**Suplementary file 1**

***DS patient and control inclusion using QF-PCR analysis:*** QF-PCR was performed with the QSTR-plus, v2-kit of Elucigene (Manchester, United Kingdom) for chromosomes 13,18, 21, X and Y. The manufacturer’s protocol was used for analysis of all patients. Eleven DS patients showed a trisomy 21 with QF-PCR (Supplementary file table 1) and 18 control samples showed normal ploidy for chromosome 21. One DS subject and one control subject showed inconclusive QF-PCR results and were submitted for additional CGH analysis. One Control subject showed failed QF-PCR results and was omitted in further analysis. Note that the additional markers annotated to chromosome 13, 18 and both allosomes showed no aberrant profile (data not shown).

**Suplementary file table 1:** Trisomy 21 confirmation whole cohort QF PCR

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **D21S11** | **D21S1246** | **D21S1280** | **D21S1409** | **D21S1411** | **D21S1442** | **D21S1446** | **D21S2055** |
| **DS\_1 \*** | 1:1:1 | 1 | 1:2 | 2:1 | 2:1 | 1:2 | 1:2 | **1:1** |
| **DS\_2** | 1:1:1 | 1:2 | 1:1:1 | 1 | 1:2 | 1:1:1 | 2:1 | 1:1:1 |
| **DS\_3**  | 1:1:1 | 1:1:1 | 2:1 | 1:2 | 1:1:1 | 1:2 | 2:1 | 1:1:1 |
| **DS\_4** | 1:2 | 1:1:1 | 1:1:1 | 1:2 | 2:1 | 1:1:1 | 2:1 | 1:1:1 |
| **DS\_5** | 1:1:1 | 1:1:1 | 1:1:1 | 1:2 | 2:1 | 1:1:1 | 1:1:1 | 1:1:1 |
| **DS\_6** | 2:1 | 2:1 | 1:1:1 | 1 | 1:1:1 | 2:1 | 1:2 | 2:1 |
| **DS\_7** | 1 | 2:1 | 1:2 | 2:1 | 1:1:1 | 2:1 | 1:1:1 | 1:1:1 |
| **DS\_8** | 1:1:1 | 1:1:1 | 1:1:1 | 2:1 | 1:1:1 | 1:2 | 2:1 | 2:1 |
| **DS\_9** | 1:2 | 2:1 | 2:1 | 2:1 | 1:2 | 2:1 | 1:2 | 2:1 |
| **DS\_10** | 1:1:1 | 1 | 1:1:1 | 1:1:1 | 1 | 1:2 | 1 | 2:1 |
| **DS\_11** | 2:1 | 1 | 1:2 | 1:2 | 2:1 | 1:2 | 1 | 1:1:1 |
| **DS\_12** | 1 | 2:1 | 2:1 | 1:1:1 | 1:2 | 1:2 | 1:2 | 1:1:1 |
| **C\_1** | 1:1 | 1:1 | 1 | 1 | 1 | 1:1 | 1 | 1:1 |
| **C\_2** | 1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 |
| **C\_3** | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1 | 1 | 1:1 |
| **C\_4** | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 |
| **C\_5 \*** | 1 | 1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | **8:1** |
| **C\_6** | 1:1 | 1 | 1:1 | 1:1 | 1 | 1:1 | 1:1 | 1:1 |
| **C\_7** | 1:1 | 1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1 |
| **C\_8** | 1:1 | 1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 |
| **C\_9** | 1:1 | 1:1 | 1:1 | 1:1 | 1 | 1:1 | 1:1 | 1 |
| **C\_10** | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1 | 1 |
| **C\_11** | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 |
| **C\_12** | 1:1 | 1:1 | 1:1 | 1:1 | 1 | 1:1 | 1 | 1:1 |
| **C\_13** | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1 | 1 | 1:1 |
| **C\_14** | 1:1 | 1:1 | 1:1 | 1 | 1:1 | 1:1 | 1 | 1:1 |
| **C\_15** | 1:1 | 1 | 1:1 | 1:1 | 1:1 | 1 | 1 | 1:1 |
| **C\_16**  | 1 | 1 | 1 | 1:1 | 1:1 | 1:1 | 1 | 1 |
| **C\_17 \*\*** | **NA** | **NA** | **NA** | **NA** | **NA** | **NA** | **NA** | **NA** |
| **C\_18** | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1 | 1:1 |
| **C\_19** | 1:1 | 1 | 1:1 | 1:1 | 1:1 | 1:1 | 1 | 1:1 |
| **C\_20** | 1:1 | 1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 | 1:1 |

DS: Down syndrome patient; C: control; Ratio between peaks reflecting number of alleles chromosome 21; NA: Failed.

\* Sample submitted for CGH assay for second trisomy 21 confirmation.

 \*\* Sample excluded from further analysis

***DS patient and control inclusion using Comparative Genomic Hybridisation (CGH):***In order to confirm or exclude aneuploidy regarding chromosome 21 for the DS sample 1 and control sample 5 respectively, we performed comparative genomic hybdrization (CGH) arrays using 4x180K slides, AMADID 023363 (Agilent Technologies, Santa Clara, CA, USA) according to the manufacturer’s protocol. Results were processed and visualized in genome view software (v. download N42), wherein the x-axis represents the chromosome number and the y-axis represents duplication/deletion level. Supplementary file figure 1 A illustrates the CGH result for DS sample 1 where we confirmed the presence of trisomy 21. Supplementary file figure 1 B confirmed normal diploidic status for control sample 5.

**Suplementary file figure 1**: CGH confirmation inconclusive QF-PCR; (A) DS-1, (B) Control-5

A CGH DS-1



B CGH Control-5



***Differential Methylation analysis: Description top five DMPs***

Differential DNA-methylation analyses of neuronal fractions between DS and controls yielded in total 9908 significant positions (table 2-I , Suplementary table 1). The DMP top hit was annotated 3’downstream of the *MSRB3* gene, which encodes for the Methionine Sulfoxide Reductase B3 protein and was previously linked to deafness[1]. The second top hit was annotated 5’upstream of the *SSBP3* gene, encoding for single stranded DNA binding protein 3. This protein may be involved in panhypopituitarism and regulation of cell differentiation[2]. Thirdly, we detected a DMP, located at chromosome 11, annotated to the promoter and region of the *KDM2A* gene, encoding for the lysine demethylase 2A protein. This protein might be an important player within the scope of the epigenetic machinery and previously it has been suggested that KDM2A is involved in a broad scope of diseases and developmental abberations[3]. Fourth, we observed a DS differential methylated locus annotated to the promoter region of the *ZNF2* gene, encoding for the zinc and ring finger 2 protein. It has been suggested that this protein plays a role in the establishment and maintenance of neuronal transmission and plasticity via its ubiquitin ligase activity[4]. Finally, our top five included a DMP located 5’upstream of the *DYNC1L12* gene, encoding the zinc finger CCCH-type containing 14 protein, which previously has been associated with Bardet-Biedl Syndrome 7[5].

