gWQS: An R Package for Linear and Generalized Weighted Quantile Sum (WQS) Regression

Stefano Renzetti
University of Brescia

Chris Gennings
Icahn School of Medicine at Mount Sinai

Paul C. Curtin
Icahn School of Medicine at Mount Sinai

Abstract

Weighted Quantile Sum (WQS) regression is a statistical model for multivariate regression in high-dimensional datasets commonly encountered in environmental exposures. The model constructs a weighted index estimating the mixture effect associated with all predictor variables on an outcome. The package **gWQS** extends WQS regression to applications with continuous, categorical and count outcomes. We provide four examples to illustrate the usage of the package.

Keywords: WQS, Weighted Quantile Sum, regression, mixture.

1. Introduction

Statistical methods appropriate for the simultaneous evaluation of high-dimensional predictor sets are a critical focus in biostatistics and related quantitative fields, as well as in applied contexts including epidemiology, genomics, and related biological disciplines. While classical strategies for addressing high-dimensional feature sets have focused either on variable-selection methods or dimensionality-reduction techniques(like Principal Component Analysis (PCA) ridge regression (Hoerl and Kennard 1970), lasso (Tibshirani 1996), adaptive lasso (Zou 2006), and elastic net (Zou and Hastie 2005)), alternative strategies focusing on the mixture effect are becoming increasingly popular. Weighted quantile sum (WQS) regression (Carrico, Gennings, Wheeler, and Factor-Litvak 2015; Czarnota, Gennings, and Wheeler 2015; Gennings, Carrico, Factor-Litvak, Krigbaum, Cirillo, and Cohn 2013; Horton, Blount, Valentin-Blasini, Wapner, Whyatt, Gennings, and Factor-Litvak 2015; Brunst, Guerra, Gennings, Hacker, Jara, Enlow, Wright, Baccarelli, and Wright 2017) is such a mixture effect strategy that incorporates elements of both feature selection and dimensionality reduction to assess both the overall mixture effect of a given set of predictors, and the discrete contribution of constituent predictors to this overall effect. Here we introduce the **gWQS** package of the statistical software R (R Core Team 2017) for the implementation of WQS regression in contexts with continuous, categorical, and count-based outcomes.
WQS regression constructs a weighted index estimating the mixture effect of mixture components on an outcome, which may then be used in a regression model with relevant covariates.

The mixture effect associated with the additive combination of the mixture components is thereby assessed through a standard regression test on the weighted index, while the estimation of weights associated with each individual predictors allows for the assessment of the discrete effects of each individual predictor on the dependent variable.
The WQS model (Carrico *et al.* 2015) has the following equation:

$$g(\mu) = \beta_0 + \beta_1 \left(\sum_{i=1}^{c} w_i q_i \right) + \boldsymbol{z'\varphi} \tag{1}$$

where g is the link function as in generalized linear model, μ is the mean of the outcome, q_i is the quantile of the i^{th} component, w_i is the weight (to be estimated) associated with the i^{th} component, $\boldsymbol{z'}$ is the vector of covariates and $\boldsymbol{\varphi}$ is the vector of parameters associated with the covariates. The $(\sum_{i=1}^{c} w_i q_i)$ term represents the index that weights and sums the components included in the mixture. Two constrains are applied to the weights: $\sum_{i=1}^{c} w_i = 1$ and $0 \leq w_i \leq 1$. To estimate the model, the dataset may be split in a training and a validation dataset: the first one to be used for the weight estimation, the second one to test the significance of the final WQS index. In order to estimate the weights, the bootstrap method is applied. For each bootstrap sample (usually $B = 2$ total samples) a dataset is created sampling with replacement from the training dataset and the parameters of the model in equation 1 ($\theta = (\beta_0, \beta_1, w_1, \ldots, w_c, \boldsymbol{\varphi})$) are estimated through an optimization algorithm where the loglikelihood is used as the objective function:

$$\hat{\theta}_{WQS} = \underset{\theta}{\operatorname{argmax}} \left[l(\theta; y) + \lambda \left(\sum_{i=1}^{c} w_i - 1 \right) \right]$$

where $l(\theta; y)$ is the log-likelihood function and λ is the lagrangian coefficient associated with the equality constraint in which the weights have to sum to 1. An inequality constraint is also applied in order to impose that $0 \leq w_i \leq 1$.
Once the weights are estimated the model is fitted in order to find the regression coefficients in each ensemble step. After the bootstrap ensemble is complete, the estimated weights are averaged across bootstrap samples to obtain the WQS index:

$$WQS = \sum_{i=1}^{c} \bar{w}_i q_i$$

where $\bar{w}_i = \frac{1}{\sum_{b=1}^{B} f(\beta_{1(b)})} \sum_{b=1}^{B} w_{i(b)} f(\beta_{1(b)})$ and $f(\beta_{1(b)})$ is a signal function that we will specify later in the text. Typically weights are estimated in a training set then used to construct a WQS index in a validation set, which can be used to test to evaluate the association and significance of the mixture to the health outcome in a standard generalized linear model, as:

$$g(\mu) = \beta_0 + \beta_1 WQS + \boldsymbol{z'\varphi}$$

Due to the structure of the model either a positive or a negative direction of the association between the dependent variable and the WQS index has to be chosen; that is, the model is inherently one-directional, in that it tests only for mixture effects positively or negatively associated with a given outcome. In practice analyses should therefore be run twice to test for

associations in either direction. The specification of a test for positive or negative association determines the form of the signal function:

$$f(\hat{\beta}_{1(b)}) = \begin{cases} 1, & \text{if } \hat{\beta}_{1(b)} \text{ and the chosen direction have the same sign} \\ 0, & \text{if } \hat{\beta}_{1(b)} \text{ and the chosen direction have different sign} \end{cases}$$

After the final model is fitted we can test the significance of the β_1 to see if there is an association between the WQS index and the outcome. In the case the coefficient is significantly different from 0 then we can interpret the weights: the highest values identify the associated components as the relevant contributors in the association. A selection threshold can be decided a priori as $\tau = 1/c$ to identify those chemicals that have a significant weight in the index.
Since the WQS regression can be generalised and applied to multiple types of dependent variables, different objective functions have to be defined to find the optimal weights. For a linear regression the following function is minimised:

$$\hat{\theta}_{WQS} = \underset{\theta}{\operatorname{argmin}} \left[\sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \beta_1 \sum_{j=1}^{c} w_j q_j + \boldsymbol{z'\varphi} \right) \right)^2 + \lambda \left(\sum_{j=1}^{c} w_j - 1 \right) \right]$$

