proportional interaction model (PIM) postulates a constant effect modification on a score of the prognostic treatment-response factors. The score is linear for the specified regression model. To inspect proportional interactions, we use the PR-plot, for ‘proportional response’. This bins the outcomes of the logistic model based on the baseline prognostic score then computes the treatment effect within the decile groups. If the treatment effects are linear with respect to the prognostic score, this would be evidence of proportional interactions.

To generate the PR-plot, apply the `plot` method of the `anoimt` class.

```
plot(obj, ylab = "treatment effect (odds ratio)")
```

The shaded region of Figure 1 denotes the 95% confidence region for the overall treatment effect. Thus, the plot can give an idea about heterogeneity as a function of baseline prognosis, as determined by the candidate prognostic treatment-response factors.

## Forest plot

We can also create a more traditional plot of subgroup effects using the function `forest`. All of the factors included must be categorical, so we first create categorical representations of the continuous variables. Because all of the continuous variables have been centered and scaled, a value greater than zero corresponds to an individual that has a greater than average value for the specified variable.

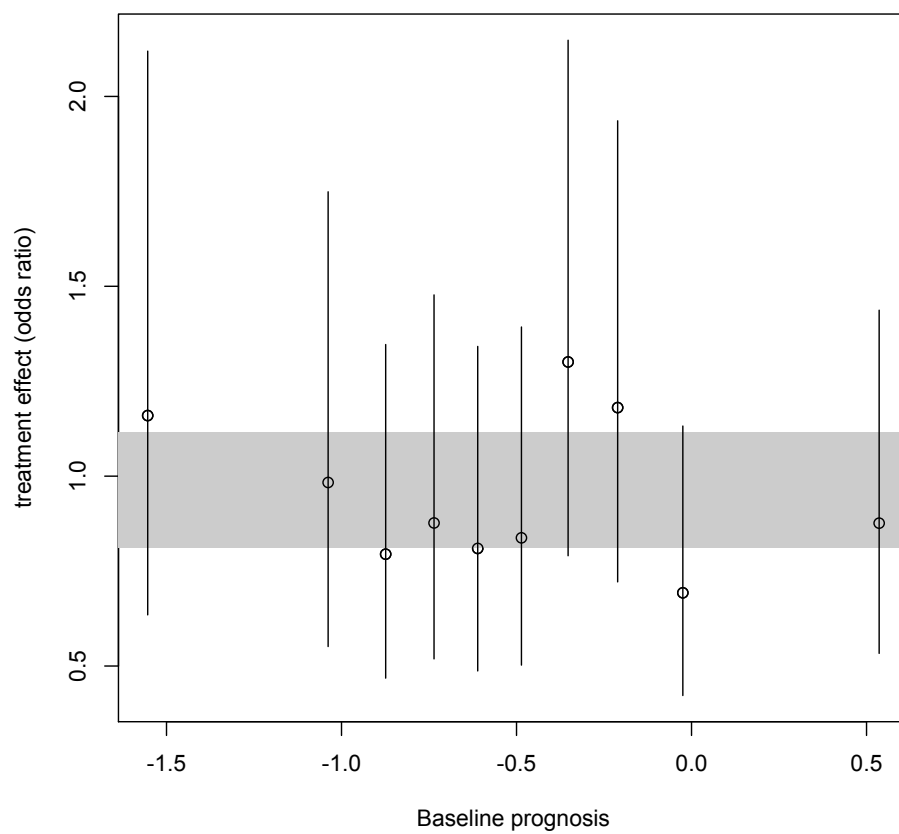


Figure 1: PR-plot

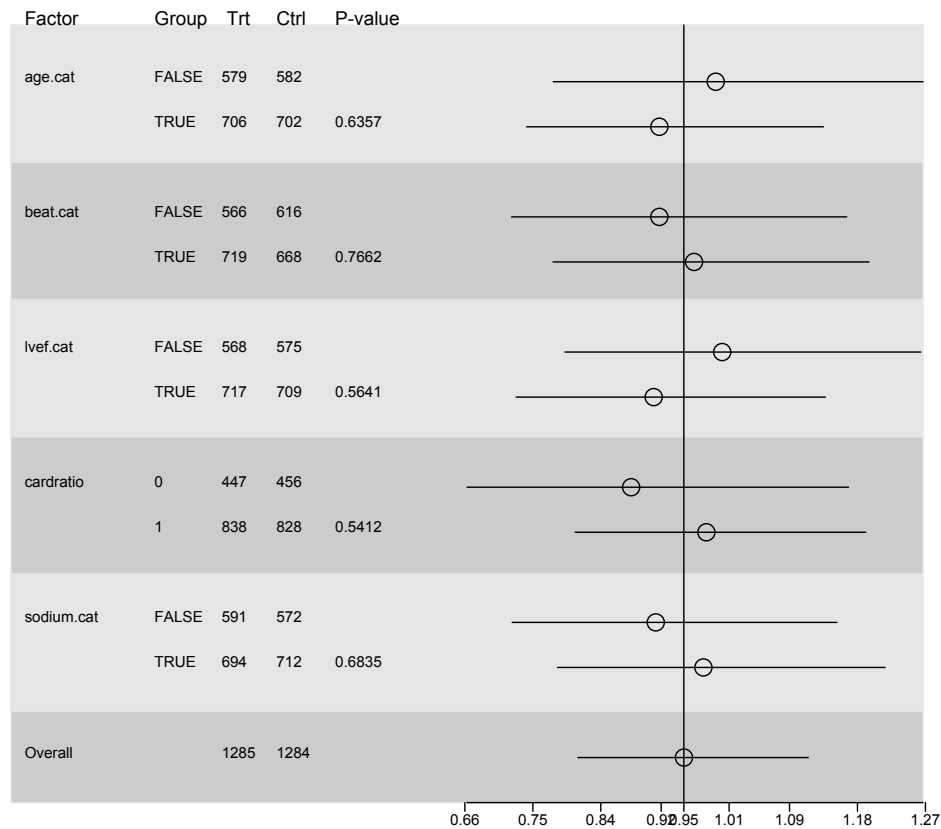


Figure 2: Forest plot of `anoint` object.

```

simsolvd$age.cat <- factor(simsolvd$age > 0)
simsolvd$beat.cat <- factor(simsolvd$beat > 0)
simsolvd$lvef.cat <- factor(simsolvd$lvef > 0)
simsolvd$sodium.cat <- factor(simsolvd$sodium > 0)

obj <- anoint(event ~ (age.cat + beat.cat + lvef.cat + cardratio +
  sodium.cat) * trt, family = "binomial", data = simsolvd)

forest(obj)

```

The result is a forest plot that shows the treatment effect estimate within each of the subgroups defined by the set of candidate prognostic treatment-response factors (Figure 2). The default settings also include the number of treated and control subjects in each subgroup and the p-value for the univariate test of interaction, based on the likelihood ratio statistic. Additional arguments can be supplied to modify the plot labeling and the number of terms included.

