Summary of TMB runs

Lena Morrill

24/05/2021

Contents

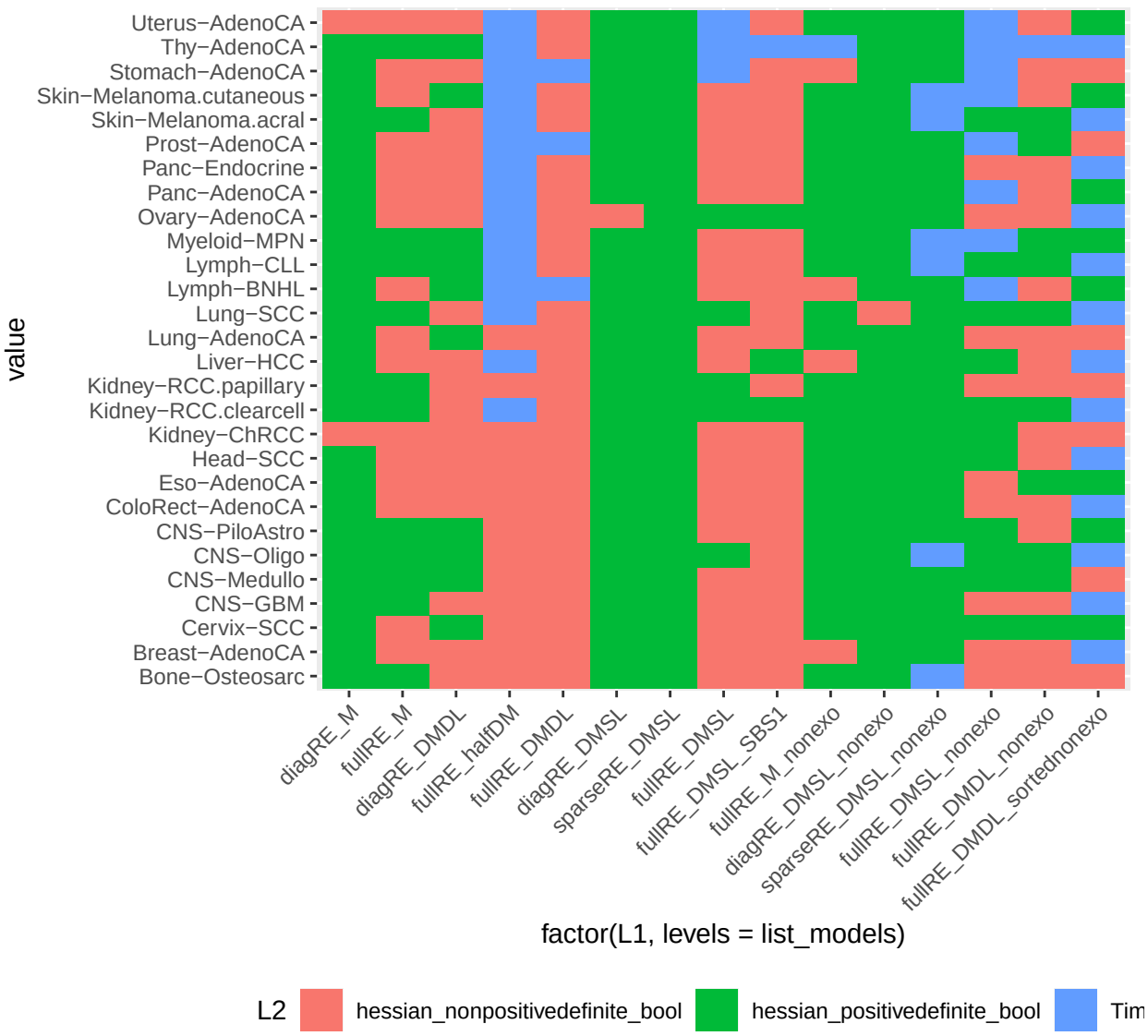
Information about models

Default order of categories for each model

Name model	Extension	Sorted	File in which they were created
fullREDMsinglelambda	fullRE_DMSL_	Not sorted	run_TMB_PCAWG.R
fullREDMsinglelambda2	fullRE_DMSL2_	Sorted	run_TMB_PCAWG.R
diagREDMsinglelambda	diagRE_DMSL_	Unknown	run_TMB_PCAWG.R
fullRE_M	fullRE_M_	Sorted in previous version of wrapper_run_TMB	run_TMB_PCAWG.R
diagRE_DM	diagRE_DM_	Sorted in previous version of wrapper_run_TMB	run_TMB_PCAWG.R
fullRE_DM	fullRE_DM_	Sorted in previous version of wrapper_run_TMB	run_TMB_PCAWG.R
sparseRE_DMSL2	sparseRE_nonexo_DMSL_	Sorted	find_subset_signatures.R
fullREDMsinglelambda	fullRE_nonexo_DMSL_	Not sorted	find_subset_signatures.R
fullRE_M	fullRE_nonexo_M_	Not sorted	find_subset_signatures.R
diagREDMsinglelambda	diagRE_nonexo_DMSL_	Not sorted	find_subset_signatures.R
fullRE_DM	fullLRE_nonexo_DM_	Not sorted	find_subset_signatures.R
diagREDMsinglelambda	diagRE_DMSL_	Not sorted	find_subset_signatures.R

General results of all models

Check the results of all of the models


value
Uterus−AdenoCA
Thy−AdenoCA
Stomach−AdenoCA
Skin−Melanoma.cutaneous
Skin−Melanoma.acral
Prost−AdenoCA
Panc−Endocrine
Panc−AdenoCA
Ovary−AdenoCA
Myeloid−MPN
Lymph−CLL
Lymph−BNHL
Lung−SCC
Lung−AdenoCA
Liver−HCC
Kidney−RCC.papillary
Kidney−RCC.clearcell
Kidney−ChRCC
Head−SCC
Eso−AdenoCA
ColoRect−AdenoCA
CNS−PiloAstro
CNS−Oligo
CNS−Medullo
CNS−GBM
Cervix−SCC
Breast−AdenoCA
Bone−Osteosarc
diagRE_M
fullRE_M
diagRE_DMDL
fullRE_halfDM
fullRE_DMDL
diagRE_DMSL
sparseRE_DMSL
fullRE_DMSL
fullRE_DMSL_SBS1
fullRE_M_nonexo
diagRE_DMSL_nonexo
sparseRE_DMSL_nonexo
fullRE_DMSL_nonexo
fullRE_DMDL_nonexo
fullRE_DMDL_sortednonexo
factor(L1, levels = list_models)
L2
hessian_nonpositivedefinite_bool
hessian_positivedefinite_bool
Tim

