Heterogeneity in the relationship between BMI and risk biomarkers

---

Authors:

- Daniel E. Coral
- Femke Smit
- Elena Santos
- Ali Farzaneh

Introduction

This is a guideline to be followed by analysts in SOPHIA who are participating in the cross-sectional clustering project in the general population in Working Group 1. The idea is to standardize every step of the analysis across cohorts.

Not everyone has to follow all the steps, as we require different things from every cohort. We have divided the participating cohorts into 4 groups:

Cohort group	Cohorts
Discovery	UK Biobank
Validation	Maastricht

	Rotterdam
	GHS
Mental health	Girona
	Maastricht
Intervention	SCALE
	ABOS

This guideline is designed to be applied in discovery and validation cohorts. We will then apply what we learn in these cohorts to the mental health and intervention cohorts.

As a background, generally the relationship between BMI and multiple diseases is assumed to follow a continuum -- the higher the BMI, the higher the risk. However, it has also been found that in certain groups of people this relationship is disproportionally stronger or weaker for any given BMI. Our objective is to test the hypothesis that clustering-based approaches can be used to better capture these subgroups. An overview of the steps of our pipeline to test this hypothesis is shown in Figure 1.

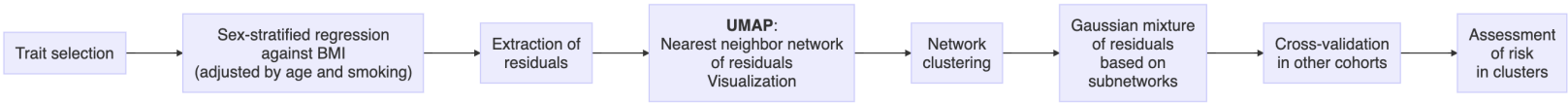

Figure 1. Pipeline overview

Software

All steps are intended to be followed in the R environment. To facilitate the analyses we have put together a list of functions that can be used to run every step of this guideline. They are located in the accompanying file `cross_sectional_FX.R`, which you can

load like this:

In [1]:
```
source("cross_sectional_FX.R")
```

These functions have dependencies on the following packages:

In [2]:
```
suppressMessages({
    library(tibble)
    library(readr)
    library(dplyr)
    library(tidyr)
    library(purrr)
    library(uwot)
    library(igraph)
    library(mvtnorm)
    library(survival)
})
```

The environment where discovery analysis in UK Biobank was executed is then the following:

In [3]:
```
sessionInfo()
```

```
R version 4.1.2 (2021-11-01)
Platform: x86_64-conda-linux-gnu (64-bit)
Running under: RHEL

Matrix products: default
BLAS/LAPACK: /gpfs/gpfs0/Home/daniel_c/miniconda3/envs/NewR/lib/libopenblasp-r0.3.18.so

locale:
[1] C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] survival_3.3-1   mvtnorm_1.1-3    igraph_1.3.0     uwot_0.1.11.9000
```

```
 [5] Matrix_1.4-1     purrr_1.0.1      tidyr_1.3.0      dplyr_1.1.1
 [9] readr_2.1.2      tibble_3.2.1

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.8.3     pillar_1.9.0     compiler_4.1.2   base64enc_0.1-3
 [5] tools_4.1.2      digest_0.6.31    uuid_1.0-4       jsonlite_1.8.4
 [9] evaluate_0.20    lifecycle_1.0.3  lattice_0.20-45  pkgconfig_2.0.3
[13] rlang_1.1.0      IRdisplay_1.1    cli_3.6.0        IRkernel_1.3
[17] fastmap_1.1.1    repr_1.1.4       generics_0.1.2   vctrs_0.6.1
[21] hms_1.1.1        grid_4.1.2       tidyselect_1.2.0 glue_1.6.2
[25] R6_2.5.1         fansi_1.0.4      pbdZMQ_0.3-7     tzdb_0.3.0
[29] magrittr_2.0.3   splines_4.1.2    ellipsis_0.3.2   htmltools_0.5.4
[33] utf8_1.2.3       crayon_1.5.1
```

Initial input

We have selected 10 traits, based on biological systems that are commonly affected by obesity:

- Blood pressure: SBP and DBP.
- Lipids: HDL, LDL, TG.
- Fat distribution: WHR.
- Glycemia: Fasting glucose.
- Liver metabolism: ALT.
- Kidney function: Creatinine.
- Inflammation: CRP.

The covariates that will be needed are:

- Sex.
- Age.
- Current smoking status, coded as 1 if current smoker, 0 otherwise.

The initial input table should look like the following:

In [4]:
```
recoded_dat <- read_tsv("../data/recoded_dat.tsv", show_col_types = FALSE)
head(recoded_dat)
```

A tibble: 6 × 15

eid	age	sex	bmi	whr	sbp	dbp	alt	scr	crp	hdl	tg	ldl	fg	smoking
<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1000039	44	Male	36.6959	0.9911504	124.5	64.5	34.97	93.0	3.60	1.158	2.800	3.956	5.427	0
1000071	67	Male	39.4807	0.8857143	179.5	103.0	46.74	68.7	9.41	1.372	1.127	2.311	7.079	0
1000088	60	Male	24.2786	0.8761905	152.0	89.0	13.14	80.6	1.20	0.983	1.590	4.200	5.401	0
1000096	41	Male	26.5744	0.9587629	143.0	90.0	30.32	80.1	6.13	1.041	2.713	4.029	4.239	0
1000109	62	Male	33.8719	1.0818182	156.5	104.5	16.26	89.3	14.42	0.890	2.437	3.525	6.100	0
1000117	47	Female	23.8408	0.7254902	147.5	84.0	14.07	61.0	0.24	1.972	0.591	2.252	4.395	0

The units of the contiuous variables we expect to run the analysis:

- Age in years.
- Sex as a string of either "Female" or "Male".
- BMI in kg/m2.
- WHR is unitless, calculated by dividing waist and hip circumferences measured in cm.
- SBP and DBP in mmHg.
- ALT in U/L.
- sCr in umol/L.
- CRP in mg/L.
- HDL in mmol/L.
- TG in mmol/L.
- LDL in mmol/L.

- FG in mmol/L.
- Smoking as a dummy variable: 1 if currently smoking, 0 otherwise.