For a logistic regression the following likelihood is maximised:

$$\begin{aligned} \hat{\theta}_{WQS} &= \underset{\theta}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \left(y_i \times \log \left(\frac{1}{1 + \exp(\beta_0 + \beta_1 \sum_{j=1}^{c} w_j q_j + \boldsymbol{z'\varphi})} \right) \right. \right. \\ &\quad \left. \left. + \left(1 - y_i \right) \times \log \left(1 - \beta_0 + \beta_1 \sum_{j=1}^{c} w_j q_j + \boldsymbol{z'\varphi} \right) \right] \end{aligned}$$

The equation to be maximised for a multinomial regression is the following:

$$\begin{aligned} \hat{\theta}_{WQS} &= \underset{\theta}{\operatorname{argmax}} \left\{ \sum_{i=1}^{n} \left[\sum_{l=1}^{L-1} \left(y_{ij} \left(\beta_{0l} + \beta_{1l} \sum_{j=1}^{c} w_{lj} q_{ij} + \boldsymbol{z'\varphi} \right) \right. \right. \right. \\ &\quad \left. \left. \left. - \log \left(1 + \sum_{l=1}^{L-1} \exp \left(\beta_{0l} + \beta_{1l} \sum_{j=1}^{c} w_{lj} q_{ij} + \boldsymbol{z'\varphi} \right) \right) \right) \right] \right\} \end{aligned}$$

The objective function used to estimate the weights in a Poisson regression is:

$$\hat{\theta}_{WQS} = \underset{\theta}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \left(y_i \times \left(\beta_0 + \beta_1 \sum_{j=1}^{c} w_j q_j + \boldsymbol{z'\varphi} \right) - \exp \left(\beta_0 + \beta_1 \sum_{j=1}^{c} w_j q_j + \boldsymbol{z'\varphi} \right) \right] \right]$$

In the case of a negative binomial regression the likelihoods to be maximised is:

$$\begin{aligned} \hat{\theta}_{WQS} &= \underset{\theta}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \left(y_i \log(\alpha) + y_i \left(\beta_0 + \beta_1 \sum_{j=1}^{c} w_j q_j + \boldsymbol{z'\varphi} \right) \right. \right. \\ &\quad - \left(y_i + 1/\alpha \right) \log \left(1 + \alpha \exp \left(\beta_0 + \beta_1 \sum_{j=1}^{c} w_j q_j + \boldsymbol{z'\varphi} \right) \right) \\ &\quad \left. \left. + \log(\Gamma(y_i + 1/\alpha)) - \log(\Gamma(y_i + 1)) - \log(\Gamma(1/\alpha)) \right) \right] \end{aligned}$$

2. The gWQS package

The R package **gWQS** extends WQS regression to applications with continuous, categorical and count outcomes. In particular, this package uses the `solnp()` function from the **Rsolnp** package as optimization algorithm to estimates the weights. This function solves general nonlinear programming problems through the augmented Lagrangian multiplier method (Ye 1987; Ghalanos and Theussl 2015).
We list four examples to illustrate the usage of the package.

2.1. Example 1

The main function of the **gWQS** package is `gwqs()`, which allows the implementation of WQS regression for linear, logistic, multinomial, Poisson, quasi-Poisson and negative binomial regression. For Poisson and negative binomial regression a zero inflated option is also implemented. We created the `wqs_data` dataset (available once the package is installed and loaded) to demonstrate the use of this function. These data reflect 34 exposure concentrations simulated from a distribution of PCB exposures measured in subjects participating in the NHANES study (2001-2002). Additionally, an end-point measure, simulated from a distribution of leukocyte telomere length (LTL), a biomarker of chronic disease, is provided as well (variable name: `y`), along with simulated dichotomous (variable name: `y_bin`), multinomial (variable name: `y_multinom`) and count (variable name: `y_count`) outcome variables and covariates, e.g. `sex`. This dataset can thus be used to test the **gWQS** package by analyzing the mixture effect of the 34 simulated PCBs on the outcomes, with adjustments for covariates. The following script calls a WQS model for a continuous outcome using the function `gwqs()`; we also show the script to reproduce the plots and tables that are automatically generated when setting the options `plots = TRUE, tables = TRUE`:

```
R> # we save the names of the mixture variables in the variable "toxic_chems"
R> toxic_chems <- names(wqs_data)[1:34]
R> # we run the model and save the results in the variable "results"
R> results <- gwqs(y ~ wqs, mix_name = toxic_chems,
+                  data = wqs_data, q = 4, validation = 0.6, b = 2,
+                  b1_pos = TRUE, b1_constr = FALSE, family = "gaussian",
+                  seed = 2016, plots = TRUE, tables = TRUE)
R> #
R> # bar plot
R> w_ord <- order(results$final_weights$mean_weight)
R> mean_weight <- results$final_weights$mean_weight[w_ord]
R> mix_name <- factor(results$final_weights$mix_name[w_ord],
+                      levels = results$final_weights$mix_name[w_ord])
R> data_plot <- data.frame(mean_weight, mix_name)
R> ggplot(data_plot, aes(x = mix_name, y = mean_weight, fill = mix_name)) +
+    geom_bar(stat = "identity", color = "black") + theme_bw() +
+    theme(axis.ticks = element_blank(),
+          axis.title = element_blank(),
+          axis.text.x = element_text(color='black'),
+          legend.position = "none") + coord_flip()
```

```
R> #
R> # scatter plot y vs wqs
R> ggplot(results$y_wqs_df, aes(wqs, y_adj)) + geom_point() +
+    stat_smooth(method = "loess", se = FALSE, size = 1.5) + theme_bw()
R> #
R> # scatter plot residuals vs fitted values
R> fit_df <- broom::augment(results$fit)
R> ggplot(fit_df, aes(x = .fitted, y = .resid)) + geom_point() +
+    theme_bw() + xlab("Fitted values") + ylab("Residuals")
R>
```