***Differential Methylation analysis: Description top five DMRs***

In total, we observed 1291 DMRs at a Stouffer Coefficient (SC) < 0.05 involving two or more individual sites (table 2-II , Suplementary table 1). Two DMRs that both comprised a region of 17 adjacent CpGs, were annotated to the *CHKB-CPT1B* gene cluster and *CYP26C1* respectively. The *CHKB-CPT1B* gene cluster encoded proteins, choline kinase alpha and carnitine palmitoyltransferase 1B, previously have been associated with narcolepsy[6] and the *CYP26C1* gene, encoding cytochrome P450 family 26 subfamily C member 1 has not been linked to neuronal development or pathogenesis previously. Our third and fifth DMR were annotated to the gene body and promoter region respectively of the *NR2F1* gene, encoding for the nuclear receptor subfamily 2 group F member 1 protein that previously was linked to intellectual disability within the context of Bosch-Boonstra-Schaaf Optic Atrophy Syndrome[7]. Our fourth DMR, annotated to the promoter region of the *ADNP2* gene, encoding the ADNP homeobox 2 protein, was previously linked to neurodevelopment impairment[8]. We replicated 20 out 64 DMRs, based on the supplementary S9, of Mendioroz *et al* (2015), annotated to the following genes: *TNFRSF25, C1orf35, CELSR3, STK19, TNXB, HLA-DQB2, TMEM151B, GLI4, FAM83H, GPT / LRRC24, VPS37B, PIWIL1, DNAJC15, UNC45A, ADAMTS10, RYR1, FKRP, ZNF837, CECR2 and CPT1B.*  None of the loci was previously associated with DS or intellectual disability, with exception for: (1) *TNFRSF25*, encoding the TNF Receptor Superfamily Member 25 protein, which was previously associated with intellectual disability[9]; (2) *UNC45A*, encoding for the Unc-45 Myosin Chaperone A protein, that has been suggested to play a role in the early development of the human brain and aortic arches and structure of the human heart[10]; (3) *CECR2*, encoding for the CECR2 Histone Acetyl-Lysine Reader protein, which previously was associated with aberrant neural tube closure[11]

***Differential Expressed Gene analysis: Description top five DEGs***

Significant differential gene expression (DEG) was detected for 725 transcripts (table 2-III , Suplementary table 1). The top hit included DS overexpression of the *GAL3ST2* gene, encoding the Galactose-3-O-Sulfotransferase 2 protein, which was not previously associated with DS. The second most significant DEG was annotated to the *HMCN2* gene, encoding the Hemicentin 2. Although this transcript was not previously associated with DS, an important paralog of the latter is the gene *DSCAM* encoding the DS Cell Adhesion Molecule, which is highly conserved among species and plays essential roles in neurological development[12]. Whether HMCN2 for fills similar roles, needs further studies. Thirdly, we detected association of the *HAR1A* transcript with DS. *HARA1A* encodes the Highly Accelerated Region 1A protein that was previously linked to intellectual developmental disorders, but was previously not linked to DS directly[13]. The fourth and fifth top DEG, annotated to the *EXPH5* and *TRBV26OR9-2* both were not previously linked to DS or brain development processes.

***Overlap significant DMRs and DEGs: Description top five DMR/DEGs pairs***

We explored the overlap between 173 significant differentially methylated and expressed genes (figure 4a, table 2\_IV, and Suplementary table 1). The top five loci we detected here were annotated to the *EXPH5, ADAMTS18. LHX2, HSPA12A* and *ITPR2* genes. *EXPH5* annotated DMR involved included 6 adjacent CpG sites, and its DEG represented a 2,6 log2fold overexpression. Secondly, we detected DMR/DEG significance for *ADAMTS18* locus, eight CpGs and a 3,2 log2fold overexpression. Both these loci were not previously reported to be associated with DS or neuronal development. The third DMR/DEG pair was annotated to the *LHX2* locus, encoding the LIM Homeobox 2 protein. This locus may be associated with neural crest differentiation (Weizmann inst., *Pathcards*). Although, aberrant neural crest differentiation has been linked to intellectual disability, we found no previous reports linking it directly to DS[14]. The fourth locus we found was annotated to the *HSPA12A* gene, encoding Heat Shock 70 KDa Protein 12A. Although significant, the DEG effect size (overexpression) we observed was limited. Suppression of this locus was previously was linked to schizophrenia, which indeed is not frequently observed in DS[15, 16]. The fifth top hit DMR/DEG pair we detected was annotated to the *ITPR2* locus, encoding for the Inositol 1,4,5-Trisphosphate Receptor Type 2 protein, for which to the best of our knowledge no neuronal development or DS was previously reported.

***Limitations: Confounders, NeuN+ in development:***The present study was based on a DS and healthy control cohort obtained from the National Institute of Health (NIH) Neurobank, University of Maryland, Baltimore, MD, USA. The criteria for sample selection was primarily focused on matching the DS and controls for in particular sex, gestational age and post mortem interval of the samples. Ethnical background, and in particular section and region of the brain were in this context impossible to match appropriately (supplementary file table I). In relation to ethnicity, DS is known to involve relative large numbers and effect sizes of differential methylated loci and expressed genes, which already can be detected by means of principal component analysis (PCA)[17-20]. However, such PCA did not indicate that ethnicity explained substantial variation in the first component. Additional evidence of confounding effects of ethnicity in genome-wide DNA methylation survey is limited and to our knowledge only reported to affect epigenetic biological age estimation, as described by Horvath *et al.* and Philibert *et al.*[21, 22]. Therefore we are convinced that down syndrome itself represents a much higher order of differential DNA methylation and gene expression than effects (if any) that may be expected on the basis of the ethnical differences between groups.

Finally, Sarnat *et al.*(1998) studied the presence of the neuronal nuclear (NeuN) antigen extensively during early developmental stages of the human brain. This study clearly showed that NeuN is poorly represented in gestational ages (GA) from 8 wk up to 14 weeks over the majority of the regions and their variable neuronal cell fraction. During GA range of 14-24 weeks NeuN expression stabilizes for virtual all subtype neuronal cells[23]. Nevertheless, in this study, we cannot exclude that NeuN expression over the different types of neuronal cell differed over GA range and thus may affect cell heterogeneity of samples. In that context, our study holds an explorative character, and results should taken with caution. Since GA itself is prone to a certain error as well, cohort based analysis of prenatal DS, by defintition, cannot provide a perfect match. Validation of our results require therefore translational study follow-up, *in-vitro* and/or, cohort based, single cell analysis.