Note:

For the functions included in the `cross_sectional_FX.R` to work, the input table should be ***exactly*** as shown above.

Missing data

Since our clustering method ignores individuals with missing values for any biomarker, the input data should only contain individuals who have all biomarker values. From the previous analyses, we know that:

- Only including complete cases without losing too much data is possible in UK Biobank, Maastricht, GHS and ABOS.
- In Rotterdam the initial input table contains some values have been imputed using a random forest algorithm.
- In Girona only a small subset of individuals have CRP values.
- In SCALE there are values for waist but not for hip circumference, so it is not possible to calculate WHR.

Based on these observations we have made the following decisions on how to deal with missing values:

How to handle missing values	Cohorts
Only include complete cases.	UK Biobank
	Maastricht
	GHS
	ABOS
Use data that has been imputed.	Rotterdam
Retain all individuals.	Girona

For the cohorts in the last group, to be able to apply our method, we will assume that BMI explains the variability in the biomarkers that are missing. This assumption is based on what we have observed in the other cohorts. In practice, this means that the clustering method will focus on the biomarkers that are available to group individuals into clusters. The input table should still have the same columns so that the functions in `cross_sectional_FX.R` work properly.

Remove possible errors in measurement

In discovery and validation cohorts we will exclude biomarker measurements that are 5 SD away from the mean, under the assumption that these are most likely measurement errors. This can be done using the `remove_outliers` function that we have provided, which replaces outliers with `NA` values. Then we again make sure to have only complete cases:

In [5]:
```
recoded_dat <- mutate(recoded_dat,
                      across(c(bmi, whr, sbp, dbp, alt, scr, crp, hdl, tg, ldl, fg),
                             ~remove_outliers(.x, sdunits = 5)))
recoded_dat <- recoded_dat[complete.cases(recoded_dat),]
```

Stratify by sex

All the pipeline is applied separately in each sex group. The functions we have designed work on a list containing two dataframes for each sex group, which we can obtain like this:

In [6]:
```
strat_dat <- split(recoded_dat, ~sex)
```

To see the first lines of the two elements in the list:

In [7]:
```
lapply(strat_dat, head)
```

$Female

A tibble: 6 × 15

eid	age	sex	bmi	whr	sbp	dbp	alt	scr	crp	hdl	tg	ldl	fg
<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1000117	47	Female	23.8408	0.7254902	147.5	84.0	14.07	61.0	0.24	1.972	0.591	2.252	4.395
1000132	43	Female	35.6559	0.8403361	137.0	100.5	18.89	60.5	4.31	1.236	2.037	3.686	5.214
1000176	69	Female	38.1271	0.8897638	137.5	93.5	36.39	68.9	3.69	1.601	1.988	4.551	4.266
1000223	63	Female	25.4603	0.7789474	163.0	94.0	6.10	67.1	1.29	1.453	2.829	3.491	5.876
1000282	48	Female	25.4297	0.7708333	135.5	89.0	9.63	46.2	0.16	2.185	0.722	3.584	5.212
1000367	42	Female	19.3280	0.6777778	107.0	72.5	9.34	57.1	0.69	2.346	0.395	3.072	4.649

$Male

A tibble: 6 × 15

eid	age	sex	bmi	whr	sbp	dbp	alt	scr	crp	hdl	tg	ldl	fg
<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1000039	44	Male	36.6959	0.9911504	124.5	64.5	34.97	93.0	3.60	1.158	2.800	3.956	5.427
1000071	67	Male	39.4807	0.8857143	179.5	103.0	46.74	68.7	9.41	1.372	1.127	2.311	7.079
1000088	60	Male	24.2786	0.8761905	152.0	89.0	13.14	80.6	1.20	0.983	1.590	4.200	5.401
1000096	41	Male	26.5744	0.9587629	143.0	90.0	30.32	80.1	6.13	1.041	2.713	4.029	4.239
1000109	62	Male	33.8719	1.0818182	156.5	104.5	16.26	89.3	14.42	0.890	2.437	3.525	6.100
1000125	66	Male	36.1100	1.0625000	155.0	102.5	25.59	88.7	1.91	1.061	1.320	2.538	4.531

Summary of initial input

We need a table summarising the initial input, which can be generated like this:

In [8]:
```
gendesc_tab <- get_general_descriptives(strat_dat)
gendesc_tab
```

A data.frame: 28 × 7

sex	Variable	Type	N	N_miss	Summary1	Summary2
<chr>	<chr>	<chr>	<int>	<int>	<chr>	<chr>
Female	bmi	Numeric	76051	0	27.04 (5.01)	26.14 (23.49 - 29.69)
Female	age	Numeric	76051	0	57.43 (7.85)	59 (51 - 64)
Female	smoking	Categorical	76051	0	0	69334 (91.17%)
Female	smoking	Categorical	76051	0	1	6717 (8.83%)
Female	whr	Numeric	76051	0	0.82 (0.07)	0.81 (0.77 - 0.86)
Female	sbp	Numeric	76051	0	137.47 (19.47)	135.5 (123 - 149.5)
Female	dbp	Numeric	76051	0	81.47 (9.91)	81 (74.5 - 88)
Female	alt	Numeric	76051	0	19.82 (9.26)	17.54 (13.95 - 22.92)
Female	scr	Numeric	76051	0	64.28 (10.58)	63.2 (57.2 - 70)
Female	crp	Numeric	76051	0	2.36 (2.71)	1.4 (0.67 - 2.91)
Female	hdl	Numeric	76051	0	1.61 (0.38)	1.57 (1.34 - 1.84)
Female	tg	Numeric	76051	0	1.53 (0.79)	1.33 (0.98 - 1.87)
Female	ldl	Numeric	76051	0	3.68 (0.87)	3.62 (3.06 - 4.24)
Female	fg	Numeric	76051	0	4.95 (0.54)	4.91 (4.62 - 5.21)
Male	bmi	Numeric	66341	0	27.84 (4.18)	27.32 (25.02 - 30.07)
Male	age	Numeric	66341	0	57.72 (8.03)	60 (52 - 64)
Male	smoking	Categorical	66341	0	0	58346 (87.95%)
Male	smoking	Categorical	66341	0	1	7995 (12.05%)