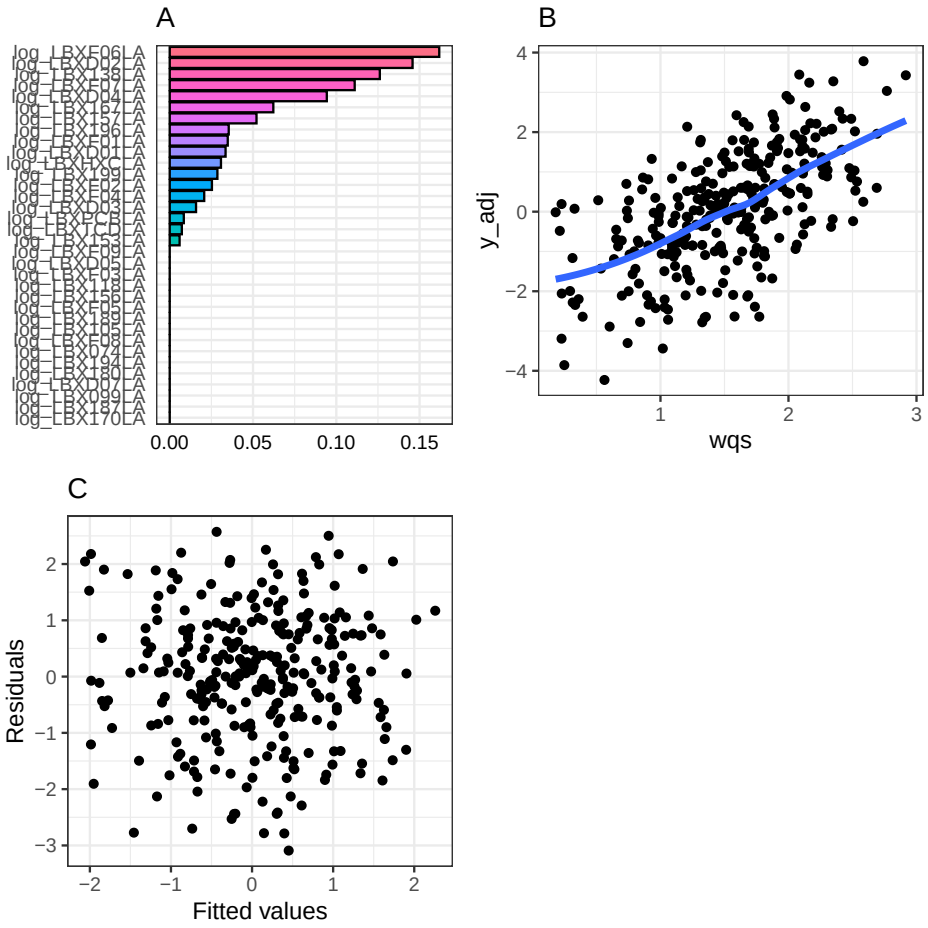


Figure 1: Plots displayed for linear outcomes when `plots = TRUE`

This WQS model tests the relationship between our dependent variable, `y`, and a WQS index estimated from ranking exposure concentrations in quartiles (`q = 4`); the `wqs` term must be included in the `formula`). It also divided the data for training and validation, with 40% of the dataset for training and 60% for validation (`validation = 0.6`), and assigned 2 bootstrap samples (`b = 2`) for parameter estimation (in practical applications we suggest at least 100 bootstrap samples to be used). Because WQS provides a unidirectional evaluation

of mixture effects, we first examined weights derived from bootstrap models where β_1 was positive (`b1_pos = TRUE`); we could test for negative associations by setting that parameter to be false (`b1_pos = FALSE`). We can also choose to constrain the β_1 to be positive (`b1_pos = TRUE` and `b1_constr = TRUE`) or negative (`b1_pos = FALSE` and `b1_constr = TRUE`) when we estimate the weights; in the case of example 1 we are not applying a constraint to β_1. We linked our model to a gaussian distribution to test for relationships between the continuous outcome and exposures (`family = "gaussian"`), and fixed the seed to 2016 for reproducible results (`seed = 2016`). We plotted a summary model with loess fit, and a summary of each variables' relative weight, and the residuals vs fitted values plot (`plots = TRUE`). The command `tables = TRUE` automatically generates in the Viewer pane the tables of the weight ranked list and the model summary.
Figure 1 A is a barplot showing the weights assigned to each variable ordered from the highest weight to the lowest. These results indicate that the variables `log_LBXF06LA` and `log_LBXD02LA` are the largest contributors to this mixture effect, with the first 6 chemicals explaining more than the 70% of the total weights.
In plot B of figure 1 we have a representation of the wqs index vs the outcome (adjusted for the model residual when covariates are included in the model) that shows the direction and the shape of the association between the exposure and the outcome. For example, in this case we can observe a linear and positive relationship between the mixture and the `y` variable.
In plot C a diagnostic graph of the residuals vs the fitted values is shown to check if they are randomly spread around zero or if there is a trend.
To test the statistical significance of the association between the variables in the model, the following code has to be run:

```
R> summary(results$fit)

Call:
glm(formula = formula, family = family, data = bdtf)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-3.09131  -0.71326   0.06459   0.78517   2.57249

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.3453     0.1909  -12.29   <2e-16 ***
wqs           1.5785     0.1187   13.30   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 1.304685)

    Null deviance: 619.48  on 299  degrees of freedom
Residual deviance: 388.80  on 298  degrees of freedom
AIC: 935.14

Number of Fisher Scoring iterations: 2
```

This result tells us that the association is positive and statistically significant (p=0.025).
To have the exact values of the estimated weights we can apply the command results$final_weights. The following code shows the first six highest weights; the full list of weights can be called by omitting the head function:

```
R> head(results$final_weights)
```

```
                mix_name mean_weight
log_LBXF06LA log_LBXF06LA  0.16181177
log_LBXD02LA log_LBXD02LA  0.14579100
log_LBX138LA log_LBX138LA  0.12612569
log_LBXF07LA log_LBXF07LA  0.11104863
log_LBXD04LA log_LBXD04LA  0.09430793
log_LBX167LA log_LBX167LA  0.06219165
```

These tables are also shown in the Viewer window when we set tables = TRUE.
The gwqs() function gives back other outputs like the vector of the values that indicate whether the solver has converged (0) or not (1 or 2) (results$conv), the matrix with all the estimated weights and the associated β_1, standard errors, statistics and p-values for each bootstrap sample (results$bres), the vector of the estimated wqs index (results$wqs), the vector containing the cutoffs used to determine the quantiles (results$q_i), the list of vectors containing the rows of the subjects included in each bootstrap dataset (results$bindex), the rows identifying the subjects used to estimate the weights in each bootstrap (results$tindex) and the rows identifying the subjects used to estimate the parameters of the final model (results$vindex).