**Supplementary file table I:** Characteristics of cerebrum specimens qc passed

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Group** | **Sex** | **GA (wks)** | **Section\*** | **Region** | **PMI (hrs)** | **Ethnicity** |
| DS | F | 20 | 19 | *Premotor cortex and Suppl. Motor Cortex* | 2 | AFR-AM |
| DS | F | 19 | 1 | *Primary visual cortex* | 1 | AFR-AM |
| DS | F | 23 | 16 | *Primary Motor Cortex* | 2 | AFR-AM |
| DS | F | 16 | 18 | *Premotor cortex and Suppl. Motor Cortex* | 1 | AFR-AM |
| DS | M | 21 | 17 | *Premotor cortex and Suppl. Motor Cortex* | 2 | AFR-AM |
| DS | M | 22 | 13 | *Primary Somatosensory Cortex* | 3 | CAU |
| DS | M | 19 | 17 | *Premotor cortex and Suppl. Motor Cortex* | 6 | AFR-AM |
| DS | M | 22 | 23 | *Dorsal anterior cingulate cortex* | 3 | AFR-AM |
|  |  |  |  |  |  |  |
| Control | F | 19 | 6 | *Associative visual cortex* | 1 | CAU |
| Control | F | 19 | 7 | *Visuo-Motor Coordination* | 2 | CAU |
| Control | F | 18 | 8 | *Visuo-Motor Coordination* | 2 | CAU |
| Control | F | 19 | 23 | *Dorsal anterior cingulate cortex* | 3 | CAU |
| Control | F | 19 | 6 | *Associative visual cortex* | 1 | CAU |
| Control | F | 18 | 8 | *Visuo-Motor Coordination* | 1 | CAU |
| Control | M | 19 | 7 | *Visuo-Motor Coordination* | 1 | CAU |
| Control | M | 19 | 8 | *Visuo-Motor Coordination* | 2 | CAU |
| Control | M | 19 | 15 | *Primary Motor Cortex* | 4 | CAU |

qc: quality control DNA methylation and gene expression profiles; F: Female; M: Male; GA (wks): gestational age in weeks; PMI: Post mortem interval until tissue preservation; AFR- AM: African American; CAU: Caucasian. \*representing the section within the region of the cerebrum tissue (<https://neurobiobank.nih.gov/about-best-practices>).

***References :***

1. Ahmed ZM, Yousaf R, Lee BC *et al*. Functional null mutations of MSRB3 encoding methionine sulfoxide reductase are associated with human deafness DFNB74*.* *Am J Hum Genet* 88(1), 19-29 (2011).

2. Liu J, Luo X, Xu Y *et al*. Single-stranded DNA binding protein Ssbp3 induces differentiation of mouse embryonic stem cells into trophoblast-like cells*.* *Stem Cell Res Ther* 7(1), 79 (2016).

3. Vacik T, Ladinovic D, Raska I. KDM2A/B lysine demethylases and their alternative isoforms in development and disease*.* *Nucleus* 9(1), 431-441 (2018).

4. Araki T, Milbrandt J. ZNRF proteins constitute a family of presynaptic E3 ubiquitin ligases*.* *J Neurosci* 23(28), 9385-9394 (2003).

5. Gholkar AA, Senese S, Lo YC *et al*. Tctex1d2 associates with short-rib polydactyly syndrome proteins and is required for ciliogenesis*.* *Cell Cycle* 14(7), 1116-1125 (2015).

6. Cingoz S, Agilkaya S, Oztura I *et al*. Identification of the variations in the CPT1B and CHKB genes along with the HLA-DQB1\*06:02 allele in Turkish narcolepsy patients and healthy persons*.* *Genet Test Mol Biomarkers* 18(4), 261-268 (2014).

7. Bosch DG, Boonstra FN, Gonzaga-Jauregui C *et al*. NR2F1 mutations cause optic atrophy with intellectual disability*.* *Am J Hum Genet* 94(2), 303-309 (2014).

8. Malishkevich A, Leyk J, Goldbaum O, Richter-Landsberg C, Gozes I. ADNP/ADNP2 expression in oligodendrocytes: implication for myelin-related neurodevelopment*.* *J Mol Neurosci* 57(2), 304-313 (2015).

9. Isidor B, Le Cunff M, Boceno M *et al*. Complex constitutional subtelomeric 1p36.3 deletion/duplication in a mentally retarded child with neonatal neuroblastoma*.* *Eur J Med Genet* 51(6), 679-684 (2008).

10. Anderson MJ, Pham VN, Vogel AM, Weinstein BM, Roman BL. Loss of unc45a precipitates arteriovenous shunting in the aortic arches*.* *Dev Biol* 318(2), 258-267 (2008).

11. Juriloff DM, Harris MJ. Insights into the Etiology of Mammalian Neural Tube Closure Defects from Developmental, Genetic and Evolutionary Studies*.* *J Dev Biol* 6(3), (2018).

12. Lemieux M, Thiry L, Laflamme OD, Bretzner F. Role of DSCAM in the Development of Neural Control of Movement and Locomotion*.* *Int J Mol Sci* 22(16), (2021).

13. Gurkan H, Atli EI, Atli E *et al*. Chromosomal Microarray Analysis in Turkish Patients with Unexplained Developmental Delay and Intellectual Developmental Disorders*.* *Noro Psikiyatr Ars* 57(3), 177-191 (2020).

14. Utami KH, Winata CL, Hillmer AM *et al*. Impaired development of neural-crest cell-derived organs and intellectual disability caused by MED13L haploinsufficiency*.* *Hum Mutat* 35(11), 1311-1320 (2014).

15. Pongrac JL, Middleton FA, Peng L, Lewis DA, Levitt P, Mirnics K. Heat shock protein 12A shows reduced expression in the prefrontal cortex of subjects with schizophrenia*.* *Biol Psychiatry* 56(12), 943-950 (2004).

16. Startin CM, D'souza H, Ball G *et al*. Health comorbidities and cognitive abilities across the lifespan in Down syndrome*.* *J Neurodev Disord* 12(1), 4 (2020).

17. Henneman P, Bouman A, Mul A *et al*. Widespread domain-like perturbations of DNA methylation in whole blood of Down syndrome neonates*.* *PLoS One* 13(3), e0194938 (2018).

18. Jones MJ, Farre P, Mcewen LM *et al*. Distinct DNA methylation patterns of cognitive impairment and trisomy 21 in Down syndrome*.* *BMC Med Genomics* 6 58 (2013).

19. El Hajj N, Dittrich M, Bock J *et al*. Epigenetic dysregulation in the developing Down syndrome cortex*.* *Epigenetics* 11(8), 563-578 (2016).

20. Laan L, Klar J, Sobol M *et al*. DNA methylation changes in Down syndrome derived neural iPSCs uncover co-dysregulation of ZNF and HOX3 families of transcription factors*.* *Clin Epigenetics* 12(1), 9 (2020).

21. Horvath S, Gurven M, Levine ME *et al*. An epigenetic clock analysis of race/ethnicity, sex, and coronary heart disease*.* *Genome Biol* 17(1), 171 (2016).

22. Philibert R, Beach SRH, Lei MK *et al*. Array-Based Epigenetic Aging Indices May Be Racially Biased*.* *Genes (Basel)* 11(6), (2020).

23. Sarnat HB, Nochlin D, Born DE. Neuronal nuclear antigen (NeuN): a marker of neuronal maturation in early human fetal nervous system*.* *Brain Dev* 20(2), 88-94 (1998).