Male	whr	Numeric	66341	0	0.94 (0.06)	0.94 (0.89 - 0.98)
Male	sbp	Numeric	66341	0	142.28 (17.65)	140.5 (130 - 153)
Male	dbp	Numeric	66341	0	84.94 (9.98)	84.5 (78 - 91.5)
Male	alt	Numeric	66341	0	26.28 (11.64)	23.56 (18.31 - 31.23)
Male	scr	Numeric	66341	0	81.38 (12.68)	80.2 (72.9 - 88.3)
Male	crp	Numeric	66341	0	2.09 (2.43)	1.28 (0.67 - 2.5)
Male	hdl	Numeric	66341	0	1.3 (0.32)	1.26 (1.08 - 1.47)
Male	tg	Numeric	66341	0	1.88 (0.99)	1.64 (1.16 - 2.34)
Male	ldl	Numeric	66341	0	3.51 (0.86)	3.49 (2.9 - 4.08)
Male	fg	Numeric	66341	0	4.99 (0.6)	4.94 (4.63 - 5.27)

For smoking:

- `Summary1` contains the categories.
- `Summary2` contains the proportion of each category.

For the rest (continuous) variables:

- `Summary1` contains the mean and standard deviation.
- `Summary2` contains the median and interquartile range.

Estimates of BMI-biomarker associations

The first step of the pipeline is to generate sex-specific linear models of BMI for each variable, adjusting for age, and smoking. To do that we have the following function:

In [9]:
```
mods <- get_bmimods(strat_dat)
```

The result is a table with a column that contains the models specific for each sex and biomarker:

In [10]:
```
print(mods)
```

```
# A tibble: 20 x 3
   sex    Biomarker mod
   <chr>  <chr>     <list>
 1 Female whr       <lm>
 2 Female sbp       <lm>
 3 Female dbp       <lm>
 4 Female alt       <lm>
 5 Female scr       <lm>
 6 Female crp       <lm>
 7 Female hdl       <lm>
 8 Female tg        <lm>
 9 Female ldl       <lm>
10 Female fg        <lm>
11 Male   whr       <lm>
12 Male   sbp       <lm>
13 Male   dbp       <lm>
14 Male   alt       <lm>
15 Male   scr       <lm>
16 Male   crp       <lm>
17 Male   hdl       <lm>
18 Male   tg        <lm>
19 Male   ldl       <lm>
20 Male   fg        <lm>
```

As an example, we can print the summary of the female model for CRP:

In [11]:
```
summary(mods$mod[mods$sex == "Female" & mods$Biomarker == "crp"][[1]])
```

```
Call:
lm(formula = reformulate(response = biomarker, termlabels = c(BODYSIZEINDEX,
    COVARIATES)), data = X, na.action = na.exclude)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-6.5224 -1.2999 -0.5891  0.4163 16.8738

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.992832   0.079321  -62.94   <2e-16 ***
bmi          0.231435   0.001771  130.67   <2e-16 ***
age          0.018077   0.001134   15.94   <2e-16 ***
smoking      0.641565   0.031328   20.48   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.441 on 76047 degrees of freedom
Multiple R-squared:  0.1903,    Adjusted R-squared:  0.1903
F-statistic:  5957 on 3 and 76047 DF,  p-value: < 2.2e-16
```

We then use this table to generate a table containing the estimates of the effect of BMI on every biomarker:

In [12]:
```
bmicoefs_tab <- get_bmicoefs(mods)
bmicoefs_tab
```

A tibble: 20 × 6

sex	Biomarker	Estimate	SE	lowerCI	upperCI
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
Female	whr	0.00626	0.00004	0.00617	0.00634
Female	sbp	0.60205	0.01288	0.57680	0.62730
Female	dbp	0.56666	0.00687	0.55319	0.58013
Female	alt	0.46317	0.00646	0.45051	0.47583
Female	scr	0.22335	0.00761	0.20844	0.23827
Female	crp	0.23144	0.00177	0.22796	0.23491
Female	hdl	-0.02796	0.00025	-0.02846	-0.02747
Female	tg	0.04919	0.00053	0.04815	0.05023

Female	ldl	0.01041	0.00062	0.00919	0.01163
Female	fg	0.01019	0.00038	0.00944	0.01093
Male	whr	0.00926	0.00005	0.00917	0.00935
Male	sbp	0.54364	0.01572	0.51282	0.57446
Male	dbp	0.51112	0.00905	0.49338	0.52886
Male	alt	0.87115	0.01011	0.85134	0.89096
Male	scr	0.24886	0.01164	0.22605	0.27167
Male	crp	0.14446	0.00216	0.14022	0.14870
Male	hdl	-0.02494	0.00028	-0.02548	-0.02440
Male	tg	0.06482	0.00088	0.06309	0.06655
Male	ldl	-0.00427	0.00079	-0.00583	-0.00272
Male	fg	0.01482	0.00055	0.01374	0.01590

Generate table with residual values

The next step in the pipeline is to calculate for each individual how much they deviate from the expected given the BMI. To do that we have the following function:

In [13]:
```
residtab <- get_residtabs(strat_dat, mods)
```

In [14]:
```
lapply(residtab, head)
```

$Female

A tibble: 6 × 11

eid	whr	sbp	dbp	alt	scr	crp	hdl	tg
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>

1000117	-0.9418402	1.1938755	0.5159529	-0.3638692	-0.1724967	-0.4646662	0.8537702	-0.7958077	-1.3774491	-0.76972765
1000132	-0.1792318	0.4062052	1.5761452	-0.3892039	-0.4399567	0.1120275	-0.2644245	0.4751430	0.2437055	0.65593408
1000176	-0.1770372	-0.9768242	0.5273488	1.1364654	0.1018258	-0.5687282	0.6992130	-0.3596230	0.6253109	-1.80648726
1000223	-0.5698414	1.1949817	1.3753895	-1.5352973	0.2475230	-0.3065696	-0.6740994	1.7741363	-0.3194735	1.64990373
1000282	-0.3781328	0.4131752	0.9426128	-0.9573206	-1.6249405	-0.6554686	1.5799995	-0.7469813	0.1340274	0.73413804
1000367	-1.1562861	-0.6785952	-0.3972053	-0.6004841	-0.4084902	0.1845008	1.6168411	-0.6445614	-0.2512526	-0.08102865