## Global tests of effect modification

Several tests for the presence of effect modification among the candidate treatment-response factors are provided by the **anoaint** package. These include tests based on one-by-one (OBO), unrestricted interactions (UIM), and proportional interactions (PIM) models of treatment-covariate interaction. OBO tests each prognostic factor one-at-a-time in a model of treatment, prognostic factor, and interaction effects. Thus, each model in the OBO is univariate with respect to treatment-covariate interaction. The UIM is a full regression model with all main effects and pairwise treatment-covariate interactions. The PIM is a proportional interactions model and is fit with either an exact or approximate method.

Likelihood ratio tests for any interaction (any proportional interaction in the case of PIM) is provided with the **anoaint.fit** function. This includes one-stage tests for each method and two-stage tests which have the exact PIM global test at the first stage and, if no evidence of proportional interactions is found, OBO or UIM at the second stage. For the two-stage testing, each stage uses a one-half of the global  $\alpha$ , which is specified by the **level** option.

In what follows, we obtain an **anoaint.fit** object at the 5% global level of significance. The **summary** method gives a matrix with logical indicators for whether a given method rejected the null hypothesis of no interaction. Note that for PIM, the null is no *proportional* interaction. Thus, if the null is not rejected with the PIM test, it does not exclude the possibility that non-proportional interaction is present.

```
obj <- anoaint(event ~ (age + beat + lvef + cardratio + sodium) *
  trt, family = "binomial", data = simsolvd)

fit <- anoaint.fit(obj, level = 0.05)

summary(fit)

##               Global null rejected
## OBO                      FALSE
## OBO (adj.)                 FALSE
## UIM                        FALSE
## PIM (exact)                FALSE
## PIM (approx)               FALSE
## PIM/OBO (adj.)             FALSE
## PIM/UIM                    FALSE
##
##               Global LRT (p-value)
## OBO (max)                   0.2623
## UIM                         0.5242
## PIM (exact)                 0.6309
## PIM (approx)                0.6344
```

The **summary** method also reports each likelihood-ratio test p-value. The global test is based on the maximum LRT among the univariate tests for OBO. When corrected for multiplicity using a Bonferroni correction, this is compared to the  $\chi^2(1)$  at level  $\alpha/K$ , where  $K$  is the number of univariate tests. The p-value for UIM is for the LRT comparing a model with all pairwise treatment-covariate interactions, to a model with only main effects. The PIM models are one degree of freedom tests of the responsiveness parameter  $\theta$ .