***R scripts:***The following R scripts were used in quality control, explorative and statistical analysis of the DNA methylation and gene expression profiles of DS and controls:

set.seed(123)

library(doParallel)

registerDoParallel(cores = 16)

##Read raw EPIC data

library(minfi)

targets = read.metharray.sheet("~/L-personal/Down-iHD/DNA/", pattern = "Pheno\_downbrains.csv")

rgSet = read.metharray.exp(targets=targets)

qcReport(rgSet,sampNames = targets$Sample\_Name,sampGroups = targets$Sample\_Group,pdf="~/L-personal/Down-iHD/all/qcReport\_JovanaMaksikovic.pdf")

##Methylaid quality control for detecting outliers

library(MethylAid)

Methylaid.data = summarize(targets, file ="~/L-personal/Down-iHD/all/Methylaid-report")

visualize(Methylaid.data,thresholds=list(epic=list(MU=10.5, OP = 12, BS = 12, HC = 13, DP = 0.95)))

##EPIC

featu = getAnnotation(rgSet)

featu.gr = makeGRangesFromDataFrame(featu, keep.extra.columns = T, seqnames.field = "chr", strand.field = "strand", start.field = "pos", end.field = "pos")

GMset = preprocessNoob(rgSet)

library(Hmisc) #For use of %nin%

autosomes = featu[!featu$chr %in% c("chrX","chrY"), ]

GMset.noXY = GMset[featureNames(GMset) %in% row.names(autosomes),] #Remove XY

popprobes = read.csv("~/L-personal/Down-iHD/DNA/annotation/potentially-polymorphic-targets.csv", sep="\t",header=T)

popprobes$AMR\_AF[popprobes$AMR\_AF < 0.01]=NA #Give population boundaries

popprobes = popprobes[!is.na(popprobes$AMR\_AF),]

GMset.noXY.noPOP = GMset.noXY[which(featureNames(GMset.noXY) %nin% popprobes$IlmnID),]

promprobes = read.csv("~/L-personal/Down-iHD/DNA/annotation/Non-specific-probes-Illumina850k.csv", sep = ",",header = T)

GMset.noXY.noPOP.noPP = GMset.noXY.noPOP[which(featureNames(GMset.noXY.noPOP) %nin% promprobes$TargetID),]

GMset.noXY.noPOP.noPP.noCH = GMset.noXY.noPOP.noPP[grep("ch.", featureNames(GMset.noXY.noPOP.noPP),invert=TRUE),]

beta = getBeta(GMset.noXY.noPOP.noPP.noCH) #Extract betas from desired GMset

colnames(beta) = targets$Sample\_Name

samplenames = targets$Sample\_Name

group = as.factor(targets$Sample\_Group) #Set variables

sex = as.factor(targets$Sex)

section = as.factor(targets$Sec)

design.cor = model.matrix(~0+group+sex+section) #Make design matrix

rownames(design.cor) <- targets$Sample\_Name

library(limma) #For makeContrasts

cont.matrix = makeContrasts(groupDown-groupControl, levels=design.cor) #Set contrasting groups

Mval = getM(GMset.noXY.noPOP.noPP.noCH) #Extract Methylation values from desired GMset

colnames(Mval) = targets$Sample\_Name

beta.rld <- as.matrix(beta)

beta.rld\_cor <- cor(beta.rld)

require(pheatmap)

pheatmap(beta.rld\_cor)

##PCA plots

require("purrr")

beta.centered <- beta-rowMeans(beta)

beta.centered <- beta.centered[apply(beta.centered, 1, compose(is.finite, all)),] # Remove NaN from data

beta.svd <- svd(beta.centered)

beta.svd2 <- svd(round(beta.centered,digits=0))

##CA plot first 8 components

region = as.factor(targets$Region)

ethnicity = as.factor(targets$Ethnicity)

slide = as.factor(targets$Slide)

array = as.factor(targets$Array)

pmi = as.factor(targets$PMI\_h)

group.corr <- sapply(1:ncol(beta.svd$v), function(i){

 fit <- lm(beta.svd$v[,i]~as.factor(group))

 r\_squared <- summary(fit)$adj.r.squared

 p\_value <- pf(summary(fit)$fstatistic[1], summary(fit)$fstatistic[2], summary(fit)$fstatistic[3], lower.tail = F)

 return(data.frame(r\_squared = r\_squared, p\_value = p\_value))})

sex.corr <- sapply(1:ncol(beta.svd$v), function(i){

 fit <- lm(beta.svd$v[,i]~as.factor(sex))

 r\_squared <- summary(fit)$adj.r.squared

 p\_value <- pf(summary(fit)$fstatistic[1], summary(fit)$fstatistic[2], summary(fit)$fstatistic[3], lower.tail = F)

 return(data.frame(r\_squared = r\_squared, p\_value = p\_value))})

section.corr <- sapply(1:ncol(beta.svd$v), function(i){

 fit <- lm(beta.svd$v[,i]~as.factor(section))

 r\_squared <- summary(fit)$adj.r.squared

 p\_value <- pf(summary(fit)$fstatistic[1], summary(fit)$fstatistic[2], summary(fit)$fstatistic[3], lower.tail = F)

 return(data.frame(r\_squared = r\_squared, p\_value = p\_value))})

region.corr <- sapply(1:ncol(beta.svd$v), function(i){

 fit <- lm(beta.svd$v[,i]~as.factor(region))

 r\_squared <- summary(fit)$adj.r.squared

 p\_value <- pf(summary(fit)$fstatistic[1], summary(fit)$fstatistic[2], summary(fit)$fstatistic[3], lower.tail = F)

 return(data.frame(r\_squared = r\_squared, p\_value = p\_value))})

ethnicity.corr <- sapply(1:ncol(beta.svd$v), function(i){

 fit <- lm(beta.svd$v[,i]~as.factor(ethnicity))

 r\_squared <- summary(fit)$adj.r.squared

 p\_value <- pf(summary(fit)$fstatistic[1], summary(fit)$fstatistic[2], summary(fit)$fstatistic[3], lower.tail = F)

 return(data.frame(r\_squared = r\_squared, p\_value = p\_value))})

slide.corr <- sapply(1:ncol(beta.svd$v), function(i){

 fit <- lm(beta.svd$v[,i]~as.factor(slide))

 r\_squared <- summary(fit)$adj.r.squared

 p\_value <- pf(summary(fit)$fstatistic[1], summary(fit)$fstatistic[2], summary(fit)$fstatistic[3], lower.tail = F)

 return(data.frame(r\_squared = r\_squared, p\_value = p\_value))})

array.corr <- sapply(1:ncol(beta.svd$v), function(i){

 fit <- lm(beta.svd$v[,i]~as.factor(array))

 r\_squared <- summary(fit)$adj.r.squared

 p\_value <- pf(summary(fit)$fstatistic[1], summary(fit)$fstatistic[2], summary(fit)$fstatistic[3], lower.tail = F)

 return(data.frame(r\_squared = r\_squared, p\_value = p\_value))})

pmi.corr <- sapply(1:ncol(beta.svd$v), function(i){

 fit <- lm(beta.svd$v[,i]~as.factor(pmi))

 r\_squared <- summary(fit)$adj.r.squared

 p\_value <- pf(summary(fit)$fstatistic[1], summary(fit)$fstatistic[2], summary(fit)$fstatistic[3], lower.tail = F)

 return(data.frame(r\_squared = r\_squared, p\_value = p\_value))})