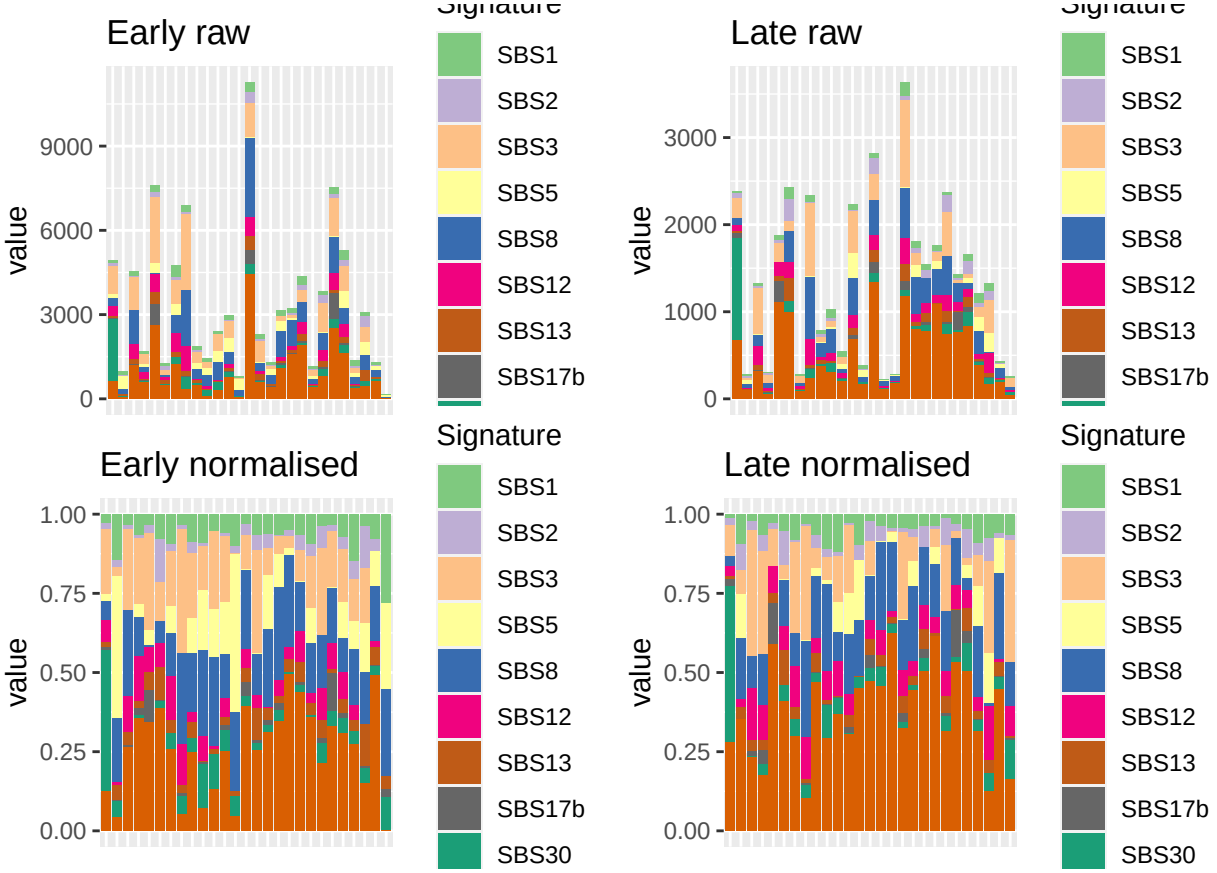
Analysis per cancer type

Bone osteosarcoma

Barplot and general statistics

```
## [1] 27
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
```



The number of samples and signatures is:

```
## [1] 54 10
```

The signatures are:

```
##  [1] "SBS1"   "SBS2"   "SBS3"   "SBS5"   "SBS8"   "SBS12"  "SBS13"  "SBS17b"
##  [9] "SBS30"  "SBS40"
```

Convergence table

We only have converged results for the multinomial with full RE, and the DM with a single lambda (diag and full RE). It is the same for nonexogenous signatures.

```
##              value                          L2                          L1
```

```
## 1  Bone-Osteosarc    hessian_positivedefinite_bool                diagRE_M
## 2  Bone-Osteosarc    hessian_positivedefinite_bool                fullRE_M
## 3  Bone-Osteosarc hessian_nonpositivedefinite_bool             diagRE_DMDL
## 4  Bone-Osteosarc hessian_nonpositivedefinite_bool            fullRE_halfDM
## 5  Bone-Osteosarc hessian_nonpositivedefinite_bool              fullRE_DMDL
## 6  Bone-Osteosarc    hessian_positivedefinite_bool              diagRE_DMSL
## 7  Bone-Osteosarc    hessian_positivedefinite_bool            sparseRE_DMSL
## 8  Bone-Osteosarc hessian_nonpositivedefinite_bool              fullRE_DMSL
## 9  Bone-Osteosarc hessian_nonpositivedefinite_bool         fullRE_DMSL_SBS1
## 10 Bone-Osteosarc    hessian_positivedefinite_bool          fullRE_M_nonexo
## 11 Bone-Osteosarc    hessian_positivedefinite_bool       diagRE_DMSL_nonexo
## 12 Bone-Osteosarc                         Timeout     sparseRE_DMSL_nonexo
## 13 Bone-Osteosarc hessian_nonpositivedefinite_bool       fullRE_DMSL_nonexo
## 14 Bone-Osteosarc hessian_nonpositivedefinite_bool       fullRE_DMDL_nonexo
## 15 Bone-Osteosarc hessian_nonpositivedefinite_bool fullRE_DMDL_sortednonexo
```

Re-running of fitting

```
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
```

If we use the values of the fullRE M as initial values for the fullRE DM, we also don't get convergence:

```
## [1] FALSE
```

Potentially problematic signatures

We notice that we have several signatures with low exposures, and many zero exposures

```
colSums(obj_Bone_Osteosarc$Y == 0)/nrow(obj_Bone_Osteosarc$Y)
```

```
##       SBS1       SBS2       SBS3       SBS5       SBS8      SBS12      SBS13
## 0.00000000 0.03703704 0.14814815 0.37037037 0.01851852 0.09259259 0.00000000
##     SBS17b      SBS30      SBS40
## 0.37037037 0.12962963 0.01851852
```

```
colSums(obj_Bone_Osteosarc$Y)/sum(obj_Bone_Osteosarc$Y)
```

```
##       SBS1       SBS2       SBS3       SBS5       SBS8      SBS12      SBS13
## 0.05099661 0.03376971 0.17876022 0.05053018 0.17164713 0.07538325 0.04159022
##     SBS17b      SBS30      SBS40
## 0.02866227 0.06128922 0.30737119
```

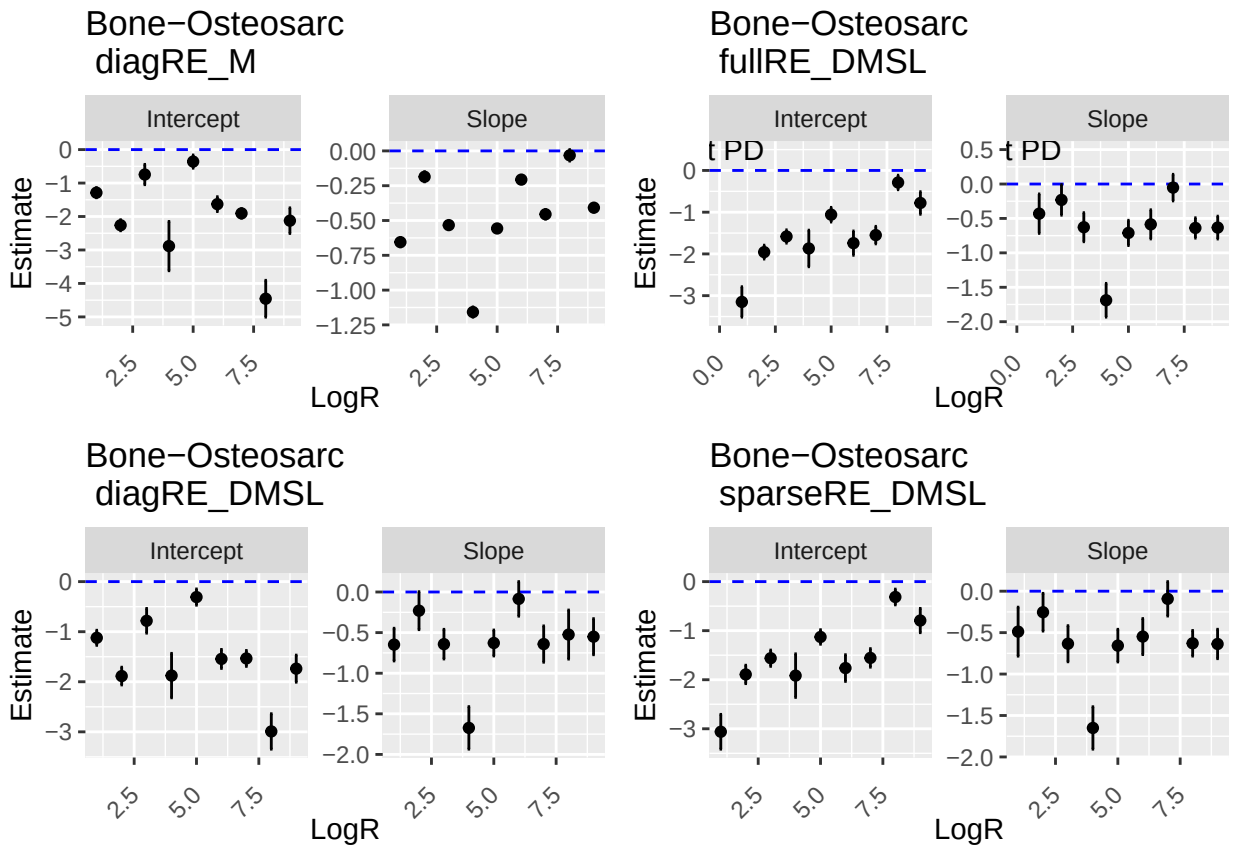
E.g.