2.2. Example 2

In the following code we run a logistic regression (family = binomial) to test the association between the exposure to the mixture and the outcome y_bin and we also add the covariate sex. Since the mixture concentrations in this example are already standardized we can also run a model without categorizing for quantiles (q = NULL) after checking that there were no skewed distributions. Furthermore we examined the ability of our model to predict the outcome on a third part of the dataset (pred = 0.3). As we see from the script below validation = 0.4; that means that the 30% of the data are used as test dataset, 40% for validation and the last 30% for prediction; the script to generate the additional plot is reported:

```
R> # we run the logistic model and save the results in the variable
R> # "results2"
R> results2 <- gwqs(y_bin ~ wqs + sex, mix_name = toxic_chems,
+                   data = wqs_data, q = NULL, validation = 0.4, b = 2,
+                   b1_pos = TRUE, b1_constr = FALSE, family = binomial,
+                   seed = 2018, plots = TRUE, tables = FALSE, pred = 0.3)
R> #
R> # plot ROC curve
```

```
R> gg_roc <- ggplot(results2$df_pred, aes(d=y, m=p_y)) + geom_roc(n.cuts = 0) +
+    style_roc(xlab = "1 - Specificity", ylab = "Sensitivity")
R> auc_est <- plotROC::calc_auc(gg_roc)
R> gg_roc + annotate("text", x=0.75, y=0.25,
+                    label=paste0("AUC = ", round(auc_est[, "AUC"], 3)))
R>
```

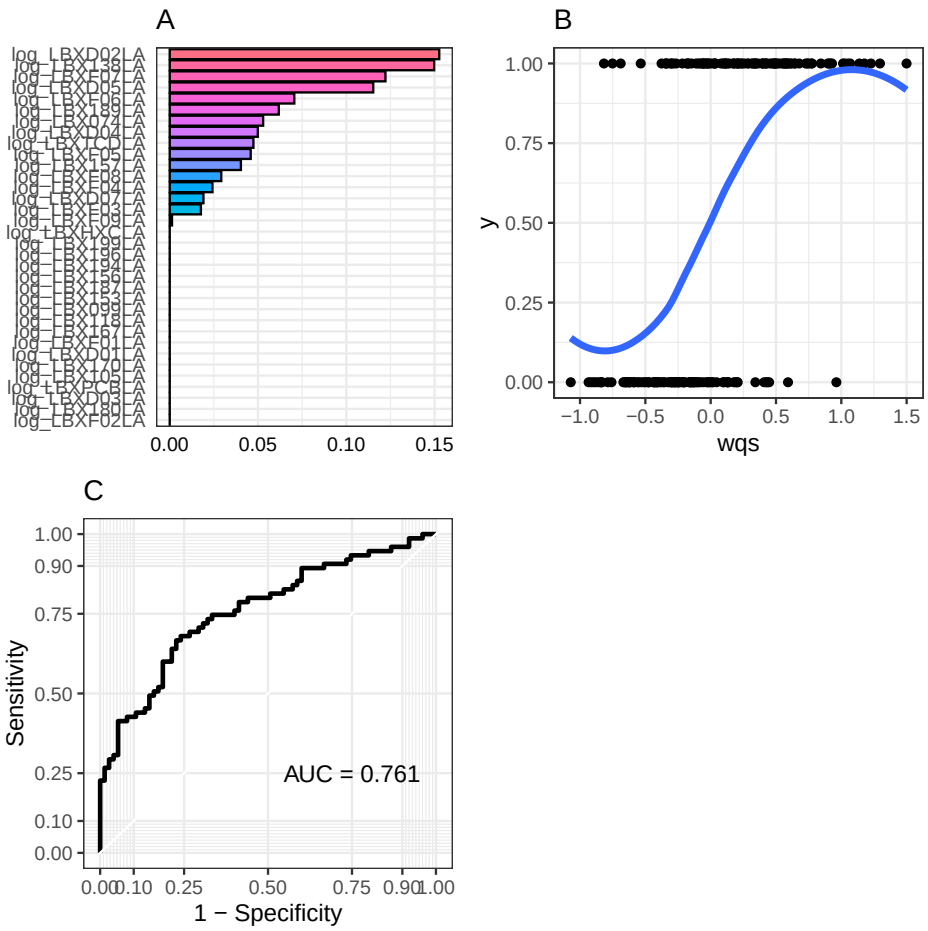


Figure 2: Plots displayed for binary outcomes when `plots = TRUE` and `pred > 0`

From figure 2 we see the per-variable calculated weights, ordered by relative magnitude. Plot B shows a positive relationship between the mixture and the outcome and as we can see from the following code it is statistically significant ($p<0.001$):

```
R> summary(results2$fit)

Call:
glm(formula = formula, family = family, data = bdtf)

Deviance Residuals:
```

```
    Min       1Q   Median       3Q      Max
-2.5127  -0.8162   0.1340   0.7440   2.3262

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1152     0.2540  -0.454    0.650
wqs           3.3557     0.4950   6.779 1.21e-11 ***
sex           0.2871     0.3513   0.817    0.414
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.24  on 199  degrees of freedom
Residual deviance: 198.48  on 197  degrees of freedom
AIC: 204.48

Number of Fisher Scoring iterations: 5
```

In plot C we show the Receiver Operating Characteristic (ROC) curve related to the predictive model: we can see that the cutoff that is closer to the left-hand border and the top border has around 70% sensitivity (true positive) and 30% specificity (false positive).
In this case two more parameters are returned by the `gwqs()` function: `df_pred`, which is a `data.frame` including a first column the actual value of the dependent variable and as a second column the predicted values; and `pindex`, the dataset rows identifying the observations used for prediction.
The `gwqs` function implements the `predict` function to run the predictive model. The following code shows how to reproduce the prediction:

```
R> # create a dataset exluding the data where we want to apply the prediction
R> # and define the group variable to identify the test and validation dataset
R> wqs_data$group <- 0
R> wqs_data$group[results2$vindex] <- 1
R> wqs_data_train <- wqs_data[-results2$pindex,]
R> # fit the model on the training dataset
R> results2_pred <- gwqs(y_bin ~ wqs + sex, mix_name = toxic_chems,
+                   data = wqs_data_train, q = NULL, validation = NULL,
+                   b = 2, valid_var = "group", b1_pos = TRUE,
+                   b1_constr = FALSE, family = binomial, seed = 2018)
R> # creat the dataset on which we apply the prediction
R> wqs_data_pred <- wqs_data[results2$pindex,]
R> # create wqs variable for the prediction dataset
R> mix_matrx <- as.matrix(wqs_data_pred[, rownames(results2$final_weights)])
R> wqs_data_pred$wqs <- as.numeric(mix_matrx%*%results2$final_weights$mean_weight)
R> # apply the predict() function
R> pred <- predict(results2$fit, newdata = wqs_data_pred, type = "response")
R> df_pred <- data.frame(y = wqs_data_pred$y_bin, p_y = pred)
```