PCAmatrix = rbind.data.frame(group.corr,sex.corr,section.corr,region.corr,ethnicity.corr,slide.corr,array.corr,pmi.corr)

rownames(PCAmatrix) = c("Group\_r.squared","Group\_p.value","Sex\_r.squared","Sex\_p.value","Section\_r.squared","Section\_p.value","Region\_r.squared","Region\_p.value","Ethnicity\_r.squared","Ethnicity\_p.value","Slide\_r.squared","Slide\_p.value","Array\_r.squared","Array\_p.value","PMI\_r.squared","PMI\_p.value")

colnames(PCAmatrix)[1:ncol(PCAmatrix)]<- paste0("PCA-", 1:ncol(PCAmatrix))

library(data.table)

fwrite(PCAmatrix,"~/L-personal/Down-iHD/all/PCAmatrix.txt",sep="\t",col.names=T,row.names=T)

##See PCA component's contribution

require(ggplot2)

##group

rsq.group <- unlist(t(group.corr))[1:ncol(beta.svd$v)]

pval.group <- unlist(t(group.corr))[(1:ncol(beta.svd$v))+17]#ncol(beta.svd$v)]

group.corr.df <- data.frame(Correlation = rsq.group, Pval = pval.group, PC = 1:length(group.corr))

group.corr.df$padj <- p.adjust(p = group.corr.df$Pval, method = "BH")

group.corr.df$Significance <- ifelse(group.corr.df$padj < 0.05 ,"Significant" ,"Non-significant")

group.corr.df = group.corr.df[1:nrow(targets),]

ggplot(group.corr.df,aes(x=PC,y=Correlation,ymax=0.3,color=Significance))+geom\_point(size=3)+theme\_bw()+ggtitle("Correlation of group")+ylab(expression(R^2))+theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14))

##sex

rsq.sex <- unlist(t(sex.corr))[1:ncol(beta.svd$v)]

pval.sex <- unlist(t(sex.corr))[(1:ncol(beta.svd$v))+17]#ncol(beta.svd$v)]

sex.corr.df <- data.frame(Correlation = rsq.sex, Pval = pval.sex, PC = 1:length(sex.corr))

sex.corr.df$padj <- p.adjust(p = sex.corr.df$Pval, method = "BH")

sex.corr.df$Significance <- ifelse(sex.corr.df$padj < 0.05 ,"Significant" ,"Non-significant")

sex.corr.df = sex.corr.df[1:nrow(targets),]

ggplot(sex.corr.df,aes(x=PC,y=Correlation,ymax=0.3,color=Significance))+geom\_point(size=3)+theme\_bw()+ggtitle("Correlation of sex")+ylab(expression(R^2))+theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14))

##section

rsq.section <- unlist(t(section.corr))[1:ncol(beta.svd$v)]

pval.section <- unlist(t(section.corr))[(1:ncol(beta.svd$v))+17]#ncol(beta.svd$v)]

section.corr.df <- data.frame(Correlation = rsq.section, Pval = pval.section, PC = 1:length(section.corr))

section.corr.df$padj <- p.adjust(p = section.corr.df$Pval, method = "BH")

section.corr.df$Significance <- ifelse(section.corr.df$padj < 0.05 ,"Significant" ,"Non-significant")

section.corr.df = section.corr.df[1:nrow(targets),]

ggplot(section.corr.df,aes(x=PC,y=Correlation,ymax=0.3,color=Significance))+geom\_point(size=3)+theme\_bw()+ggtitle("Correlation of section")+ylab(expression(R^2))+theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14))

##region

rsq.region <- unlist(t(region.corr))[1:ncol(beta.svd$v)]

pval.region <- unlist(t(region.corr))[(1:ncol(beta.svd$v))+17]#ncol(beta.svd$v)]

region.corr.df <- data.frame(Correlation = rsq.region, Pval = pval.region, PC = 1:length(region.corr))

region.corr.df$padj <- p.adjust(p = region.corr.df$Pval, method = "BH")

region.corr.df$Significance <- ifelse(region.corr.df$padj < 0.05 ,"Significant" ,"Non-significant")

region.corr.df = region.corr.df[1:nrow(targets),]

ggplot(region.corr.df,aes(x=PC,y=Correlation,ymax=0.3,color=Significance))+geom\_point(size=3)+theme\_bw()+ggtitle("Correlation of region")+ylab(expression(R^2))+theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14))

##ethnicity

rsq.ethnicity <- unlist(t(ethnicity.corr))[1:ncol(beta.svd$v)]

pval.ethnicity <- unlist(t(ethnicity.corr))[(1:ncol(beta.svd$v))+17]#ncol(beta.svd$v)]

ethnicity.corr.df <- data.frame(Correlation = rsq.ethnicity, Pval = pval.ethnicity, PC = 1:length(ethnicity.corr))

ethnicity.corr.df$padj <- p.adjust(p = ethnicity.corr.df$Pval, method = "BH")

ethnicity.corr.df$Significance <- ifelse(ethnicity.corr.df$padj < 0.05 ,"Significant" ,"Non-significant")

ethnicity.corr.df = ethnicity.corr.df[1:nrow(targets),]

ggplot(ethnicity.corr.df,aes(x=PC,y=Correlation,ymax=0.3,color=Significance))+geom\_point(size=3)+theme\_bw()+ggtitle("Correlation of ethnicity")+ylab(expression(R^2))+theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14))

##pmi

rsq.pmi <- unlist(t(pmi.corr))[1:ncol(beta.svd$v)]

pval.pmi <- unlist(t(pmi.corr))[(1:ncol(beta.svd$v))+17]#ncol(beta.svd$v)]

pmi.corr.df <- data.frame(Correlation = rsq.pmi, Pval = pval.pmi, PC = 1:length(pmi.corr))

pmi.corr.df$padj <- p.adjust(p = pmi.corr.df$Pval, method = "BH")

pmi.corr.df$Significance <- ifelse(pmi.corr.df$padj < 0.05 ,"Significant" ,"Non-significant")

pmi.corr.df = pmi.corr.df[1:nrow(targets),]

ggplot(pmi.corr.df,aes(x=PC,y=Correlation,ymax=0.3,color=Significance))+geom\_point(size=3)+theme\_bw()+ggtitle("Correlation of pmi")+ylab(expression(R^2))+theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14))