- SBS17b is 0 in 37% of cases and has an overall exposure of 2.9%
- SBS30 is 0 in 13% of cases and overal has an exposure of only 6.1%
- SBS5 is 0 in 37% of cases and has an overall exposure of 5.1%

Betas

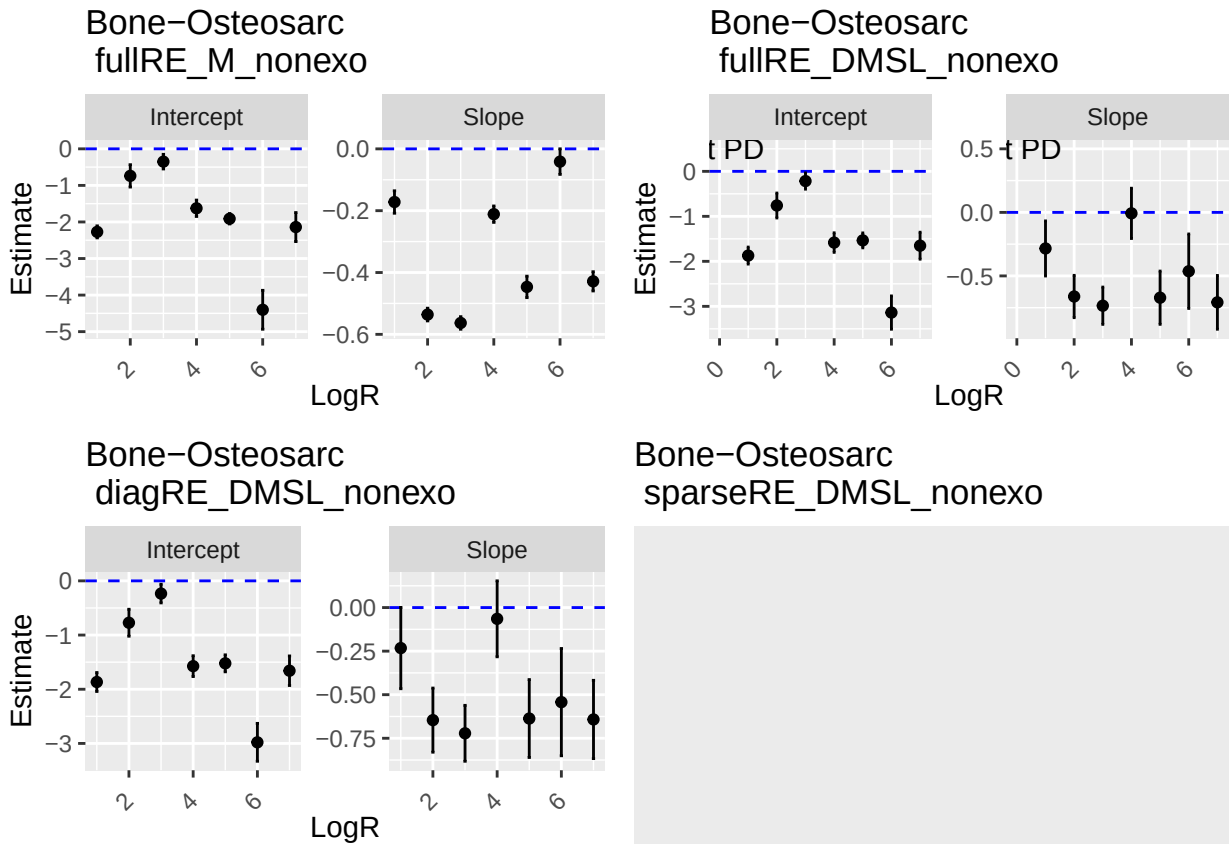
```
ct <- "Bone-Osteosarc"

grid.arrange(plot_betas(diagRE_M[[ct]])+ggtitle(paste0(ct, '\n diagRE_M')),
plot_betas(fullRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL')),
```

```
plot_betas(diagRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL')),
plot_betas(sparseRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL')), nrow=2)
```




```
grid.arrange(
  plot_betas(fullRE_M_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_M_nonexo')),
  plot_betas(fullRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL_nonexo')),
plot_betas(diagRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL_nonexo')),
plot_betas(sparseRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL_nonexo')), nrow=2)
```

```
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
```




```
## Warning in select_slope_2(which(names(i$par.fixed) == "beta"), verbatim
## = verbatim): As per 27 August it seems clear that this version, and not
## <select_slope>, is correct
```

```
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used
```

```
## Warning in wald_generalised(v = i$par.fixed[idx_beta], sigma =
## i$cov.fixed[idx_beta, : 20201218: sigma**(1/2) has now been replaced by (as we
## had before sometime in November) sigma
```

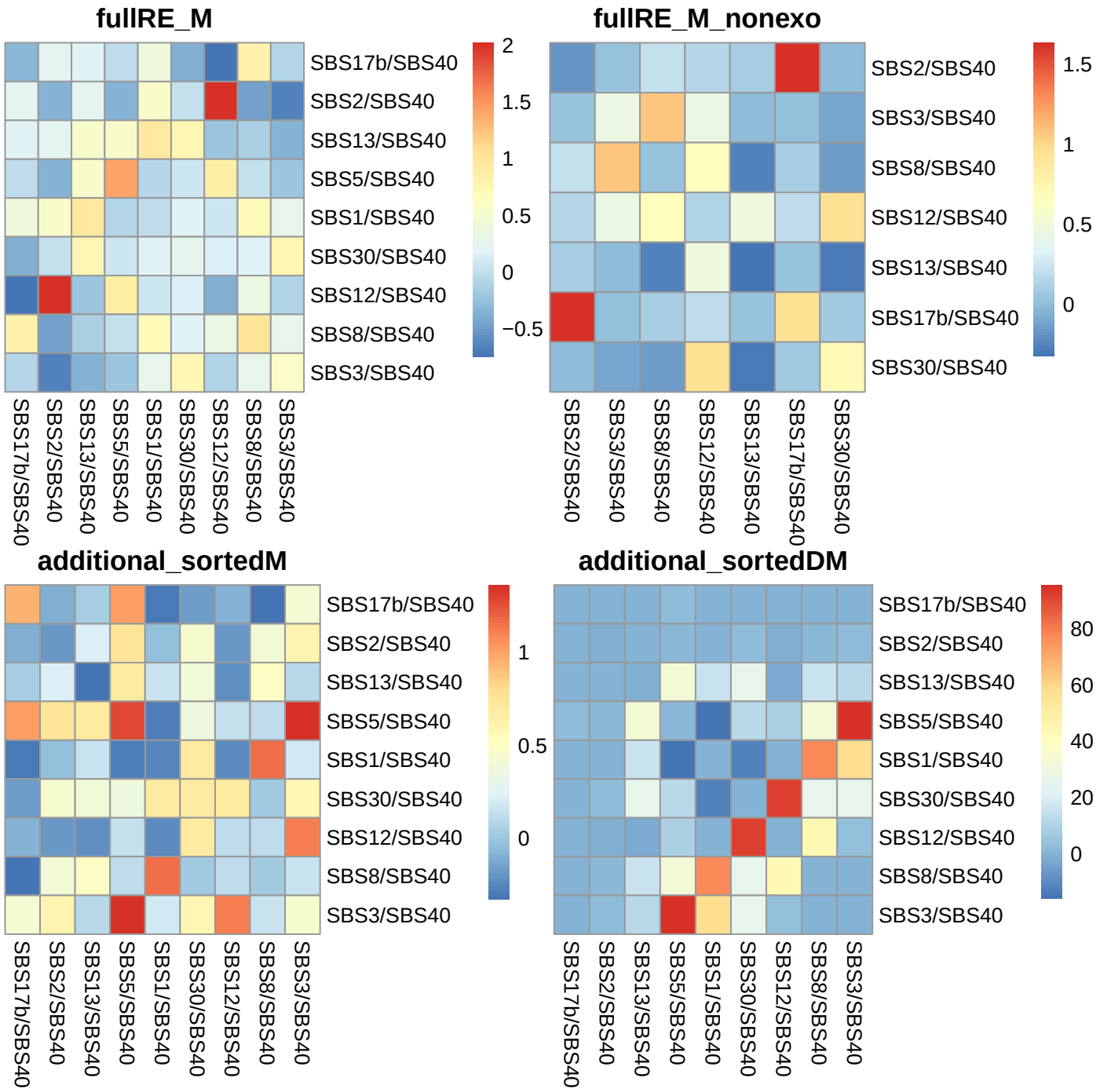
We use the results from the diagonal single lambda DM to test for differential abundance, giving a p-value of 3.8923434×10^{-5}.