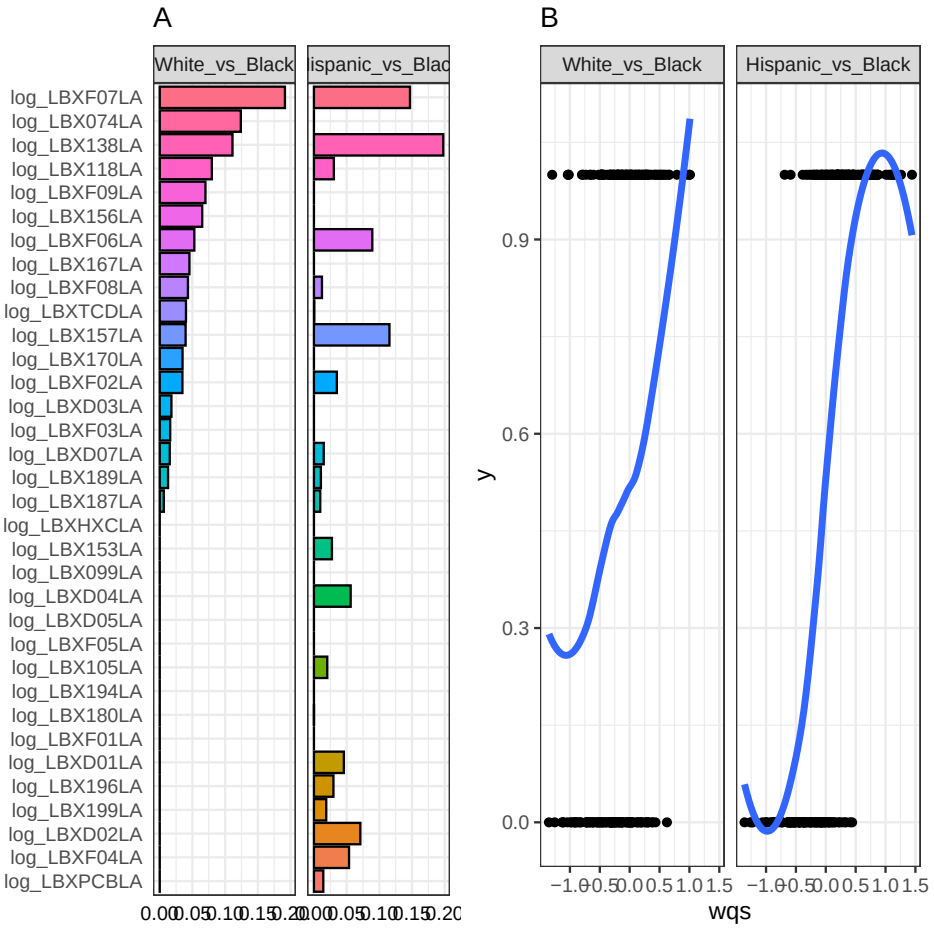
```
R> # plot the roc curve
R> gg_roc <- ggplot(df_pred, aes(d=y, m=p_y)) + geom_roc(n.cuts = 0) +
+    style_roc(xlab = "1 - Specificity", ylab = "Sensitivity")
R> auc_est <- plotROC::calc_auc(gg_roc)
R> gg_roc + annotate("text", x=0.75, y=0.25,
+                    label=paste0("AUC = ", round(auc_est[, "AUC"], 3)))
```

2.3. Example 3

In this third case we fit a multinomial model (`family = "multinomial"`) for categorical data: the outcome is `y_multinom`, representing the race of each subject. This modeling strategy creates a distinct logistic model comparing each level of the outcome variable to a reference level (in this case the "Black" category). We chose to create the training and validation dataset and assign to `valid_var` the name of the variable that identifies the two datasets (`valid_var = "group"`) and to use deciles in the estimate of the 'wqs' index (`q = 10`). In this case we had to choose two directions for each level of the outcome variable (in this case both positive: `b1_pos = c(TRUE, TRUE)`). We also decided to run the bootstrap in parallel on multiple cores (`plan_strategy = "multisession"`).

```
R> # we create the variable "group" in the dataset to identify the training
R> # and validation dataset: we choose 300 observations for the validation
R> # dataset and the remaining 200 for the training dataset
R> set.seed(123)
R> wqs_data$group <- 0
R> wqs_data$group[rownames(wqs_data) %in%
+                   sample(rownames(wqs_data), 300)] <- 1
R> #
R> # we run the logistic model and save the results in the variable
R> # "results3"
R> results3 <- gwqs(y_multinom ~ wqs, mix_name = toxic_chems,
+                   data = wqs_data, q = NULL, validation = 0.6,
+                   valid_var = "group", b = 2, b1_pos = c(TRUE, TRUE),
+                   b1_constr = FALSE, family = "multinomial", seed = 123,
+                   plots = TRUE, tables = TRUE,
+                   plan_strategy = "multiprocess")
R> #
R> # bar plot
R> data_plot <- results3$final_weights[order(results3$final_weights[, 2]),]
R> pos <- match(data_plot$mix_name, sort(data_plot$mix_name))
R> data_plot$mix_name <- factor(data_plot$mix_name,
+                                levels(data_plot$mix_name)[pos])
R> data_plot_l <- melt(data_plot, id.vars = "mix_name")
R> ggplot(data_plot_l, aes(x = mix_name, y = value, fill = mix_name)) +
+    facet_wrap(~ variable) + geom_bar(stat = "identity", color = "black") +
+    theme_bw() + theme(axis.ticks = element_blank(),
+                       axis.title = element_blank(),
+                       axis.text.x = element_text(color='black'),
```

```
+                         legend.position = "none") + coord_flip()
R> #
R> # scatter plot y vs wqs
R> ggplot(results3$y_wqs_df, aes(wqs, y)) +
+    geom_point() + stat_smooth(method = "loess", se = FALSE, size = 1.5) +
+    theme_bw() + facet_wrap(~ level)
R> #
R> # scatter plot of weights for the two levels of the dependent variable
R> ggplot(data_plot, aes_string(names(data_plot)[2], names(data_plot)[3])) +
+    geom_point() + theme_bw() + xlab(names(data_plot)[2]) +
+    ylab(names(data_plot)[3]) + geom_abline(linetype = 2) +
+    ggrepel::geom_text_repel(aes(label=mix_name))
R>
```