##slide

rsq.slide <- unlist(t(slide.corr))[1:ncol(beta.svd$v)]

pval.slide <- unlist(t(slide.corr))[(1:ncol(beta.svd$v))+17]#ncol(beta.svd$v)]

slide.corr.df <- data.frame(Correlation = rsq.slide, Pval = pval.slide, PC = 1:length(slide.corr))

slide.corr.df$padj <- p.adjust(p = slide.corr.df$Pval, method = "BH")

slide.corr.df$Significance <- ifelse(slide.corr.df$padj < 0.05 ,"Significant" ,"Non-significant")

slide.corr.df = slide.corr.df[1:nrow(targets),]

ggplot(slide.corr.df,aes(x=PC,y=Correlation,ymax=0.3,color=Significance))+geom\_point(size=3)+theme\_bw()+ggtitle("Correlation of slide")+ylab(expression(R^2))+theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14))

##array

rsq.array <- unlist(t(array.corr))[1:ncol(beta.svd$v)]

pval.array <- unlist(t(array.corr))[(1:ncol(beta.svd$v))+17]#ncol(beta.svd$v)]

array.corr.df <- data.frame(Correlation = rsq.array, Pval = pval.array, PC = 1:length(array.corr))

array.corr.df$padj <- p.adjust(p = array.corr.df$Pval, method = "BH")

array.corr.df$Significance <- ifelse(array.corr.df$padj < 0.05 ,"Significant" ,"Non-significant")

array.corr.df = array.corr.df[1:nrow(targets),]

ggplot(array.corr.df,aes(x=PC,y=Correlation,ymax=0.3,color=Significance))+geom\_point(size=3)+theme\_bw()+ggtitle("Correlation of array")+ylab(expression(R^2))+theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14))

##PCA plots known covariates

par(mar=c(5.1, 4.1, 4.1, 8.1), xpd=TRUE)

plot(head(group.corr, n=8L),pch=1, col="darkgreen",ylim=c(-0.7,1),xlab="PC's", ylab="correlation")

points(head(sex.corr, n=8L),pch=16, col="magenta")

points(head(section.corr, n=8L),pch=3, col="black")

points(head(region.corr, n=8L),pch=17, col="blue")

points(head(pmi.corr, n=8L),pch=5, col="cyan")

points(head(ethnicity.corr, n=8L),pch=15, col="lightgreen")

points(head(array.corr, n=8L),pch=7, col="red")

points(head(slide.corr, n=8L),pch=4, col="orange")

legend("topright", c("Group","Sex", "Section","Region","PMI","Ethnicity","Array","slide"),col=c("darkgreen","magenta","black","blue","cyan","lightgreen","red","orange"), pch =c(1,16,3,17,5,15,7,4))

##PCA plots known covariates biological

par(mar=c(5.1, 4.1, 4.1, 8.1), xpd=TRUE)

plot(head(group.corr, n=8L),pch=1, col="darkgreen",ylim=c(-0.7,1),xlab="PC's", ylab="correlation")

points(head(sex.corr, n=8L),pch=16, col="magenta")

points(head(section.corr, n=8L),pch=3, col="black")

points(head(region.corr, n=8L),pch=17, col="blue")

points(head(pmi.corr, n=8L),pch=5, col="cyan")

points(head(ethnicity.corr, n=8L),pch=15, col="lightgreen")

legend("topright", inset=c(-0.15,0), c("Group","Sex", "Section","Region","PMI","Ethnicity"),col=c("darkgreen","magenta","black","blue","cyan","lightgreen"), pch =c(1,16,3,17,5,15))

legend("topright", inset=c(-0.15,0), c("Group","Sex", "Section","Region","PMI","Ethnicity"),col=c("darkgreen","magenta","black","blue","cyan","lightgreen"), pch =c(1,16,3,17,5,15))

##PCA plots known covariates technical

par(mar=c(5.1, 4.1, 4.1, 8.1), xpd=TRUE)

plot(head(array.corr, n=8L),pch=7, col="red",ylim=c(-1,1),xlim=c(-0.1,0.7),xlab="PC's", ylab="correlation")

points(head(slide.corr, n=8L),pch=4, col="orange")

legend("topright", c("Array","Slide"),col=c("red","orange"), pch =c(7,4))

##Interactive PCA plot

require(plotly)

PCA.df <- data.frame(PC1 = beta.svd$v[,1], PC2 = beta.svd$v[,2], Cohort = targets$Sample\_Group)

PCA.df.plot <- ggplot(PCA.df, aes(x = PC1, y = PC2, color = Cohort, ID = samplenames, Sex=sex, shape = Cohort)) +

 geom\_point(size = 2) + theme\_bw() + ggtitle("PC1 vs PC2") + xlim(-0.5, 0.5) + ylim(-0.5, 0.5) + ylab("PC2") + xlab("PC1") +

 theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14),legend.title=element\_text(size=14),legend.text=element\_text(size=12),legend.position="right")

gg <- highlight(ggplotly(PCA.df.plot))

crosstalk::bscols(gg)

remove("gg")

DT::datatable(highlight\_key(PCA.df))

##Varaince explained

svd.d <- svd(matrix(rnorm(nrow(beta)\*ncol(beta)), nrow(beta), ncol(beta)))$d

LIM <- range(c(svd.d^2/sum(svd.d^2), beta.svd$d^2/sum(beta.svd$d^2)))

plot(svd.d^2/sum(svd.d^2), ylab = "Variance Explained", xlab = "Principal Component", pch = 16, ylim = LIM)

 points(beta.svd$d^2/sum(beta.svd$d^2), ylab = "Variance Explained", xlab = "Principal Component", pch = 16, col = "red")

 legend("topright", c("Down", "Randomized"), col = c("red", "black"), pch = 16)

##Boxplots of the PCA components

mypar <- function(a=1,b=1,brewer.n=8,brewer.name="Dark2",cex.lab=1,cex.main=1.2,cex.axis=1,mar=c(2.5,2.5,1.6,1.1),mgp=c(1.5,.5,0),...){

 par(mar=mar,mgp=mgp,cex.lab=cex.lab,cex.main=cex.main,cex.axis=cex.axis)

 par(mfrow=c(a,b),...)

 palette(RColorBrewer::brewer.pal(brewer.n,brewer.name))}

mypar(4,6) # row, column

for(i in 1:dim(beta.svd2$v)[1]){

 boxplot(split(beta.svd2$v[,i], targets$Sample\_Group), las = 1, ylim = c(-1,1), range = 0, main = paste0("PC",i))

 stripchart(split(beta.svd2$v[,i], targets$Sample\_Group), add = T, vertical = T, pch = 1, cex = 0.5, col = 1)

 abline(h = 0, cex = 0.5)}

mypar(1,1)#reset par

remove("i")

##Old style PCA plot

library(lumi)

plotSampleRelation(beta, method='mds', color = targets$Sample\_Group, cv.Th=0, xlim=c(-225,200), ylim=c(-150,80), dimension=c(1,2), main="PCA plot")