Covariance matrices

```
ct <- "Bone-Osteosarc"
additional_sortedM <- list()
additional_sortedDM <- list()
additional_sortedM[[ct]] <- sortedM
additional_sortedDM[[ct]] <- sortedDM
```

Note that sortedDM did not convergence.

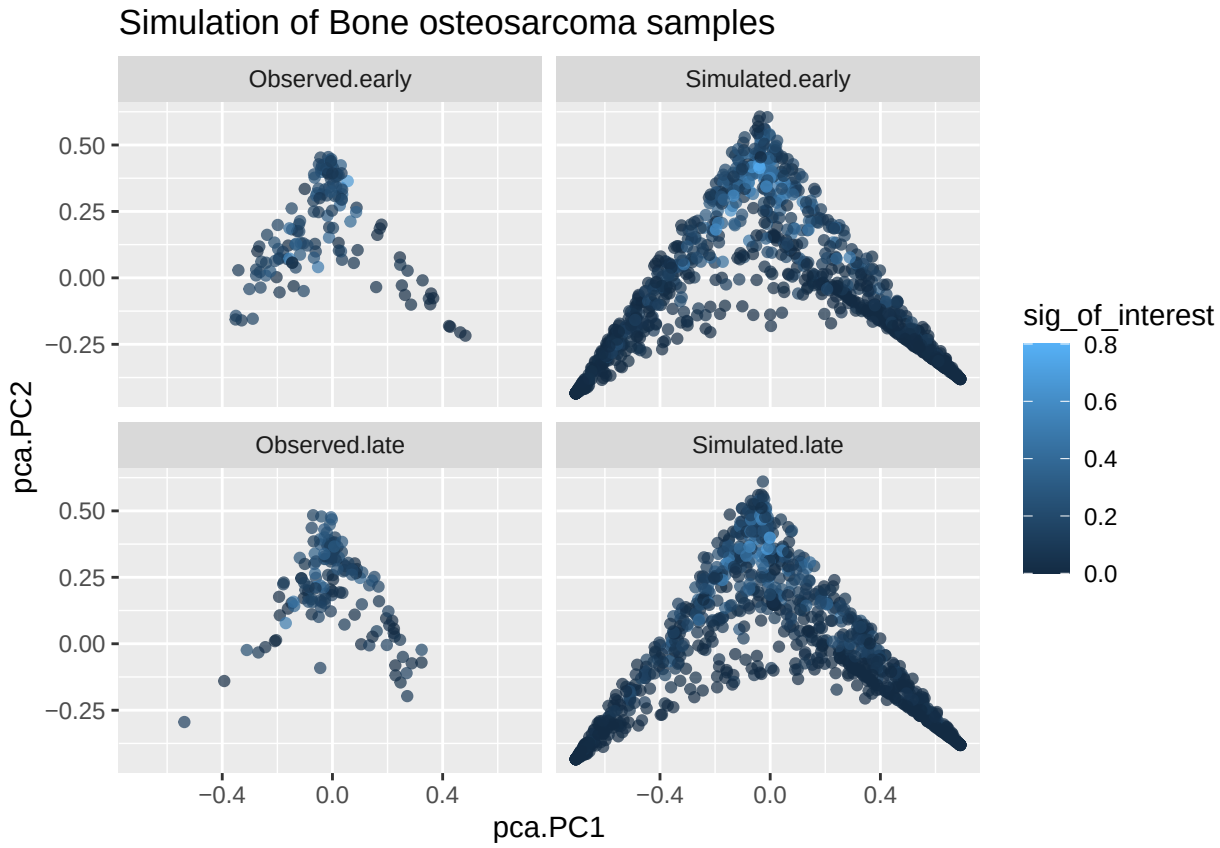
Nevertheless, both versions of fullRE M – both of which converged and use the same baseline – give very different covariances matrices.



Simulation under inferred data

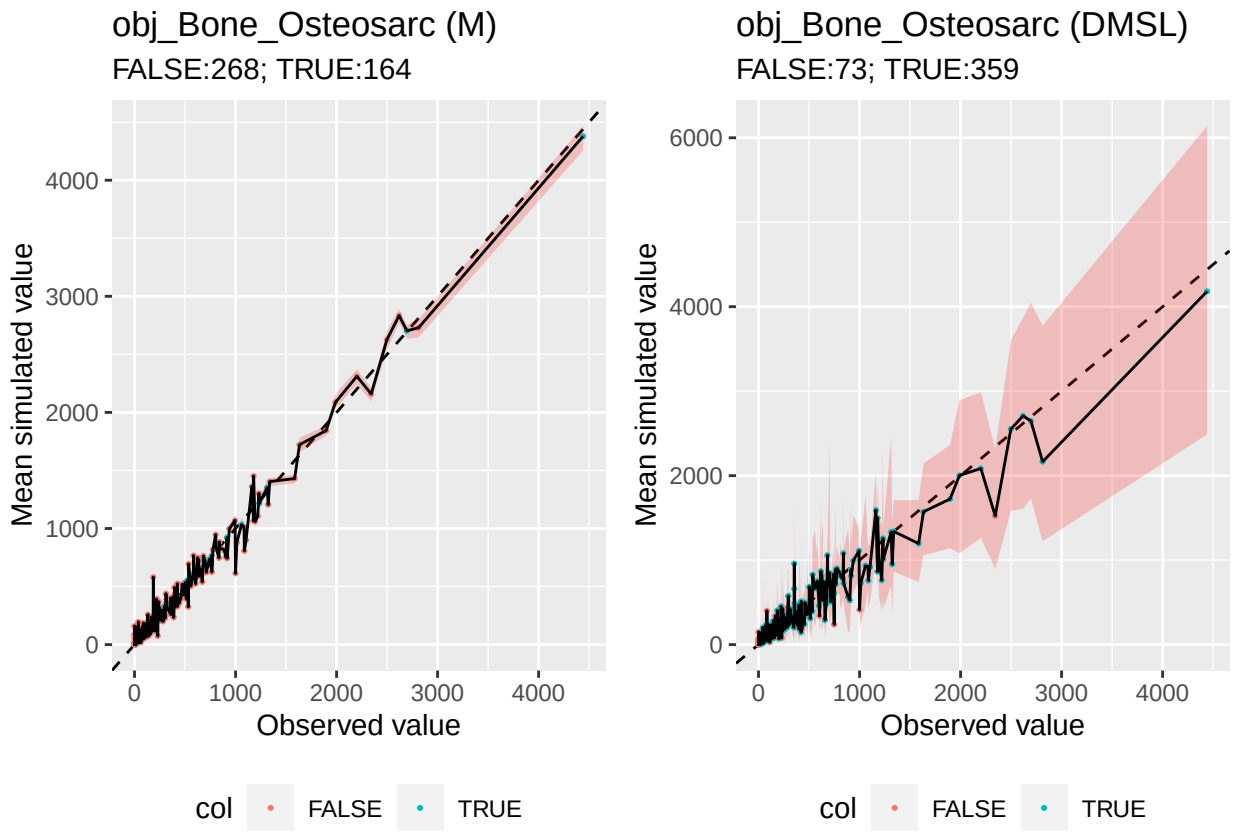
```
## [1] 136
```

```
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```



Ranked plot for coverage