## Model fits

Any of the `anoaint` models can be extracted using the function `fits` and indicating the `type` (any one or more of `'obo'`, `'uim'`, `'pim.exact'`, `'pim.approx'`).

```
pim.fit <- fits(fit, type = "pim.exact")
```

```
pim.fit

## $pim.exact
##
## Baseline/treatment effects:
##
##           Estimate
## Control   -0.76751
## Treatment -0.05184
##
## Prognostic effects:
##
##           Estimate
## age         0.23327
## beat        0.07102
## lvef        -0.32477
## cardratio   0.40897
## sodium      -0.19560
##
## Responsiveness parameter:
##
##           Estimate    LRT p-value
## theta    0.9121 0.2309  0.6309
##
```

This is a list with an object of class `'pim'`. Each type of model has the standard set of methods including: `print`, `summary`, `coef`, `vcov`, `predict`, and `confint`. For example, if we were interested in the confidence intervals for the intercept in the control group and for the `sodium` term of the exact PIM, we could obtain them with the `confint` method.

```
confint(pim.fit$pim.exact, parm = c("Control", "sodium"))
```

```
##           2.5 %  97.5 %
## Control -0.5985 -0.9365
## age      0.3220  0.1445
```

In the above, the default variance-covariance for the model terms regards the responsiveness parameter as fixed. This could underestimate the standard error. As an alternative, the variance-covariance can be estimated using bootstrap resampling with the `n.boot` option set to some positive integer in the `pim` call. An example of this is provided in the next section.

## Direct fitting with anoaint object

We could also obtain any fit directly from the `anoaint.object`. For example, to fit the one-by-one regression models:

```

obo.fit <- obo(obj) # A list of fits

obo.fit$fit

## $`event ~ age * trt`
##
## Call: glm(formula = f, family = anoint@formula@family, data = anoint@data)
##
## Coefficients:
## (Intercept)      age      trt      age:trt
##      -0.4940      0.1802     -0.0489     -0.0317
##
## Degrees of Freedom: 2568 Total (i.e. Null);  2565 Residual
## Null Deviance:      3400
## Residual Deviance: 3380  AIC: 3390
##
## $`event ~ beat * trt`
##
## Call: glm(formula = f, family = anoint@formula@family, data = anoint@data)
##
## Coefficients:
## (Intercept)      beat      trt      beat:trt
##      -0.4880      0.0516     -0.0591      0.0825
##
## Degrees of Freedom: 2568 Total (i.e. Null);  2565 Residual
## Null Deviance:      3400
## Residual Deviance: 3390  AIC: 3400
##
## $`event ~ lvef * trt`
##
## Call: glm(formula = f, family = anoint@formula@family, data = anoint@data)
##
## Coefficients:
## (Intercept)      lvef      trt      lvef:trt
##      -0.4993     -0.3105     -0.0529      0.0129
##
## Degrees of Freedom: 2568 Total (i.e. Null);  2565 Residual
## Null Deviance:      3400
## Residual Deviance: 3340  AIC: 3350
##
## $`event ~ cardratio * trt`
##
## Call: glm(formula = f, family = anoint@formula@family, data = anoint@data)
##
## Coefficients:
## (Intercept)      cardratio      trt      cardratio:trt
##      -0.753      0.402     -0.128      0.107
##
## Degrees of Freedom: 2568 Total (i.e. Null);  2565 Residual
## Null Deviance:      3400

```



```
## Residual Deviance: 3370 AIC: 3380
##
## $`event ~ sodium * trt`
##
## Call: glm(formula = f, family = anoint@formula@family, data = anoint@data)
##
## Coefficients:
## (Intercept)      sodium          trt      sodium:trt
##      -0.4837      -0.2154      -0.0649       0.0920
##
## Degrees of Freedom: 2568 Total (i.e. Null); 2565 Residual
## Null Deviance:      3400
## Residual Deviance: 3380 AIC: 3390
##
```

The `obo` function, like all of the `anoint` models, returns a list with the fitted model `fit`, the likelihood ratio test statistics for each univariate test of interaction, and the corresponding unadjusted `pvalue`.