$Male

A tibble: 6 × 11

eid	whr	sbp	dbp	alt	scr	crp	hdl	tg	ldl	fg
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1000039	-0.1434963	-0.8936365	-2.6219406	-0.2353730	0.85840152	0.2777242	0.31393849	0.30219615	0.2974455	0.7257948
1000071	-3.3845181	1.5170327	1.2684506	1.1383582	-1.37538141	2.3833741	1.14528403	-1.50513167	-1.1712396	3.0688036
1000088	-0.5801301	0.6085712	0.6033637	-0.8899948	-0.05064503	-0.1335900	-1.40276253	-0.02527338	0.8300850	0.7311632
1000096	1.1595253	0.6229005	0.5165162	0.0806867	0.06057083	2.0206701	-0.91346782	0.88210053	0.2743749	-0.9772074
1000109	1.6924144	0.5010277	1.6986367	-1.3267979	0.43268229	4.9303647	-0.92118662	0.22376839	0.1260865	1.6383562
1000125	0.4998882	0.2572578	1.4582960	-0.4119699	0.59514853	-1.0377080	-0.01753134	-1.26104815	-0.9108525	-1.0610050

Obtaining clusters and probabilities of allocation

We use this residuals to run UMAP, which we use not only to obtain a projection of residual data in 2 dimensions, but also to run a probabilistic network clustering algorithm. We have wrapped all the clustering steps in a single function that we apply to the `residtab` object:

In [15]:
```
cluster_results <- get_cluster_results(residtab)
```

```
Starting analysis for the Female subset:

        1. Setting parameters...

        2. Running UMAP...

        3. Extracting graph...

        4. Initializing graph clustering using the leading eigen vector method...

        5. Optimizing graph clustering using the Leiden algorithm...

           10 clusters found. Modularity = 0.65.

        6. Extracting cluster membership...

        7. Calculating eigen centrality in clusters...

        8. Calculating Gaussian subdistributions weighted by eigen centralities...

        9. Adding central/concordant subdistribution...

        10. Fitting a mixture of Gaussians with subdistributions found...

                Convergence reached in 34 iterations.

        11. Organizing results...

Done!

Starting analysis for the Male subset:

        1. Setting parameters...

        2. Running UMAP...

        3. Extracting graph...

        4. Initializing graph clustering using the leading eigen vector method...
```

```
       5. Optimizing graph clustering using the Leiden algorithm...

          9 clusters found. Modularity = 0.64.

       6. Extracting cluster membership...

       7. Calculating eigen centrality in clusters...

       8. Calculating Gaussian subdistributions weighted by eigen centralities...

       9. Adding central/concordant subdistribution...

       10. Fitting a mixture of Gaussians with subdistributions found...

               Convergence reached in 31 iterations.

       11. Organizing results...

Done!
```

The result is a list containing the UMAP model, the parameters of the clusters found, the modularity score, which is a measure of the quality of the network partition, and the allocation probabilities of individuals to all clusters:

In [16]:
```
lapply(cluster_results, names)
```

$Female 'umap_model' · 'probs' · 'clusters' · 'modularity'

$Male 'umap_model' · 'probs' · 'clusters' · 'modularity'

Table summarising clusters

To understand what characterizes the clusters found, we will need a table with descriptives of the distribution of residuals per cluster, which can be generated like this:

In [17]:
```
clustersummary <- get_clustersummary(cluster_results)
```

In [18]:
```
head(clustersummary)
```

A tibble: 6 × 6

sex	cluster	trait	center	SD	weight
<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
Female	cluster_1	whr	-0.3379310	0.7640368	0.08365583
Female	cluster_1	sbp	1.0019195	0.6992817	0.08365583
Female	cluster_1	dbp	1.0049908	0.6466798	0.08365583
Female	cluster_1	alt	-0.2244410	0.5424884	0.08365583
Female	cluster_1	scr	-0.2141406	0.7238875	0.08365583
Female	cluster_1	crp	-0.2891127	0.4198875	0.08365583

We also need the distribution of the biomarkers in the clusters in their natural scale. To calculate this for a specific cluster, we use all individuals, and weigh each individual by their corresponding cluster allocation probability. The function for this is the following:

In [19]:
```
cluster_descriptives <- get_cluster_descriptives(cluster_results, strat_dat)
```

In [20]:
```
head(cluster_descriptives)
```

A tibble: 6 × 9

sex	cluster	TotalN	Weighted_N	N80Perc	Variable	Type	Summary1	Summary2
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<chr>
Female	cluster_1	76051	6362.11	2899	bmi	Numeric	25.98 (4.42)	25.22 (22.9 - 28.22)
Female	cluster_1	76051	6362.11	2899	age	Numeric	57.52 (7.43)	58.98 (52.02 - 63.1)

Female	cluster_1	76051	6362.11	2899	smoking	Categorical	0	91.54 %
Female	cluster_1	76051	6362.11	2899	smoking	Categorical	1	8.46 %
Female	cluster_1	76051	6362.11	2899	whr	Numeric	0.79 (0.06)	0.79 (0.75 - 0.83)
Female	cluster_1	76051	6362.11	2899	sbp	Numeric	151.01 (15.69)	149.85 (139.75 - 161.01)

Relationship between BMI and biomarkers in each cluster

Given that the relationship between BMI and biomarkers is expected to be different in these clusters compared to the overall relationship found above, we will quantify how much it changed by comparing models weighted by probabilities of each cluster. The function is the following:

In [21]:
```
clusbmicoefs <- get_bmicoefs_clusters(cluster_results, strat_dat)
```

In [22]:
```
head(clusbmicoefs)
```

A tibble: 6 × 7

sex	cluster	Biomarker	Estimate	SE	lowerCI	upperCI
<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
Female	cluster_1	whr	0.00553	0.00004	0.00545	0.00561
Female	cluster_1	sbp	0.54778	0.01102	0.52618	0.56937
Female	cluster_1	dbp	0.57378	0.00546	0.56308	0.58448
Female	cluster_1	alt	0.26102	0.00393	0.25332	0.26871
Female	cluster_1	scr	0.15787	0.00665	0.14483	0.17092
Female	cluster_1	crp	0.15486	0.00074	0.15342	0.15630