```
ct <- "Bone-Osteosarc"
integer_overdispersion_param_DMSL <- 1
obj_Bone_Osteosarc_nonexo <- give_subset_sigs_TMBobj(obj_Bone_Osteosarc, sigs_to_remove = nonexogenous$V1
grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = ful
                  data_object = obj_Bone_Osteosarc_nonexo,
                  print_plot = F, nreps = 20, model = "M")), function(i){
                    lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
                                                           rank_number=1:length(j)) )})[[1]],
                  data_object = obj_Bone_Osteosarc_nonexo,
                  loglog = F, title = 'obj_Bone_Osteosarc (M)'),
 give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nor
                  data_object = obj_Bone_Osteosarc_nonexo,
                  print_plot = F, nreps = 20, model = "DMSL", integer_overdispersion_param = integer_overd
                    lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
                                                           rank_number=1:length(j)) )})[[1]],
                  data_object = obj_Bone_Osteosarc_nonexo,
                  loglog = F, title = 'obj_Bone_Osteosarc (DMSL)'), ncol=2)
```



73/359=20% of values are not included in the confidence interval of the DMSL.

Signatures from mutSigExtractor

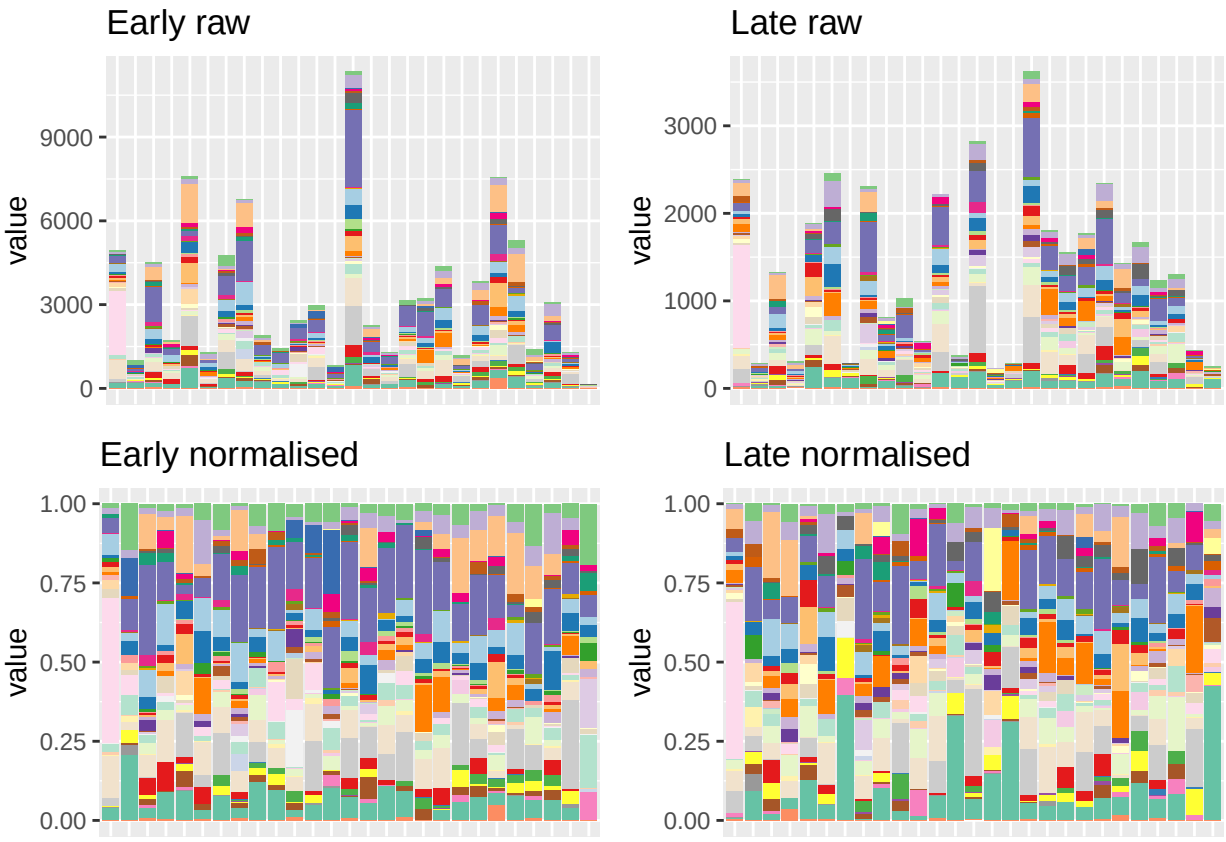
The signatures from mutSigExtractor are a bit more chaotic:

```
obj_Bone_Osteosarc_mutSigExtractor <- load_PCAWG(ct = ct, typedata = "signaturesmutSigExtractor",
                                    path_to_data = "../../data/")
```

```
## [1] 27
```

```
give_barplot_from_obj(obj = obj_Bone_Osteosarc_mutSigExtractor, legend_on = FALSE)
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
```

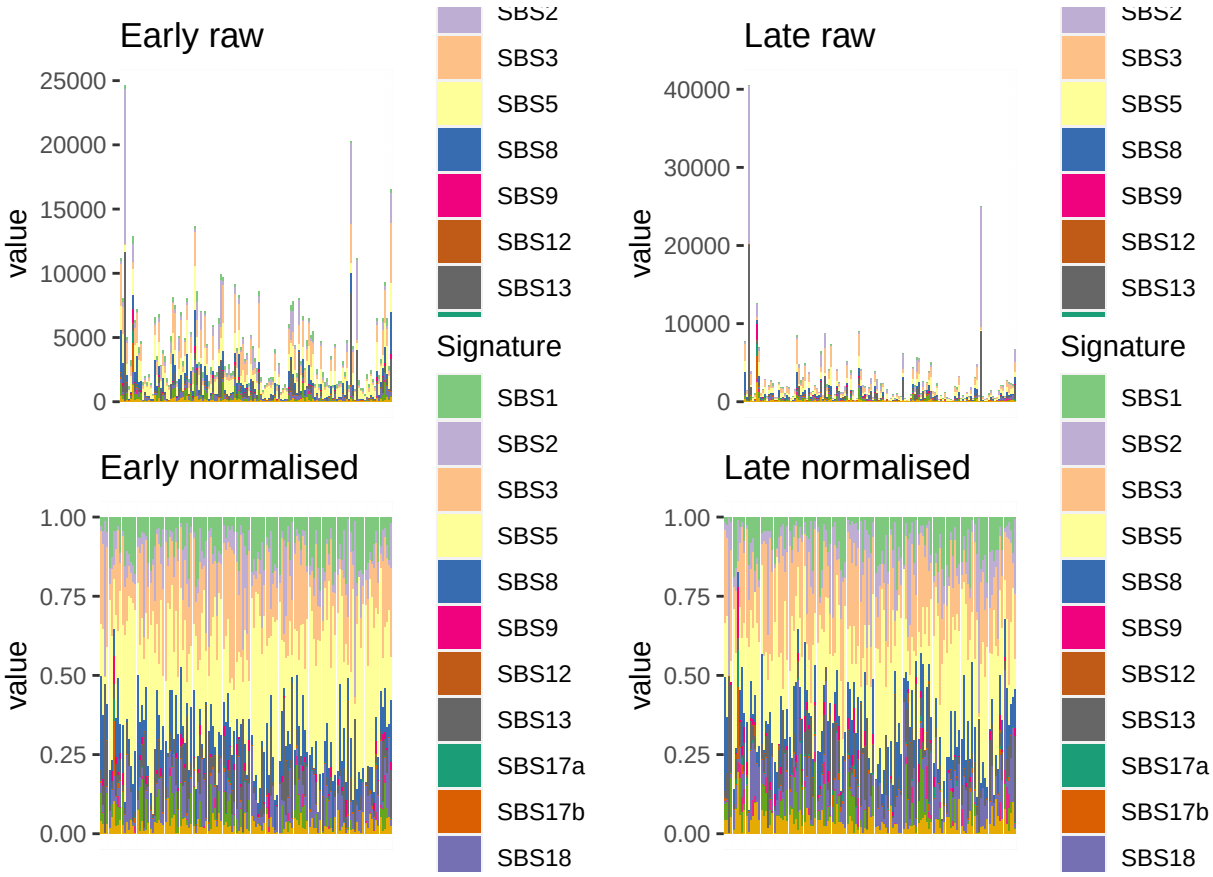


Breast-AdenoCA

Barplot and general statistics

```
## [1] 136
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
```



There are many signatures, and also many samples.

The number of samples and signatures is:

```
## [1] 272  14
```

The signatures are:

```
##  [1] "SBS1"   "SBS2"   "SBS3"   "SBS5"   "SBS8"   "SBS9"   "SBS12"  "SBS13"
##  [9] "SBS17a" "SBS17b" "SBS18"  "SBS37"  "SBS39"  "SBS41"
```

Convergence table

We only have converged results for the diagRE_DMSL, with diagonal or sparse covariance structure, and diagonal M. This is probably due to the very high number of signatures, which make it impossible to infer the whole covariance structure.