Here is an example of fitting the UIM.

```
uim.fit <- uim(obj)
summary(uim.fit$fit)

##
## Call:
## glm(formula = object@formula@formula, family = object@formula@family,
##      data = object@data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.561  -0.973  -0.784   1.265   1.946
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.7121    0.1027  -6.94 4.0e-12 ***
## age           0.2424    0.0621   3.90 9.5e-05 ***
## beat          0.0302    0.0598   0.51  0.61
## lvef          -0.3335    0.0604  -5.52 3.4e-08 ***
## cardratio     0.3249    0.1265   2.57  0.01 *
## sodium        -0.2503    0.0602  -4.16 3.2e-05 ***
## trt           -0.1696    0.1474  -1.15  0.25
## age:trt       -0.0346    0.0866  -0.40  0.69
## beat:trt      0.0828    0.0854   0.97  0.33
## lvef:trt      0.0413    0.0860   0.48  0.63
## cardratio:trt 0.1434    0.1797   0.80  0.42
## sodium:trt    0.1312    0.0839   1.56  0.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
## Null deviance: 3396.7 on 2568 degrees of freedom
## Residual deviance: 3270.1 on 2557 degrees of freedom
## AIC: 3294
##
## Number of Fisher Scoring iterations: 4
##

uim.fit$LRT

## [1] 4.177

uim.fit$pvalue

## [1] 0.5242
```

In the following, we fit an exact PIM and specify that a bootstrap estimate of the variance-covariance for all model terms be taken using 100 resamples.

```
pim.fit <- pim(obj, n.boot = 100)
summary(pim.fit)

##           Estimate      SE t value  p-value
## Control   -0.76751 0.08927 -8.5974 1.396e-17
## Treatment -0.05184 0.09217 -0.5624 5.739e-01
## age        0.23327 0.05128  4.5491 5.638e-06
## beat       0.07102 0.04668  1.5215 1.283e-01
## lvef       -0.32477 0.05112 -6.3526 2.497e-10
## cardratio  0.40897 0.09965  4.1040 4.188e-05
## sodium     -0.19560 0.04916 -3.9788 7.118e-05
## theta      0.91207 0.17035  5.3540 9.369e-08
```

## All subsets for proportional interactions model

Conditional selection can be problematic because the selection procedure influences the type I error in ways that the trialist may not be able to control. As a more reliable approach, we have developed an all subsets procedure for the proportional interaction model. Given  $p$  hypothesized covariates, we fit  $2^p - (p + 1)$  possible proportional interactions models, supposing that each factor contributes exactly one term into the regression model. A modified-Bonferroni significance criterion is used that strongly controls the familywise error rate at level **fw**, while accounting for the correlation among the proportional interactions subsets.