Disease prevalence

The next step is to check whether the prevalence of certain diseases differ in the clusters found, which would give us an initial idea of the clinical relevance of these subgroups. To do that we expect a table where each column represents a disease, coded as 1 or 0 depending on the presence or absence of the disease for each individual ***included in the clustering analysis***:

In [23]:
```
disease_dat <- inner_join(read_tsv("../data/selected_dx.tsv", show_col_types = FALSE),
                          read_tsv("../data/anycadat.tsv", show_col_types = FALSE),
                          by = "eid") %>%
    ## Only people who were included in the clustering analysis
    filter(eid %in% recoded_dat$eid)
head(disease_dat)
```

A tibble: 6 × 22

eid	HT	CHD	Stroke	T2D	T1D	Hypothyroidism	Hyperthyroidism	Asthma_COPD	Sleep_Apnea	⋯	Liver_disease	CKD	C
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	⋯	<dbl>	<dbl>	
1000039	0	0	0	0	0	0	0	0	0	⋯	0	0	
1000071	1	0	0	1	0	0	0	0	0	⋯	0	0	
1000096	0	0	0	0	0	0	0	1	0	⋯	0	0	
1000109	1	0	0	0	0	0	0	1	0	⋯	0	0	
1000117	0	0	0	0	0	0	0	0	0	⋯	0	0	
1000125	0	0	0	0	0	0	0	0	0	⋯	0	0	

Diseases in this table are commonly associated with obesity; the list includes:

In [24]:
```
colnames(disease_dat)[-1]
```

'HT' · 'CHD' · 'Stroke' · 'T2D' · 'T1D' · 'Hypothyroidism' · 'Hyperthyroidism' · 'Asthma_COPD' · 'Sleep_Apnea' · 'IBS' · 'IBD' · 'Liver_disease' · 'CKD' · 'Osteoarthrosis' · 'Osteoporosis' · 'Lumbar_pain' · 'Depression' · 'PCOS' · 'BPH' · 'RA' · 'AnyCancer'

Note: We don't expect to have all diagnoses available in all validation cohorts, but the more the better!

The first thing we need is a table summarising the sex-specific prevalence of each disease:

In [25]:
```
overall_diseasesum <- get_diseasesummary(disease_dat, strat_dat)
```

In [26]:
```
head(overall_diseasesum)
```

A data.frame: 6 × 6

	sex	Disease	N	N_miss	Summary1	Summary2
	<chr>	<chr>	<int>	<int>	<chr>	<chr>
1	Female	HT	76051	0	0	57600 (75.74%)
2	Female	HT	76051	0	1	18451 (24.26%)
3	Female	CHD	76051	0	0	73935 (97.22%)
4	Female	CHD	76051	0	1	2116 (2.78%)
5	Female	Stroke	76051	0	0	74953 (98.56%)
6	Female	Stroke	76051	0	1	1098 (1.44%)

Then, we will need a table summarising the prevalence within each cluster, which is again calculated using the cluster probabilites as weights:

In [27]:
```
cluster_diseasesum <- get_diseasesummary_clusters(disease_dat, cluster_results)
```

In [28]:
```
head(cluster_diseasesum)
```

A tibble: 6 × 5

sex	cluster	disease	Summary1	Summary2
<chr>	<chr>	<chr>	<chr>	<chr>
Female	cluster_1	HT	0	74.02 %

Female	cluster_1	HT	1	25.98 %
Female	cluster_1	CHD	0	99.05 %
Female	cluster_1	CHD	1	0.95 %
Female	cluster_1	Stroke	0	99.29 %
Female	cluster_1	Stroke	1	0.71 %

Some of the prevalences may be affected by subtle residual differences in the covariates we adjusted for before clustering (i.e., age, smoking and most importantly, BMI), and also certain medications. To control for that:

- We will treat 'cluster 0', the cluster characterized by all biomarkers being aligned with BMI, as the reference cluster.
- For each disease, we will calculate the increase in disease OR that corresponds to a unit increase in cluster allocation probability of every cluster relative to 'cluster 0', while all other covariates remain equal.

To do that we need an additional table containing medication data, which should look like this:

In [29]:
```
meds_dat <- read_tsv("../data/selected_meds.tsv", show_col_types = FALSE)
head(meds_dat)
```

A tibble: 6 × 5

eid	Insulin	AntiDM	AntiHT	LipidLower
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1000064	0	1	1	0
1000071	0	0	1	1
1000109	0	0	1	0
1000125	0	0	0	1
1000132	0	0	1	0
1000148	0	0	1	0

We defined people receiving medications using the following ATC codes:

Medication group	ATC codes
Insulin	A10A
Other antidiabetic	A10B
Antihypertensives	C01
	C02
	C03
	C07
	C08
	C09
Lipid lowering	C10

The function to run an adjusted analysis is the following:

In [30]:
```
cluster_adjdiseasesum <- get_adjdiseasesummary_clusters(cluster_results, disease_dat, meds_dat, strat_dat)
```

In [31]:
```
head(cluster_adjdiseasesum)
```

A tibble: 6 × 8

sex	disease	model	cluster	Estimate	SE	lowerCI	upperCI
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
Female	HT	Unadjusted	cluster_1	0.05022	0.00405	0.04227	0.05817
Female	HT	Unadjusted	cluster_10	0.02605	0.00246	0.02123	0.03087
Female	HT	Unadjusted	cluster_2	-0.00345	0.00422	-0.01172	0.00482
Female	HT	Unadjusted	cluster_3	-0.11688	0.00389	-0.12451	-0.10925
Female	HT	Unadjusted	cluster_4	-0.04836	0.00253	-0.05332	-0.0434

Female	HT	Unadjusted	cluster_5	-0.01054	0.00219	-0.01483	-0.00625

These latter estimates represent the change in the logarithm of the OR for a disease per each unit change in the logarithm of cluster probability. The rationale for using this transformation is to back-transform the estimates so that they represent the change in the OR for a certain percent increase/decrease in probability.