##Ensembl hg19 annotation

require("dplyr") #For hg19 en hg38 annotation and right/leftjoin

require("tidyverse")

hg19.gtf = fread("~/L-personal/Down-iHD/RNA/annotation/Homo\_sapiens.GRCh37.87.gtf", header = F)

hg19.gtf <- filter(hg19.gtf, V3 == "gene")

hg19.gtf = cbind(hg19.gtf[,1:8], data.frame(do.call('rbind', strsplit(as.character(hg19.gtf$V9), ';', fixed=FALSE))))

hg19.gtf.df = cbind(hg19.gtf[,1],hg19.gtf[,4:5],hg19.gtf[,7],hg19.gtf[,9],hg19.gtf[,11])

names(hg19.gtf.df) = c("chr","start","end","strand","ENS\_ID","Gene\_Name")

hg19.gtf.df$Gene\_Name = gsub("gene\_name ", "", hg19.gtf.df$Gene\_Name)

hg19.gtf.df$Gene\_Name = gsub('"', '', hg19.gtf.df$Gene\_Name)

hg19.gtf.df$Gene\_Name = trimws(hg19.gtf.df$Gene\_Name)

hg19.gtf.df$Gene\_Name = as.factor(hg19.gtf.df$Gene\_Name)

hg19.gtf.df$ENS\_ID = gsub("gene\_id ", "", hg19.gtf.df$ENS\_ID)

hg19.gtf.df$ENS\_ID = gsub('"', '', hg19.gtf.df$ENS\_ID)

hg19.gtf.df$ENS\_ID = trimws(hg19.gtf.df$ENS\_ID)

hg19.gtf.df = hg19.gtf.df %>% filter(str\_detect(chr, "GL", negate = TRUE))

hg19.gtf.df = hg19.gtf.df %>% filter(str\_detect(chr, "MT", negate = TRUE))

hg19.gtf.df$chr = paste("chr",hg19.gtf.df$chr, sep = "")

hg19.gtf.gr = makeGRangesFromDataFrame(hg19.gtf.df, keep.extra.columns = T, seqnames.field = "chr")

##Ensembl hg38 annotation

hg38.gtf = fread("~/L-personal/Down-iHD/RNA/annotation/Homo\_sapiens.GRCh38.97.gtf", header = F)

hg38.gtf <- filter(hg38.gtf, V3 == "gene")

hg38.gtf = cbind(hg38.gtf[,1:8], data.frame(do.call('rbind', strsplit(as.character(hg38.gtf$V9), ';', fixed=FALSE))))

hg38.gtf.df = cbind(hg38.gtf[,1],hg38.gtf[,4:5],hg38.gtf[,7],hg38.gtf[,9],hg38.gtf[,11])

names(hg38.gtf.df) = c("chr","start","end","strand","ENS\_ID","Gene\_Name")

hg38.gtf.df$Gene\_Name = gsub("gene\_name ", "", hg38.gtf.df$Gene\_Name)

hg38.gtf.df$Gene\_Name = gsub('"', '', hg38.gtf.df$Gene\_Name)

hg38.gtf.df$Gene\_Name = trimws(hg38.gtf.df$Gene\_Name)

hg38.gtf.df$Gene\_Name = as.factor(hg38.gtf.df$Gene\_Name)

hg38.gtf.df$ENS\_ID = gsub("gene\_id ", "", hg38.gtf.df$ENS\_ID)

hg38.gtf.df$ENS\_ID = gsub('"', '', hg38.gtf.df$ENS\_ID)

hg38.gtf.df$ENS\_ID = trimws(hg38.gtf.df$ENS\_ID)

hg38.gtf.df = hg38.gtf.df %>% filter(str\_detect(chr, "GL", negate = TRUE))

hg38.gtf.df = hg38.gtf.df %>% filter(str\_detect(chr, "MT", negate = TRUE))

hg38.gtf.df = hg38.gtf.df %>% filter(str\_detect(chr, "KI", negate = TRUE))

hg38.gtf.df$chr = paste("chr",hg38.gtf.df$chr, sep = "")

hg38.gtf.gr = makeGRangesFromDataFrame(hg38.gtf.df, keep.extra.columns = T, seqnames.field = "chr")

##Overlap between EPIC annotation and ensembl gtf annotation

library(plyr)

featu.hg19.gtf.overlap = as.data.frame(findOverlaps(featu.gr,hg19.gtf.gr, ignore.strand=TRUE)) #Because the UCSC gene annotation seems not to include strand, for linking Ensembl gene aslo strand is ignored

idx.gtf <- featu.hg19.gtf.overlap$subjectHits

idx.featu = featu.hg19.gtf.overlap$queryHits

featu.hg19.gtf.gene = cbind(idx.featu, as.data.frame(hg19.gtf.gr$Gene\_Name[idx.gtf],))

featu.hg19.gtf.gene = unique(featu.hg19.gtf.gene)

colnames(featu.hg19.gtf.gene) = c("ID","Gene\_Name")

featu.hg19.gtf.gene.list <- ddply(featu.hg19.gtf.gene, .(ID), summarize, Gene\_Name=paste(Gene\_Name,collapse=",")) #library(plyr)

rownames(featu.hg19.gtf.gene.list) = featu.hg19.gtf.gene.list$ID

featu.hg19.gtf.gene.list$ID = NULL

featu.df = as.data.frame(featu)

rownames(featu.df) <- c()

tmp = merge(featu.df, featu.hg19.gtf.gene.list, by = 0, all=TRUE,sort = FALSE)

rownames(featu.df) = featu.df$Name

tmp = tmp[match(row.names(featu.df), tmp$Name),]

featu.df$Gene\_Name = tmp$Gene\_Name

featu.df$Gene\_Name[is.na(featu.df$Gene\_Name)] <- ""

featu.df = featu.df %>% select(1:4, Gene\_Name, everything())

featu.df = featu.df %>% select(1:5, UCSC\_RefGene\_Name, everything())

remove("idx.gtf","idx.featu","tmp")

##DMRcate analysis, strating with fdr = 0.05

require(tidyr)

library(DMRcate)

DMRcate.cpg = cpg.annotate(Mval, datatype = "array", design = design.cor, arraytype = "EPIC", analysis.type = "differential", what = "M", contrasts = T, cont.matrix = cont.matrix, coef = "groupDown - groupControl")

DMRcate.hg19 = extractRanges(dmrcate(DMRcate.cpg, lambda = 1000, C=2),genome="hg19")

DMRcate.hg19 = as.data.frame(DMRcate.hg19)

DMRcate.hg19 = DMRcate.hg19[,1:12]

DMRcate.hg19.gr = makeGRangesFromDataFrame(DMRcate.hg19, keep.extra.columns = T)

DMRcate.hg19.overlap = as.data.frame(findOverlaps(DMRcate.hg19.gr,hg19.gtf.gr, maxgap = 2000))

idx.gene <- DMRcate.hg19.overlap$subjectHits

idx.dmr = DMRcate.hg19.overlap$queryHits

DMRcate.hg19.gene = cbind(DMRcate.hg19.overlap$queryHits, as.data.frame(hg19.gtf.gr$Gene\_Name[idx.gene],))

DMRcate.hg19.gene = unique(DMRcate.hg19.gene)

colnames(DMRcate.hg19.gene) = c("ID","Gene\_Name")