The function `pim.subsets` performs the all subsets procedure. In this example, we perform the model for the five covariates of the SOLVD-T data. The result 'fit' is a list with the items contained in the summary table.

```
fit <- pim.subsets(event ~ age + beat + lvef + cardratio + sodium,
  trt = "trt", family = "binomial", data = simsolv, fwr = 0.05)

## subset interaction num      LRT lower upper pvalue reject
## 8 01100 0.9932 2 0.0006532 0.5829 1.728 0.9796 FALSE
```

```
## 7 01011 0.9849 3 0.0027204 0.5346 1.870 0.9584 FALSE
## 23 11100 0.9685 3 0.0211576 0.6382 1.489 0.8844 FALSE
## 22 11011 0.9539 4 0.0359637 0.5759 1.547 0.8496 FALSE
## 17 10110 1.0480 3 0.0556031 0.7337 1.579 0.8136 FALSE
## 11 01111 0.9437 4 0.0810962 0.6128 1.366 0.7758 FALSE
## 13 10010 1.0976 2 0.0895017 0.6123 2.163 0.7648 FALSE
## 1 00011 0.9044 2 0.1114063 0.4656 1.649 0.7385 FALSE
## 4 00111 0.9325 3 0.1167920 0.6114 1.377 0.7325 FALSE
## 19 11000 1.1559 2 0.1181688 0.4889 3.028 0.7310 FALSE
## 15 10100 0.9224 2 0.1280426 0.5840 1.484 0.7205 FALSE
## 25 11110 1.0824 4 0.1646117 0.7486 1.639 0.6849 FALSE
## 26 11111 0.9273 5 0.1749535 0.6534 1.293 0.6757 FALSE
## 3 00110 1.1045 2 0.1945252 0.7267 1.786 0.6592 FALSE
## 14 10011 0.8708 3 0.2845912 0.5159 1.454 0.5937 FALSE
## 10 01110 1.1284 3 0.2917422 0.7276 1.730 0.5891 FALSE
## 18 10111 0.9049 4 0.2977587 0.6263 1.328 0.5853 FALSE
## 5 01001 0.7505 2 0.3858484 0.2286 1.936 0.5345 FALSE
## 20 11001 0.8018 3 0.4602770 0.3702 1.493 0.4975 FALSE
## 21 11010 1.2314 3 0.5178130 0.6972 2.084 0.4718 FALSE
## 9 01101 0.8161 3 0.7327847 0.5040 1.285 0.3920 FALSE
## 2 00101 0.8002 2 0.8671506 0.4632 1.276 0.3517 FALSE
## 6 01010 1.4185 2 0.9375340 0.7041 3.703 0.3329 FALSE
## 24 11101 0.8232 4 0.9379614 0.5658 1.208 0.3328 FALSE
## 16 10101 0.7939 3 1.2714405 0.5311 1.193 0.2595 FALSE
## 12 10001 0.6597 2 1.4005664 0.3005 1.303 0.2366 FALSE
```

Each row of the fitted result is a proportional interactions model. The `subset` variable shows which covariates specified in the formula argument were included in the particular fit (1=included, 0=not included). The rejection status on the far right is based on the multiplicity-corrected significance criterion. We see that no proportional interactions LRT was rejected at the corrected significance criterion.

We could also display the results using the `forest.subsets` function (Figure 3).

```
forest.subsets(fit)
```

We can also use a similar syntax with `anooint.subgroups` to perform one-by-one subgroup analyses and significance testing with a Bonferroni correction, and then plot these findings with `forest.subsets` (Figure 4).

```
fit <- anooint.subgroups(event ~ age + beat + lvef + cardratio + sodium,
  trt = "trt", family = "binomial", data = simsolvd, fwer = 0.05)

## subset interaction LRT lower upper pvalue covariates
## sodium 00001 0.09205 1.2566 -0.06899 0.2531 0.2623 sodium
## beat 01000 0.08247 1.0198 -0.07765 0.2426 0.3126 beat
## cardratio 00010 0.10721 0.3734 -0.23673 0.4512 0.5412 cardratio
## age 10000 -0.03173 0.1450 -0.19507 0.1316 0.7034 age
## lvef 00100 0.01288 0.0243 -0.14910 0.1749 0.8761 lvef
## reject
## sodium FALSE
```