As a complementary analysis, we would calculate these OR by allocating individuals to a certain cluster based on different thresholds. The function for that is the following:

In [32]:
```
cluster_adjdiseasesum_thresh <- get_adjdiseasesummary_clustersthresh(cluster_results, disease_dat, meds_dat,
```

```
Warning message:
"There was 1 warning in `dplyr::transmute()`.
i In argument: `data = purrr::map(...)`.
i In group 19: `disease = "T1D"`.
Caused by warning:
! glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"There was 1 warning in `dplyr::transmute()`.
i In argument: `data = purrr::map(...)`.
i In group 19: `disease = "T1D"`.
Caused by warning:
! glm.fit: fitted probabilities numerically 0 or 1 occurred"
```

In [33]:
```
head(cluster_adjdiseasesum_thresh)
```

A tibble: 6 × 11

sex	disease	thresh	cluster	N	model	Estimate	SE	lowerCI	upperCI	N0
<chr>	<chr>	<dbl>	<chr>	<int>	<chr>	<chr>	<chr>	<chr>	<chr>	<int>
Female	HT	0.6	cluster_1	4555	Unadjusted	-0.24057	0.03847	-0.31597	-0.16518	12023
Female	HT	0.6	cluster_1	4555	Adjusted	0.75167	0.05476	0.64435	0.85899	12023
Female	HT	0.6	cluster_10	1466	Unadjusted	0.31897	0.05677	0.20769	0.43024	12023
Female	HT	0.6	cluster_10	1466	Adjusted	0.09294	0.09433	-0.09195	0.27783	12023

Female	HT	0.6	cluster_2	5671	Unadjusted	0.22614	0.03361	0.16027	0.29201	12023
Female	HT	0.6	cluster_2	5671	Adjusted	1.24398	0.04878	1.14837	1.33958	12023

These estimates represent the change in the logarithm of the OR for change in cluster membership relative to cluster 0 - the concordant cluster.

MACE incidence - UK Biobank, Rotterdam (GHS?)

Next, we will calculate the risk of future disease given the probability of cluster allocation. The table we expect to able to run this part looks like the following:

In [34]:
```
survmacedat <- read_tsv("../data/survmacedat.tsv", show_col_types = FALSE)
head(survmacedat)
```

A tibble: 6 × 5

eid	outcome_value	outcome_date	date0	outcome_timeyrs
<dbl>	<dbl>	<date>	<date>	<dbl>
1000071	0	2018-02-28	2008-01-29	10.083504
1000223	1	2016-07-23	2009-09-07	6.874743
1000324	1	2010-10-20	2007-09-13	3.101985
1000583	1	2012-05-31	2008-08-26	3.761807
1001175	1	2013-11-13	2009-04-30	4.539357
1001892	1	2017-07-10	2008-05-03	9.185489

The first thing we need, as done before, is a sex-specific summary of the data available:

In [35]:
```
macesum <- get_incidencesummary(survmacedat, strat_dat)
```

In [36]:
```
macesum
```

A data.frame: 2 × 10

sex	ntotal	ncases	personyrs	casesper1e5py	avgtime	sdtime	medtime	timeq25	timeq75
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Female	74423	3357	655191.8	512.369	8.80362	1.287503	8.941821	8.098563	9.604381
Male	61821	5827	530647.4	1098.093	8.58361	1.693026	8.867899	7.994524	9.582478

And again, we would run similar cluster-specific analyses as done for prevalence:

- Crude incidence per cluster:

In [37]:
```
cluster_macesum <- get_incidencesummary_clusters(survmacedat, cluster_results)
```

In [38]:
```
cluster_macesum
```

A tibble: 21 × 5

sex	cluster	ncases	personyrs	casesper1e5py
<chr>	<chr>	<dbl>	<dbl>	<dbl>
Female	cluster_0	1016.8976	165676.03	613.7868
Female	cluster_1	216.3929	55657.30	388.7951
Female	cluster_10	113.4984	15553.85	729.7129
Female	cluster_2	371.3930	66912.76	555.0406
Female	cluster_3	141.4161	47514.83	297.6251
Female	cluster_4	359.6983	57793.35	622.3871
Female	cluster_5	230.1955	33007.08	697.4126

Female	cluster_6	293.9397	77784.58	377.8893
Female	cluster_7	173.9839	30121.80	577.6012
Female	cluster_8	250.7350	44534.28	563.0157
Female	cluster_9	188.8497	60635.97	311.4483
Male	cluster_0	1471.3207	113200.35	1299.7493
Male	cluster_1	370.3560	30822.99	1201.5578
Male	cluster_2	255.2610	15727.28	1623.0462
Male	cluster_3	573.2952	56148.49	1021.0340
Male	cluster_4	362.4407	24275.36	1493.0397
Male	cluster_5	552.9871	51275.53	1078.4619
Male	cluster_6	540.4297	58909.24	917.3938
Male	cluster_7	510.5828	67593.03	755.3779
Male	cluster_8	544.5444	52714.03	1033.0162
Male	cluster_9	645.7823	59981.08	1076.6433

- Unadjusted and adjusted survival models with cluster probabilities as predictors:

In [39]:
```
cluster_adjmacesum <- get_adjincidencesummary_clusters(cluster_results, survmacedat, meds_dat, strat_dat)
```

In [40]:
```
head(cluster_adjmacesum)
```

A tibble: 6 × 7

sex	model	cluster	Estimate	SE	lowerCI	upperCI
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
Female	Unadjusted	cluster_1	0.0028	1.0028	-0.01114	0.01674
Female	Unadjusted	cluster_10	0.01399	1.01409	0.00565	0.02233

Female	Unadjusted	cluster_2	-0.00359	0.99642	-0.01776	0.01058
Female	Unadjusted	cluster_3	-0.04377	0.95718	-0.05695	-0.03059
Female	Unadjusted	cluster_4	-0.01809	0.98207	-0.02705	-0.00914
Female	Unadjusted	cluster_5	0.00129	1.00129	-0.00626	0.00885

- Unadjusted and adjusted survival models with hard cluster allocation at different thresholds as predictor:

In [41]:
```
cluster_adjmacesum_thresh <- get_adjincidencesummary_clustersthresh(cluster_results, survmacedat, meds_dat,
```