```
##              value                                L2                 L1
## 1  Breast-AdenoCA    hessian_positivedefinite_bool          diagRE_M
## 2  Breast-AdenoCA hessian_nonpositivedefinite_bool          fullRE_M
## 3  Breast-AdenoCA hessian_nonpositivedefinite_bool       diagRE_DMDL
## 4  Breast-AdenoCA hessian_nonpositivedefinite_bool     fullRE_halfDM
## 5  Breast-AdenoCA hessian_nonpositivedefinite_bool       fullRE_DMDL
## 6  Breast-AdenoCA    hessian_positivedefinite_bool       diagRE_DMSL
## 7  Breast-AdenoCA    hessian_positivedefinite_bool     sparseRE_DMSL
## 8  Breast-AdenoCA hessian_nonpositivedefinite_bool       fullRE_DMSL
## 9  Breast-AdenoCA hessian_nonpositivedefinite_bool fullRE_DMSL_SBS1
```

```
## 10 Breast-AdenoCA hessian_nonpositivedefinite_bool          fullRE_M_nonexo
## 11 Breast-AdenoCA    hessian_positivedefinite_bool       diagRE_DMSL_nonexo
## 12 Breast-AdenoCA    hessian_positivedefinite_bool     sparseRE_DMSL_nonexo
## 13 Breast-AdenoCA hessian_nonpositivedefinite_bool       fullRE_DMSL_nonexo
## 14 Breast-AdenoCA hessian_nonpositivedefinite_bool       fullRE_DMDL_nonexo
## 15 Breast-AdenoCA                          Timeout fullRE_DMDL_sortednonexo
```

Re-running of fitting

If we use the values of the diagRE M as initial values for the diagRE DM, we that it has converged. This is probably due to a combination of things: we are using the optimiser nlminb (better in general than the alternative, optim) and we are starting with these - better - values, and we are sorting the columns so that the category with highest total value is the baseline.

```
## [1] TRUE
```

```
ct <- "Breast-AdenoCA"
additional_sorteddiagM <- list()
additional_sorteddiagDM <- list()
additional_sorteddiagM[[ct]] <- sortedM_Breast_Adeno
additional_sorteddiagDM[[ct]] <- sortedDM_Breast_Adeno
```

Potentially problematic signatures

We notice that we have several signatures with low exposures, and many zero exposures

```
colSums(obj_Breast_AdenoCA$Y == 0)/nrow(obj_Breast_AdenoCA$Y)
```

```
##         SBS1         SBS2         SBS3         SBS5         SBS8         SBS9
## 0.000000000 0.000000000 0.025735294 0.007352941 0.088235294 0.562500000
##        SBS12        SBS13       SBS17a       SBS17b        SBS18        SBS37
## 0.955882353 0.073529412 0.709558824 0.500000000 0.036764706 0.772058824
##        SBS39        SBS41
## 0.599264706 0.084558824
```

```
colSums(obj_Breast_AdenoCA$Y)/sum(obj_Breast_AdenoCA$Y)
```

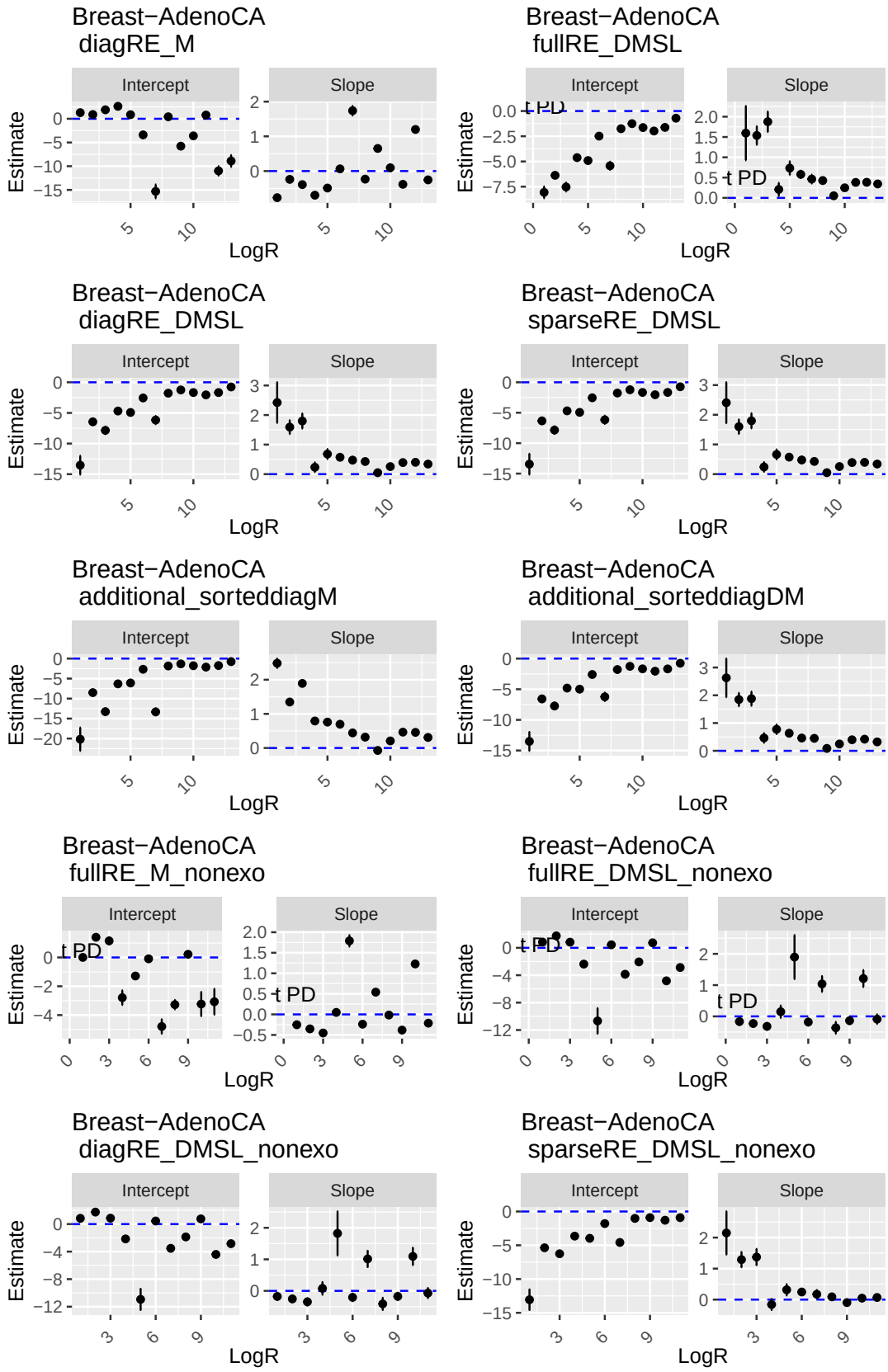
```
##          SBS1          SBS2          SBS3          SBS5          SBS8          SBS9
## 0.0553410311 0.1376261991 0.1993274971 0.2185906789 0.0969490005 0.0132833987
##         SBS12         SBS13        SBS17a        SBS17b         SBS18         SBS37
## 0.0003532317 0.1360853961 0.0036266519 0.0081714966 0.0531199688 0.0057240307
##         SBS39         SBS41
## 0.0402034279 0.0315979909
```

E.g.

- SBS9 is 0 in 56.2% of cases and has an overall exposure of 1.3%
- SBS12 is 0 in 95.6% of cases and has an overall exposure of 0%
- SBS17a is 0 in 71% of cases and has an overall exposure of 0.4%
- SBS17b is 0 in 50% of cases and has an overall exposure of 0.8%
- SBS37 is 0 in 77.2% of cases and has an overall exposure of 0.6%
- SBS39 is 0 in 59.9% of cases and has an overall exposure of 4%

Betas