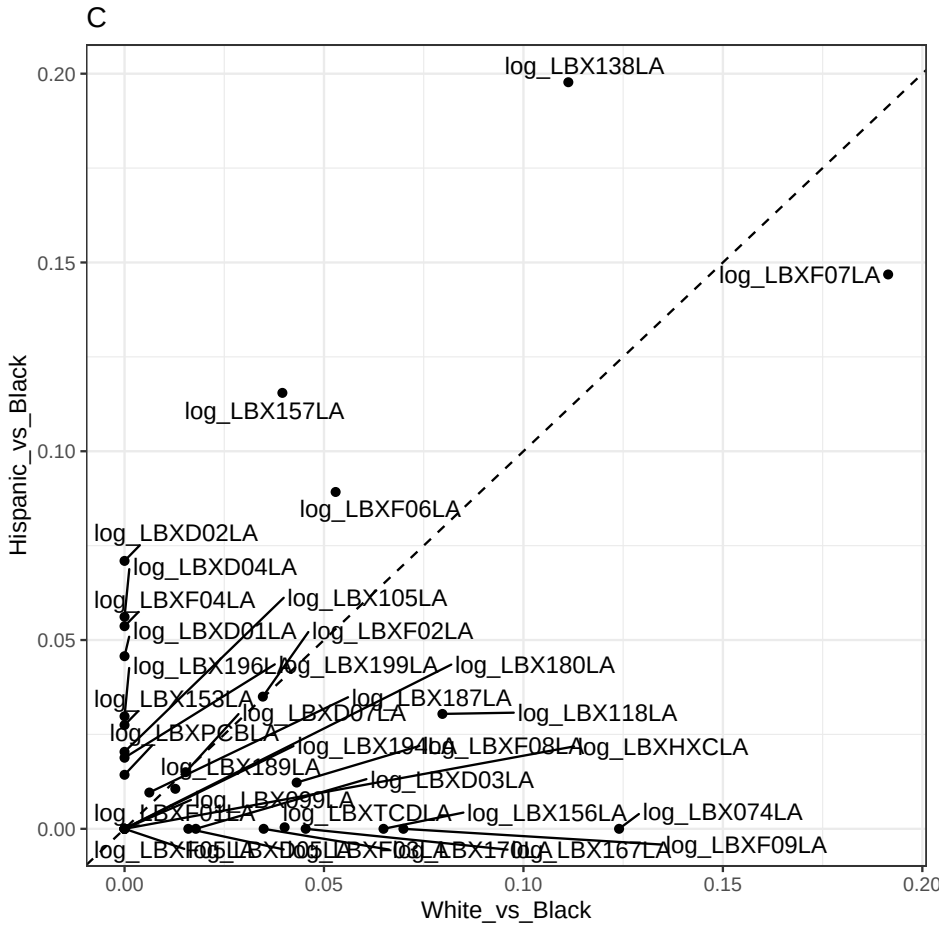


Figure 3: Plots displayed for multinomial outcomes when `plots = TRUE`

In Figure 3 while plots A and B are the same as in figure 1 and 2 but divided by the levels of the outcome variable, C is a scatter plot of the weights. This allows us to compare the magnitude of weights estimated in each model (e.g. "white vs black" or "hispanic vs black")"), with departures from the main diagonal indicating variables that are differentially-weighted for each comparison, e.g. Hispanic vs. Black, or White vs. Black. This is plotted only when the outcome has three levels.
In this case to look at the model results we do not need to use the `summary` function but instead use the following command:

```
R> results3$fit$sum_stat
```

```
                                Estimate Standard_Error      stat
(Intercept)_White_vs_Black    0.23054193      0.1643923 1.4023892
wqs_White_vs_Black            1.54411947      0.3524960 4.3805302
(Intercept)_Hispanic_vs_Black 0.05901184      0.1851028 0.3188058
wqs_Hispanic_vs_Black         4.31334550      0.4903692 8.7961175
                                  p_value
(Intercept)_White_vs_Black    1.607990e-01
```

```
wqs_White_vs_Black            1.183909e-05
(Intercept)_Hispanic_vs_Black 7.498738e-01
wqs_Hispanic_vs_Black         1.416313e-18
```

As we can see from the above results, both the wqs indices for each level are significant ($p < 0.001$), but, as shown from plot A and C in Figure 3, chemicals have different weights depending on the race.

2.4. Example 4

This last example shows how to fit the wqs on count data. The dependent variable taken into account is `y_count` and we fit a Poisson regression (`family = poisson`). We also run a stratified analysis by sex estimating different weights for males and females setting `stratified = "sex_factor"` (we created a new sex factor variable (`sex_factor`) since the previous one was numeric (0, 1)).

```
R> # we create the sex factor variable sex_factor
R> wqs_data$sex_factor <- factor(wqs_data$sex, labels = c("F", "M"))
R> #
R> # we run the poisson model and save the results in the variable
R> # "results4"
R> results4 <- gwqs(y_count ~ wqs, mix_name = toxic_chems,
+                   stratified = "sex_factor", data = wqs_data, q = 10,
+                   validation = 0.6, b = 2, b1_pos = TRUE,
+                   b1_constr = FALSE, family = poisson, seed = 123,
+                   plots = TRUE, tables = TRUE)
R>
```