In [42]:
```
head(cluster_adjmacesum_thresh)
```

A data.frame: 6 × 10

	sex	thresh	cluster	N	model	Estimate	SE	lowerCI	upperCI	N0
	<chr>	<dbl>	<chr>	<int>	<chr>	<chr>	<chr>	<chr>	<chr>	<int>
1	Female	0.6	cluster_1	4526	Unadjusted	-0.63645	0.52917	-0.81404	-0.45887	11671
2	Female	0.6	cluster_1	4526	Adjusted	-0.3008	0.74023	-0.48137	-0.12023	11671
3	Female	0.6	cluster_10	1394	Unadjusted	0.14008	1.15037	-0.07304	0.3532	11671
4	Female	0.6	cluster_10	1394	Adjusted	-0.04554	0.95549	-0.26623	0.17516	11671
5	Female	0.6	cluster_2	5598	Unadjusted	-0.19726	0.82098	-0.33484	-0.05967	11671
6	Female	0.6	cluster_2	5598	Adjusted	0.09905	1.10412	-0.04171	0.2398	11671

We want to know whether a simple survival model using all biomarkers is able to capture the risk we observe in the clusters. To do that we first construct this simple survival model:

In [43]:
```
globalmod_mace <- global_survivalmodel(strat_dat, survmacedat)
```

The result is a sex specific survival model:

In [44]:
```
lapply(globalmod_mace, class)
```

$Female 'coxph'

$Male 'coxph'

Each model includes all variables, e.g.:

In [64]:
```
summary(globalmod_mace$Female)$coefficients
```

A matrix: 13 × 5 of type dbl

	coef	exp(coef)	se(coef)	z	Pr(>|z|)
bmi	1.465949e-02	1.0147675	0.004098008	3.57722405	3.472625e-04
age	8.353594e-02	1.0871243	0.003111155	26.85045928	8.331066e-159
smoking	6.533949e-01	1.9220549	0.051945413	12.57849080	2.772765e-36
whr	1.894749e+00	6.6508785	0.287375543	6.59328533	4.301986e-11
sbp	1.035408e-02	1.0104079	0.001188302	8.71334232	2.950440e-18
dbp	-4.933364e-03	0.9950788	0.002361447	-2.08912759	3.669624e-02
alt	-4.231542e-05	0.9999577	0.001875557	-0.02256152	9.820000e-01
scr	8.148241e-03	1.0081815	0.001441293	5.65342475	1.572820e-08
crp	2.920288e-02	1.0296335	0.005913643	4.93822221	7.883798e-07
hdl	-3.248821e-01	0.7226125	0.057609680	-5.63936707	1.706763e-08
tg	7.214975e-02	1.0748163	0.023871679	3.02239941	2.507794e-03
ldl	-1.034617e-01	0.9017106	0.020658551	-5.00817608	5.494825e-07
fg	5.411103e-02	1.0556018	0.029918077	1.80863987	7.050697e-02

We then extract the probability assigned by these models to each individual to be free of MACE events after 5 years of follow-up, and then we assess both the overall and the cluster-specific predictive performance. For the first we have the following function:

In [45]:
```
global_survmetrics_mace <- global_survmetrics(globalmod_mace, strat_dat, survmacedat)
```

This function returns for each sex a list containing 2 elements:

- A table containing multiple thresholds of survival probabilities, and their corresponding number of true/false positives and true/false negatives, which we can use to calculate ROC curves.
- A calculation of the variance parameters around the ROC AUC estimate, which we will use to formally compare 2 ROC AUCs using Delong's method.

For example, these is how this object looks like for females:

In [46]:
```
str(global_survmetrics_mace$Female)
```

```
List of 2
 $ roctab   :'data.frame':      74424 obs. of  5 variables:
  ..$ thresholds: num [1:74424] 0.999 0.999 0.999 0.999 0.999 ...
  ..$ TN        : num [1:74424] 0 1 2 3 4 5 6 7 8 9 ...
  ..$ TP        : num [1:74424] 1591 1591 1591 1591 1591 ...
  ..$ FP        : num [1:74424] 72832 72831 72830 72829 72828 ...
  ..$ FN        : num [1:74424] 0 0 0 0 0 0 0 0 0 0 ...
 $ rocaucvar:List of 5
  ..$ theta: num 0.726
  ..$ X    : num [1:1591] 0.956 0.507 0.484 0.847 0.271 ...
  ..$ n    : num 72832
  ..$ Y    : num [1:72832] 0.488 0.263 0.957 0.981 0.263 ...
  ..$ m    : num 1591
```

We will also calculate these same estimates for each cluster. We will do this by weighting individuals by their cluster probabilities:

In [47]:
```
cluster_survmetrics_mace <- cluster_survmetrics(globalmod_mace, strat_dat, survmacedat, cluster_results)
```

The result is similar as before, for each cluster identified:

In [48]:
```
str(cluster_survmetrics_mace$Female$cluster_1)
```

```
List of 2
 $ roctab   :'data.frame':      74424 obs. of  5 variables:
  ..$ thresholds: num [1:74424] 0.999 0.999 0.999 0.999 0.999 ...
  ..$ TN        : num [1:74424] 0 0.000194 0.001617 0.001621 0.001621 ...
  ..$ TP        : num [1:74424] 99.8 99.8 99.8 99.8 99.8 ...
  ..$ FP        : num [1:74424] 6218 6218 6218 6218 6218 ...
  ..$ FN        : num [1:74424] 0 0 0 0 0 0 0 0 0 0 ...
 $ rocaucvar:List of 5
  ..$ theta: num 0.662
  ..$ X    : num [1:1591] 7.04e-22 1.40e-04 1.83e-14 2.38e-09 3.56e-04 ...
  ..$ n    : num 6218
  ..$ Y    : num [1:72832] 9.47e-08 1.31e-04 1.77e-12 6.78e-06 1.51e-03 ...
  ..$ m    : num 99.8
```

Diabetes risk - UK Biobank, Rotterdam (GHS?)