```
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
## Warning in sqrt(as.numeric(object$diag.cov.random)): NaNs produced
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
## Warning in sqrt(as.numeric(object$diag.cov.random)): NaNs produced
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
```


Breast−AdenoCA
diagRE_M
Intercept
Slope
Estimate
LogR
Breast−AdenoCA
fullRE_DMSL
t PD
Breast−AdenoCA
diagRE_DMSL
Breast−AdenoCA
sparseRE_DMSL
Breast−AdenoCA
additional_sorteddiagM
Breast−AdenoCA
additional_sorteddiagDM
Breast−AdenoCA
fullRE_M_nonexo
Breast−AdenoCA
fullRE_DMSL_nonexo
Breast−AdenoCA
diagRE_DMSL_nonexo
Breast−AdenoCA
sparseRE_DMSL_nonexo

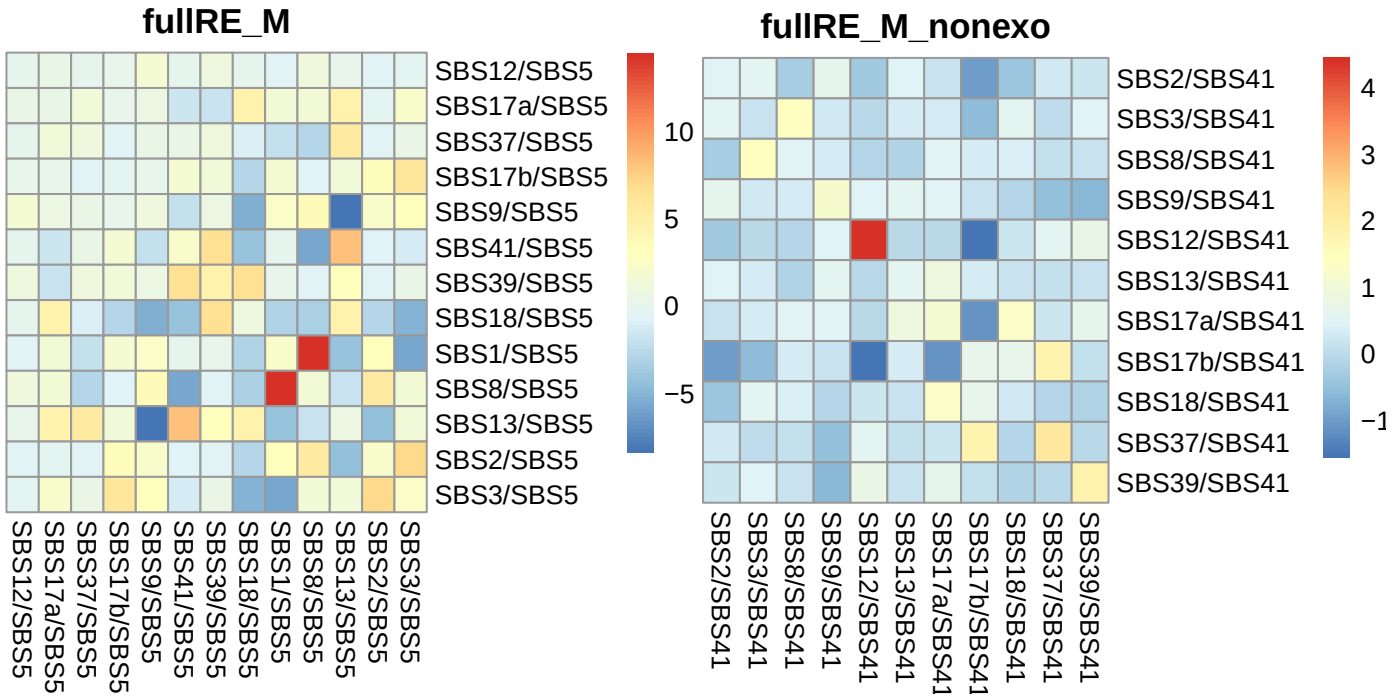
```
## Warning in select_slope_2(which(names(i$par.fixed) == "beta"), verbatim
## = verbatim): As per 27 August it seems clear that this version, and not
## <select_slope>, is correct
```

```
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used
```

```
## Warning in wald_generalised(v = i$par.fixed[idx_beta], sigma =
## i$cov.fixed[idx_beta, : 20201218: sigma**(1/2) has now been replaced by (as we
## had before sometime in November) sigma
```

We use the results from the diagonal single lambda DM to test for differential abundance, giving a p-value of 7.748574×10^{-12}.

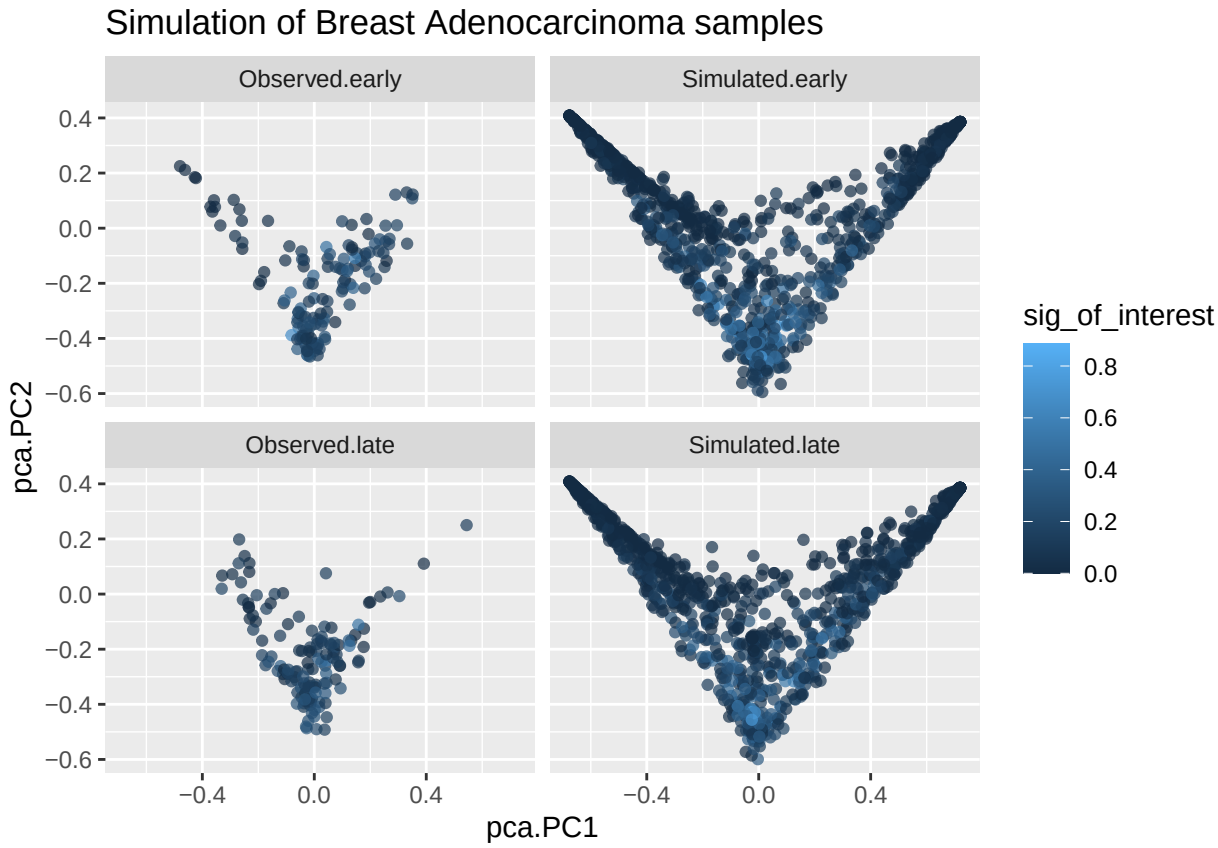
Covariance matrices



Simulation under inferred data

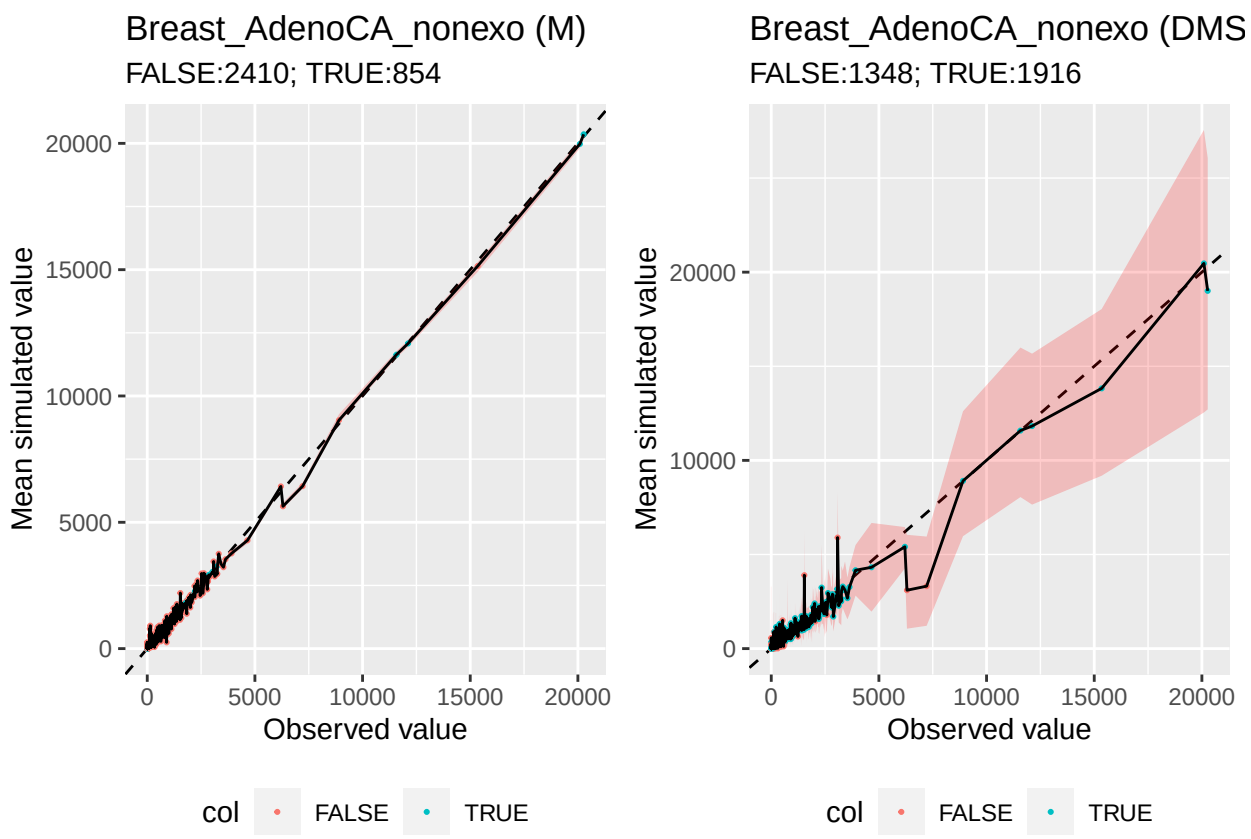
```
## [1] 136
```