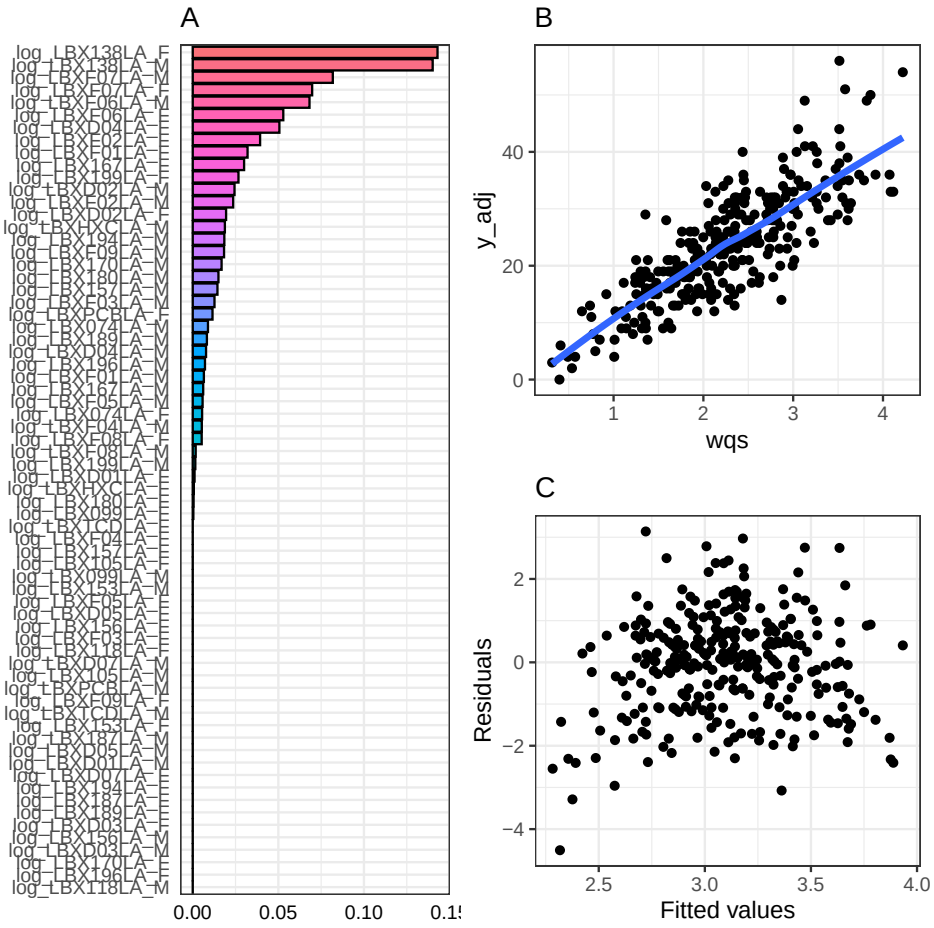


Figure 4: Plots displayed for count outcome when `plots = TRUE`

The results of the model are shown in the table below:

```
R> summary(results4$fit)

Call:
glm(formula = formula, family = family, data = bdtf)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.5053  -0.8377   0.0344   0.6967   3.1396

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.14946    0.04036   53.25   <2e-16 ***
wqs          0.42259    0.01539   27.47   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1191.69  on 299  degrees of freedom
Residual deviance:  436.11  on 298  degrees of freedom
AIC: 1908.2

Number of Fisher Scoring iterations: 4
```

We notice that there is a significant positive association between the wqs index and the dependent variable. Since we stratified by sex, we have an estimate of each weight for males and females and we can see how the weights differ between the two genders: we have a good agreement for some weights (e.g. `log_LBX138LA` has an high impact in both males and females) and differences for others (e.g. `log_LBXF07LA` is 4.5% for males being the 3rd highest weight while for females it has a lower impact (1.3% as the 12th highest weight)).
The following test allows us to test for overdisperions of the `y_count` data:

```
R> library(AER)
R> mean(wqs_data$y_count)

[1] 23.41

R> var(wqs_data$y_count)

[1] 87.2003

R> AER::dispersiontest(results4$fit)

        Overdispersion test

data:  results4$fit
z = 3.4542, p-value = 0.000276
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
  1.395178
```

Since the test indicates the data are overdispersed we fit a quasi-Poisson or a negative binomial regression (`family = "quasipoisson"` or `family = "negbin"` respectively):

```
R> # we run the quasi-poisson model and save the results in the variable
R> # "results5"
R> results5 <- gwqs(y_count ~ wqs, mix_name = toxic_chems,
+                   data = wqs_data, q = 10, validation = 0.6, b = 2,
+                   b1_pos = TRUE, b1_constr = FALSE, family = quasipoisson,
+                   seed = 123)
```

```
R> summary(results5$fit)

Call:
glm(formula = formula, family = family, data = bdtf)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.5254  -0.8962   0.0623   0.7294   2.7647

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.148395   0.045744   46.97   <2e-16 ***
wqs         0.210890   0.008671   24.32   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 1.309888)

    Null deviance: 1191.69  on 299  degrees of freedom
Residual deviance:  412.62  on 298  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4
```

A zero-inflated model can be fitted for the Poisson and negative binomial regression setting `zeroinfl = TRUE` and choosing a link function for the binomial process (we can choose among `"logit"`, `"probit"`, `"cloglog"`, `"cauchit"`, `"log"`). Here is shown the case of the negative binomial zero-inflated model using a `"logit"` link function for the binomial process (`zilink = "logit"`). To test the hypothesis that different covariates regulate the count and the binomial parts we write the formula as in the example below where the variables at the right of the symbol `"|"` are those included in the binomial process; otherwise we specify the formula in the usual way and all the variables will be included in both parts. Before fitting the model we generate a variable from a zero inflated negative binomial. In the following model `wqs` and `sex` are included in the count process while `new_var` is considered in the binomial process. Only the code for the residual vs fitted values scatter plot is reported since it is slightly different from the previous ones:

```
R> # generate new variable from normal distribution
R> set.seed(123)
R> wqs_data$new_var <- rnorm(500)
R> wqs_data$y_zinb <- rzinegbin(500, pstr0 = 0.3, mu = 3, size = 10)
R> #
R> # we run the zero-inflated negative binomial model and save the results in the variable
R> # "results6"
R> results6 <- gwqs(y_zinb ~ wqs + sex | new_var, mix_name = toxic_chems,
+                   data = wqs_data, q = 10, validation = 0.6, b = 2,
+                   zero_infl = TRUE, zilink = "logit", b1_pos = FALSE,
```

```
+                   b1_constr = FALSE, family = "negbin", seed = 1234,
+                   plots = TRUE, tables = TRUE)
R> #
R> # scatter plot residuals vs fitted values
R> fit_df <- data.frame(.fitted = results6$fit$fitted.values,
+                       .resid = results6$fit$residuals)
R> ggplot(fit_df, aes(x = .fitted, y = .resid)) + geom_point() +
+    theme_bw() + xlab("Fitted values") + ylab("Residuals")
R>
```