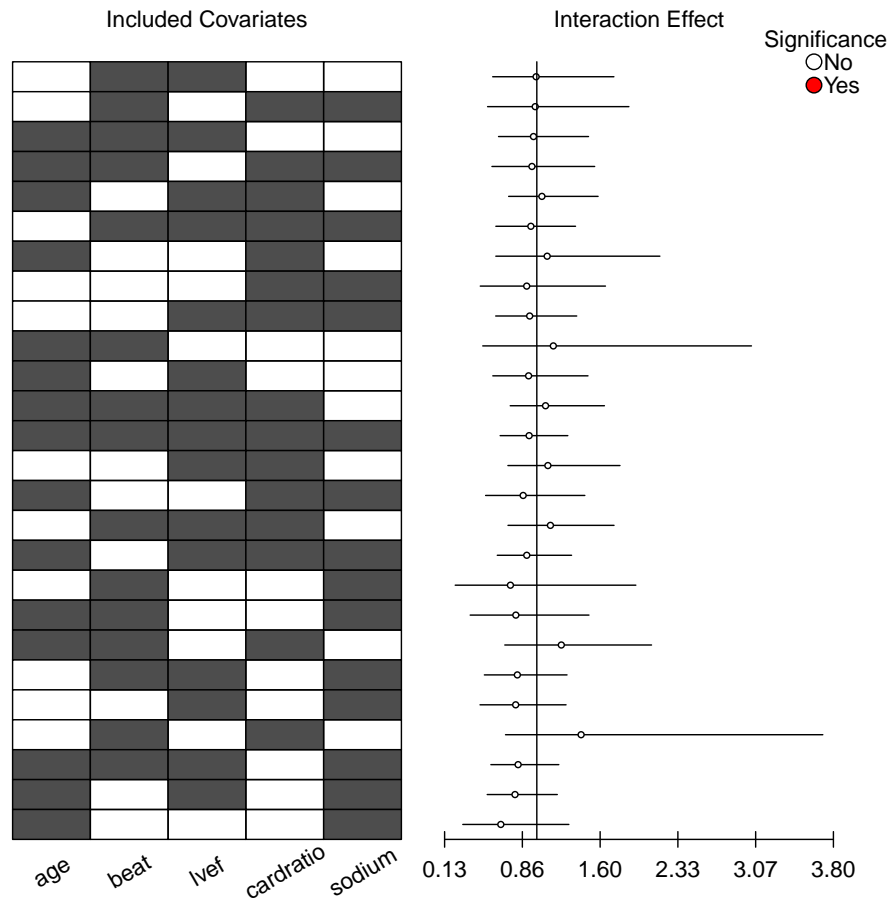


Figure 3: Forest plot of all proportional interactions model subsets.

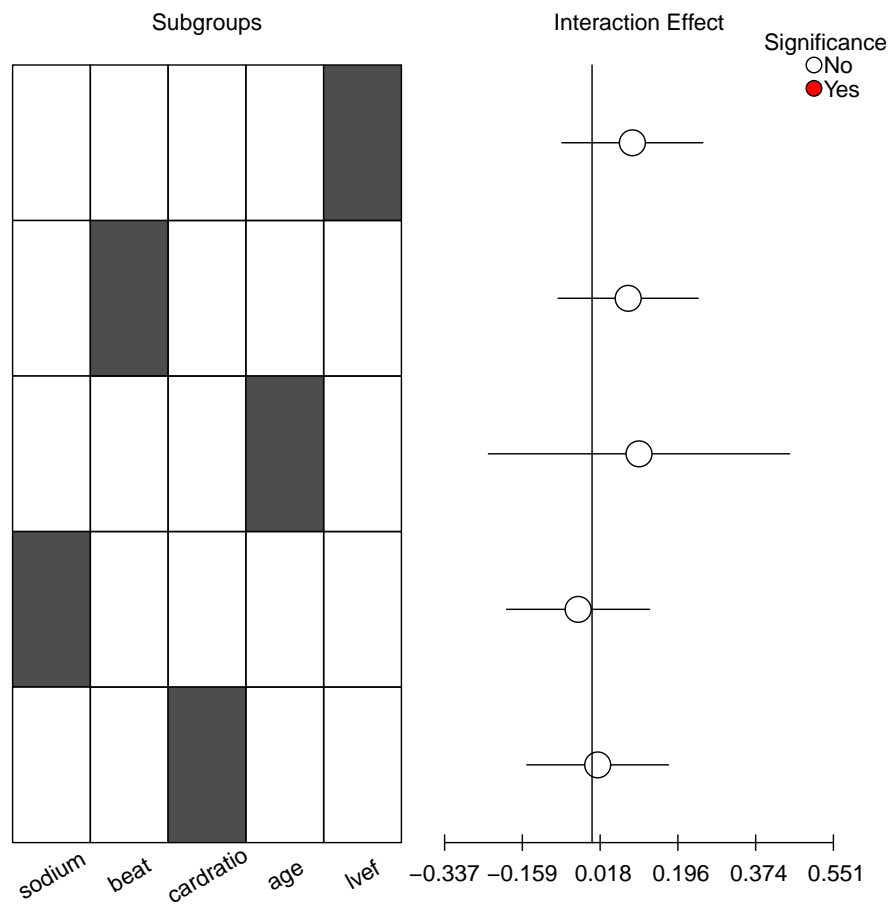


Figure 4: Forest plot of one-by-one subgroups for the SOLVD-T analysis.

```
## beat      FALSE
## cardratio FALSE
## age       FALSE
## lvef      FALSE
```

## References

Follmann DA, Proschan MA. A multivariate test of interaction for use in clinical trials. *Biometrics* 1999; 55(4):1151-1155

Yusuf, S. et al. Effect of Enalapril on Survival in Patients with Reduced Left-Ventricular Ejection Fractions and Congestive-Heart-Failure. *NEJM* 1991; 325:293-302.