```
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```



Ranked plot for coverage

```
ct <- "Breast-AdenoCA"
integer_overdispersion_param_DMSL <- 1
obj_Breast_AdenoCA_nonexo <- give_subset_sigs_TMBobj(obj_Breast_AdenoCA, sigs_to_remove = nonexogenous$V1
grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = ful
                  data_object = obj_Breast_AdenoCA_nonexo,
                  print_plot = F, nreps = 20, model = "M")), function(i){
                    lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
                                                           rank_number=1:length(j)) )})[[1]],
                  data_object = obj_Breast_AdenoCA_nonexo,
                  loglog = F, title = 'Breast_AdenoCA_nonexo (M)'),
 give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nor
                  data_object = obj_Breast_AdenoCA_nonexo,
                  print_plot = F, nreps = 20, model = "DMSL", integer_overdispersion_param = integer_overd
                    lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
                                                           rank_number=1:length(j)) )})[[1]],
                  data_object = obj_Breast_AdenoCA_nonexo,
                  loglog = F, title = 'Breast_AdenoCA_nonexo (DMSL)'), ncol=2)
```



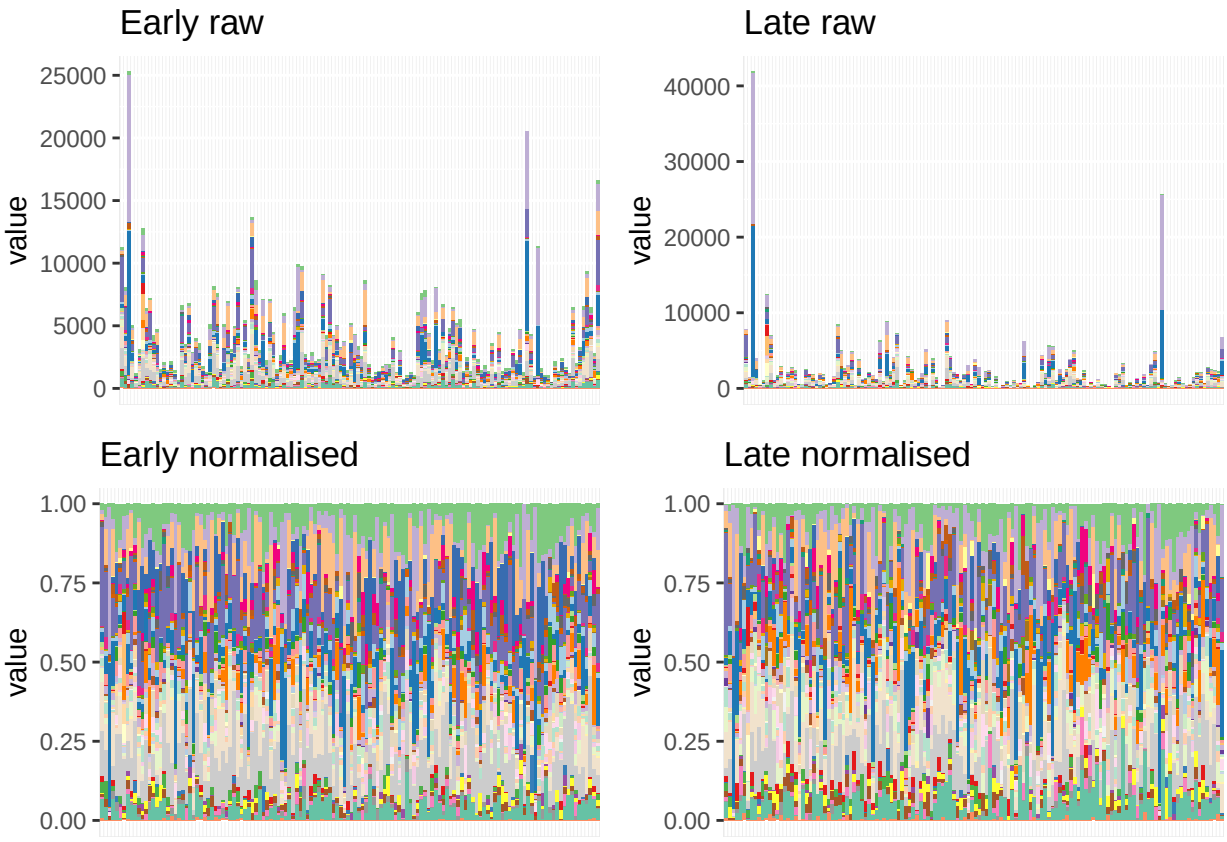
Signatures from mutSigExtractor

```
obj_Breast_AdenoCA_mutSigExtractor <- load_PCAWG(ct = "Breast-AdenoCA", typedata = "signaturesmutSigExtra
                                      path_to_data = "../../data/")
```

```
## [1] 136
```

```
give_barplot_from_obj(obj = obj_Breast_AdenoCA_mutSigExtractor, legend_on = FALSE)
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
```


Early raw
Late raw
Early normalised
Late normalised
value
25000
20000
15000
10000
5000
0
40000
30000
1.00
0.75
0.50
0.25
0.00

Cervix-SCC

CNS-GBM

CNS-Medullo

CNS-Oligo

CNS-PiloAstro

ColoRect-AdenoCA

Eso-AdenoCA

Head-SCC

Kidney-ChRCC

Kidney-RCC.clearcell

Kidney-RCC.papillary

Liver-HCC

Lung-AdenoCA

Lung-SCC

Lymph-BNHL

Lymph-CLL

Myeloid-MPN

Ovary-AdenoCA

Panc-AdenoCA

Panc-Endocrine

Prost-AdenoCA

Skin-Melanoma.acral

Skin-Melanoma.cutaneous

Stomach-AdenoCA

Thy-AdenoCA

Uterus-AdenoCA

All p-values for non-exogenous signatures

% latex table generated in R 4.0.3 by xtable 1.8-4 package % Mon May 24 23:58:16 2021

	ct	pvalue	model
1	Bone-Osteosarc	0.00	diagRE_DMSL_nonexo
2	Breast-AdenoCA	0.00	diagRE_DMSL_nonexo