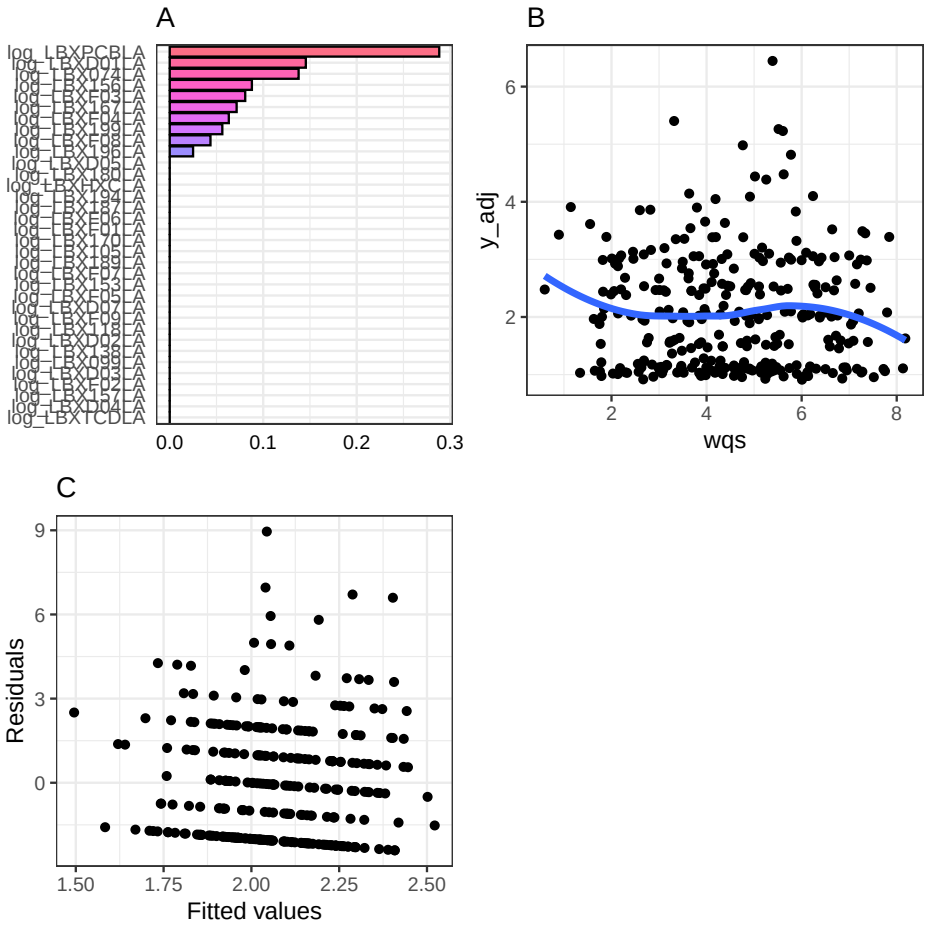


Figure 5: Plots displayed for count outcome when `plots = TRUE`

From the summary table below we note the estimate of the model; in this case the results related to both the count and the binomial process are presented.

```
R> summary(results6$fit)

Call:
zeroinfl(formula = ff, data = bdtf, dist = family$family, link = zilink$name)
```

```
Pearson residuals:
    Min      1Q  Median      3Q     Max
-1.1674 -0.9492 -0.1087  0.6255  4.4365

Count model coefficients (negbin with log link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.12152    0.14885   7.535 4.89e-14 ***
wqs         -0.01626    0.02898  -0.561    0.575
sex          0.04112    0.09642   0.426    0.670
Log(theta)   2.52240    0.64417   3.916 9.01e-05 ***

Zero-inflation model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.9087     0.1602  -5.672 1.41e-08 ***
new_var      -0.2871     0.1544  -1.860   0.0629 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 12.4585
Number of iterations in BFGS optimization: 21
Log-likelihood: -559.8 on 6 Df
```

3. Discussion

WQS regression is a new method that allows the investigation of the associations between mixtures of predictors and continuous, count, or categorical data. This approach is particularly robust against outliers and extreme values because of the ranking procedure used, and is additionally robust against collinearity through the constraints imposed during weight estimation and application of an ensemble estimation procedure. As well, the capacity for covariate adjustment and the simplicity of model interpretation are among the greatest strengths of this approach, and underlie its applicability to health-related research. Through the weighted index we are able to identify the combined impact of multiple predictors on a given outcome, while in the estimation of the weights we may simultaneously assess the discrete effects of contributing variables, with coadjustment for the overall mixture and relevant covariates.
The package **gWQS** provides a robust, generalizable implementation of this methodology in R extending the application of the model to continuous, binary, multinomial and count data applying the corresponding log-likelihood for each type of regression (zero inflated likelihoods are also available). The new version of the package allows also to run a stratified analysis for a categorial variable.
Future versions of the package will provide the ability to fit additional generalised linear models.

Acknowledgments

This package was developed at the CHEAR Data Center (Dept. of Environmental Medicine and Public Health, Icahn School of Medicine at Mount Sinai) with funding and support from NIEHS (U2C ES026555-01) with additional support from the Empire State Development Corporation.

References

Brunst KJ, Guerra MS, Gennings C, Hacker M, Jara C, Enlow MB, Wright RO, Baccarelli A, Wright RJ (2017). "Maternal Lifetime Stress and Prenatal Psychological Functioning and Decreased Placental Mitochondrial DNA Copy Number in the PRISM Study." *American Journal of Epidemiology*, **186**(11), 1227–1236. doi:10.1093/aje/kwx183.

Carrico C, Gennings C, Wheeler DC, Factor-Litvak P (2015). "Characterization of Weighted Quantile Sum Regression for Highly Correlated Data in a Risk Analysis Setting." *Journal of Agricultural Biological and Environmental Statistics*, **20**(1), 100–120. doi:10.1186/1476-069X-12-66.

Czarnota J, Gennings C, Wheeler DC (2015). "Assessment of Weighted Quantile Sum Regression for Modeling Chemical Mixtures and Cancer Risk." *Cancer Informatics*, **14**(2), 159–171. doi:10.4137/CIN.S17295.

Gennings C, Carrico C, Factor-Litvak P, Krigbaum N, Cirillo PM, Cohn BA (2013). "A Cohort study evaluation of maternal PCB exposure related to time to pregnancy in daughters." *Environmental Health*, **12**(1). doi:10.1186/1476-069X-12-66.

Ghalanos A, Theussl S (2015). ***Rsolnp**: General Non-linear Optimization Using Augmented Lagrange Multiplier Method*. R package version 1.16, URL https://cran.r-project.org/package=Rsolnp.

Hoerl AE, Kennard RW (1970). "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics*, **12**(1), 55–67. doi:10.1080/00401706.1970.10488634.

Horton MK, Blount BC, Valentin-Blasini L, Wapner R, Whyatt R, Gennings C, Factor-Litvak P (2015). "CO-occurring exposure to perchlorate, nitrate and thiocyanate alters thyroid function in healthy pregnant women." *Journal of Agricultural Biological and Environmental Statistics*, **143**, 1–9. doi:10.1016/j.envres.2015.09.013.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Tibshirani R (1996). "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society. Series B (Methodological)*, **58**(1), 267âĂŞ288.

Ye Y (1987). "Interior Algorithms for Linear, Quadratic, and Linearly Constrained Non-Linear Programming."

Zou H (2006). "The Adaptive Lasso and Its Oracle Properties." *Journal of the Royal Statistical Society. Series B (Methodological)*, **101**(476), 1418–1429. doi:10.1198/016214506000000735.

Zou H, Hastie T (2005). "Regularization and variable selection via the elastic net." *Journal of the Royal Statistical Society. Series B (Methodological)*, **67**(2), 301–320. doi:10.1111/j.1467-9868.2005.00503.x.

Affiliation:

Stefano Renzetti
Department of Occupational Health
University of Brescia
Piazzale Spedali Civili, 1, 25123 Brescia BS Italy
E-mail: stefano.renzetti@unibs.it

Chris Gennings
Department of Environmental Medicine and Public Health
Faculty in Biostatistics
Icahn School of Medicine at Mount Sinai
1 Gustave L. Levy Place New York, NY 10029
E-mail: chris.gennings@mssm.edu

Paul C. Curtin
Department of Environmental Medicine and Public Health
Faculty in Biostatistics
Icahn School of Medicine at Mount Sinai
1 Gustave L. Levy Place New York, NY 10029
E-mail: paul.curtin@mssm.edu