Given the strong link between BMI and T2D, we will also test whether one of the clusters identified is associated with higher or lower risk of developing diabetes over time. The data we need looks just like the one we used above for MACE:

In [49]:
```
survdmdat <- read_tsv("../data/survdmdat.tsv", show_col_types = FALSE)
head(survdmdat)
```

A tibble: 6 × 5

eid	outcome_value	outcome_date	date0	outcome_timeyrs
<dbl>	<dbl>	<date>	<date>	<dbl>
1000109	1	2013-02-07	2009-07-03	3.600274
1000132	1	2011-01-01	2009-10-02	1.248460
1004267	1	2012-10-02	2008-03-15	4.550308
1006281	1	2011-08-04	2009-08-19	1.957563
1007454	0	2018-02-28	2008-09-26	9.423682

1010295	1	2016-05-23	2009-07-16	6.852841

We will then apply the same functions as we did for MACE:

- Sex-specific summary of the data available:

In [50]:
```
timetodmsum <- get_incidencesummary(survdmdat, strat_dat)
```

In [51]:
```
timetodmsum
```

A data.frame: 2 × 10

sex	ntotal	ncases	personyrs	casesper1e5py	avgtime	sdtime	medtime	timeq25	timeq75
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Female	34221	1104	301498.1	366.1715	8.810324	1.215402	8.873374	8.208077	9.607118
Male	28563	1486	249099.8	596.5481	8.721065	1.403724	8.856947	8.120465	9.593429

- Crude incidence per cluster:

In [52]:
```
cluster_timetodmsum <- get_incidencesummary_clusters(survdmdat, cluster_results)
```

In [53]:
```
head(cluster_timetodmsum)
```

A tibble: 6 × 5

sex	cluster	ncases	personyrs	casesper1e5py
<chr>	<chr>	<dbl>	<dbl>	<dbl>
Female	cluster_0	385.29157	76258.237	505.24584
Female	cluster_1	27.27165	26014.757	104.83147
Female	cluster_10	72.80332	5560.017	1309.40821

Female	cluster_2	69.80268	31296.491	223.03675
Female	cluster_3	15.59357	21176.473	73.63631
Female	cluster_4	176.87306	27198.234	650.31082

- Unadjusted and adjusted survival models with cluster probabilities as predictors (here ignoring medication):

In [54]:
```
cluster_adjtimetodmsum <- get_adjincidencesummary_clusters(cluster_results, survdmdat, meds_dat["eid"], str
```

In [55]:
```
head(cluster_adjtimetodmsum)
```

A tibble: 6 × 7

sex	model	cluster	Estimate	SE	lowerCI	upperCI
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
Female	Unadjusted	cluster_1	-0.03227	0.96824	-0.05367	-0.01087
Female	Unadjusted	cluster_10	0.09617	1.10095	0.08427	0.10807
Female	Unadjusted	cluster_2	-0.01822	0.98194	-0.03902	0.00258
Female	Unadjusted	cluster_3	-0.03686	0.96382	-0.0585	-0.01521
Female	Unadjusted	cluster_4	-0.00297	0.99703	-0.01634	0.0104
Female	Unadjusted	cluster_5	0.01482	1.01493	0.00274	0.0269

- Unadjusted and adjusted survival models with hard cluster allocation at different thresholds as predictor:

In [56]:
```
cluster_adjtimetodmsum_thresh <- get_adjincidencesummary_clustersthresh(cluster_results, survdmdat, meds_dat
```

In [57]:
```
head(cluster_adjtimetodmsum_thresh)
```

A data.frame: 6 × 10

sex	thresh	cluster	N	model	Estimate	SE	lowerCI	upperCI	N0

	<chr>	<dbl>	<chr>	<int>	<chr>	<chr>	<chr>	<chr>	<chr>	<int>
1	Female	0.6	cluster_1	2105	Unadjusted	-2.03554	0.13061	-2.53826	-1.53282	5349
2	Female	0.6	cluster_1	2105	Adjusted	-1.39179	0.24863	-1.89798	-0.88561	5349
3	Female	0.6	cluster_10	494	Unadjusted	0.86534	2.37582	0.59402	1.13667	5349
4	Female	0.6	cluster_10	494	Adjusted	1.27246	3.56961	0.99843	1.54648	5349
5	Female	0.6	cluster_2	2574	Unadjusted	-1.07687	0.34066	-1.37346	-0.78029	5349
6	Female	0.6	cluster_2	2574	Adjusted	-0.41715	0.65892	-0.72073	-0.11357	5349

- Overall survival model:

In [58]:
```
globalmod_timetodm <- global_survivalmodel(strat_dat, survdmdat)
```

In [59]:
```
lapply(globalmod_timetodm, class)
```

$Female 'coxph'

$Male 'coxph'

- Overall predictive performance at 5 years:

In [60]:
```
global_survmetrics_timetodm <- global_survmetrics(globalmod_timetodm, strat_dat, survdmdat)
```

- Cluster-specific predictive performance at 5 years, weighted by probabilities:

In [61]:
```
cluster_survmetrics_timetodm <- cluster_survmetrics(globalmod_timetodm, strat_dat, survdmdat, cluster_result
```

Gathering and saving results

We'll save all results in a single file:

In [66]:
```
result_file <- list(
    General_descriptives = gendesc_tab,
    BMI_coefficients = bmicoefs_tab,
    Cluster_results = cluster_results,
    Cluster_summary = clustersummary,
    Cluster_descriptives = cluster_descriptives,
    Cluster_BMI_coefficients = clusbmicoefs,
    Disease_summary = overall_diseasesum,
    Cluster_Disease_summary = cluster_diseasesum,
    Cluster_Disease_summary_Adjusted = cluster_adjdiseasesum,
    Cluster_Disease_summary_Threshold = cluster_adjdiseasesum_thresh,
    MACE_summary = macesum,
    Cluster_MACE_summary = cluster_macesum,
    Cluster_MACE_summary_Adjusted = cluster_adjmacesum,
    Cluster_MACE_summary_Threshold = cluster_adjmacesum_thresh,
    Global_SurvMetrics_MACE = global_survmetrics_mace,
    Cluster_SurvMetrics_MACE = cluster_survmetrics_mace,
    TimetoDM_summary = timetodmsum,
    Cluster_TimetoDM_summary = cluster_timetodmsum,
    Cluster_TimetoDM_Adjusted = cluster_adjtimetodmsum,
    Cluster_TimetoDM_Threshold = cluster_adjtimetodmsum_thresh,
    Global_SurvMetrics_TimetoDM = global_survmetrics_timetodm,
    Cluster_SurvMetrics_TimetoDM = cluster_survmetrics_timetodm
)
```

In [67]:
```
save(result_file, file = "result_file.RData")
```

Uploading results

The results shown do not contain any individual level data. We require that you upload this 'result_file.RData' file to the Teams folder in SOPHIA, which would be located here:

> CrossWP > Analyst working groups > WG1 > UMAP_project > *cohort_name* > data