DMRcate.hg19.gene.list <- ddply(DMRcate.hg19.gene, .(ID), summarize, Gene\_Name=paste(Gene\_Name,collapse=","))

rownames(DMRcate.hg19.gene.list) = DMRcate.hg19.gene.list$ID

tmp = merge(DMRcate.hg19, DMRcate.hg19.gene.list, by = 0, all=TRUE,sort = FALSE)

DMRcate\_hg19 = tmp[2:15][-13][order(tmp$Stouffer),]

write.table(DMRcate\_hg19,"~/L-personal/Down-iHD/all/DMRcate\_fdr0.05\_hg19ensembl.txt", sep="\t",col.names=T,row.names=F) # write data to file

DMRcate\_unique.hg19 = separate\_rows(DMRcate\_hg19,Gene\_Name, sep = ",", convert = TRUE)

DMRcate\_unique.hg19$Gene\_Name = trimws(DMRcate\_unique.hg19$Gene\_Name)

dmrcate\_unique.hg19.gr = makeGRangesFromDataFrame(DMRcate\_unique.hg19, keep.extra.columns = T)

remove("idx.gene","idx.dmr","tmp")

##Cate 0.1

DMRcate\_01.cpg = cpg.annotate(Mval, datatype = "array", fdr = 0.1, design = design.cor, arraytype = "EPIC", analysis.type = "differential", what = "M", contrasts = T, cont.matrix = cont.matrix, coef = "groupDown - groupControl")

DMRcate\_01.hg19 = extractRanges(dmrcate(DMRcate\_01.cpg, lambda = 1000, C=2),genome="hg19")

DMRcate\_01.hg19 = as.data.frame(DMRcate\_01.hg19)

DMRcate\_01.hg19 = DMRcate\_01.hg19[,1:12]

DMRcate\_01.hg19.gr = makeGRangesFromDataFrame(DMRcate\_01.hg19, keep.extra.columns = T)

DMRcate\_01.hg19.overlap = as.data.frame(findOverlaps(DMRcate\_01.hg19.gr,hg19.gtf.gr, maxgap = 2000))

idx.gene <- DMRcate\_01.hg19.overlap$subjectHits

idx.dmr = DMRcate\_01.hg19.overlap$queryHits

DMRcate\_01.hg19.gene = cbind(DMRcate\_01.hg19.overlap$queryHits, as.data.frame(hg19.gtf.gr$Gene\_Name[idx.gene],))

DMRcate\_01.hg19.gene = unique(DMRcate\_01.hg19.gene)

colnames(DMRcate\_01.hg19.gene) = c("ID","Gene\_Name")

DMRcate\_01.hg19.gene.list <- ddply(DMRcate\_01.hg19.gene, .(ID), summarize, Gene\_Name=paste(Gene\_Name,collapse=","))

rownames(DMRcate\_01.hg19.gene.list) = DMRcate\_01.hg19.gene.list$ID

tmp = merge(DMRcate\_01.hg19, DMRcate\_01.hg19.gene.list, by = 0, all=TRUE,sort = FALSE)

DMRcate\_01.hg19 = tmp[2:15][-13][order(tmp$Stouffer),]

write.table(DMRcate\_01.hg19,"~/L-personal/Down-iHD/all/DMRcate\_fdr0.1\_hg19ensembl.txt", sep="\t",col.names=T,row.names=F) # write data to file

DMRcate\_01\_unique.hg19 = separate\_rows(DMRcate\_01.hg19,Gene\_Name, sep = ",", convert = TRUE)

DMRcate\_01\_unique.hg19$Gene\_Name = trimws(DMRcate\_01\_unique.hg19$Gene\_Name)

dmrcate\_01\_unique.hg19.gr = makeGRangesFromDataFrame(DMRcate\_01\_unique.hg19, keep.extra.columns = T)

remove("idx.gene","idx.dmr","tmp")

##Cate 0.5

DMRcate\_05.cpg = cpg.annotate(Mval, datatype = "array", fdr = 0.5, design = design.cor, arraytype = "EPIC", analysis.type = "differential", what = "M", contrasts = T, cont.matrix = cont.matrix, coef = "groupDown - groupControl")

DMRcate\_05.hg19 = extractRanges(dmrcate(DMRcate\_05.cpg, lambda = 1000, C=2),genome="hg19")

DMRcate\_05.hg19 = as.data.frame(DMRcate\_05.hg19)

DMRcate\_05.hg19 = DMRcate\_05.hg19[,1:12]

DMRcate\_05.hg19.gr = makeGRangesFromDataFrame(DMRcate\_05.hg19, keep.extra.columns = T)

DMRcate\_05.hg19.overlap = as.data.frame(findOverlaps(DMRcate\_05.hg19.gr,hg19.gtf.gr, maxgap = 2000))

idx.gene <- DMRcate\_05.hg19.overlap$subjectHits

idx.dmr = DMRcate\_05.hg19.overlap$queryHits

DMRcate\_05.hg19.gene = cbind(DMRcate\_05.hg19.overlap$queryHits, as.data.frame(hg19.gtf.gr$Gene\_Name[idx.gene],))

DMRcate\_05.hg19.gene = unique(DMRcate\_05.hg19.gene)

colnames(DMRcate\_05.hg19.gene) = c("ID","Gene\_Name")

DMRcate\_05.hg19.gene.list <- ddply(DMRcate\_05.hg19.gene, .(ID), summarize, Gene\_Name=paste(Gene\_Name,collapse=","))

rownames(DMRcate\_05.hg19.gene.list) = DMRcate\_05.hg19.gene.list$ID

tmp = merge(DMRcate\_05.hg19, DMRcate\_05.hg19.gene.list, by = 0, all=TRUE,sort = FALSE)

DMRcate\_05.hg19 = tmp[2:15][-13][order(tmp$Stouffer),]

write.table(DMRcate\_05.hg19,"~/L-personal/Down-iHD/all/DMRcate\_fdr0.5\_hg19ensembl.txt", sep="\t",col.names=T,row.names=F) # write data to file

DMRcate\_05\_unique.hg19 = separate\_rows(DMRcate\_05.hg19,Gene\_Name, sep = ",", convert = TRUE)

DMRcate\_05\_unique.hg19$Gene\_Name = trimws(DMRcate\_05\_unique.hg19$Gene\_Name)

dmrcate\_05\_unique.hg19.gr = makeGRangesFromDataFrame(DMRcate\_05\_unique.hg19, keep.extra.columns = T)

remove("idx.gene","idx.dmr","tmp")

##RNA data en statistiek RNA reads

require(DESeq2)

library(ggrepel)

gcounts <- data.frame(fread("~/L-personal/Down-iHD/RNA/output/counts/counts.txt",sep = "\t"), row.names=1)

gcounts = gcounts[,c(6:8,10,12:24)]

colnames(gcounts) = targets$Sample\_Name

gcounts\_clean <- gcounts[which(!rowSums(gcounts == 0)>ncol(gcounts)\*0.75),]

tmp = gcounts\_clean

tmp$ENS\_ID = rownames(tmp)

tmp2 = right\_join(hg38.gtf.df[,c("ENS\_ID", "Gene\_Name")], tmp, by = c("ENS\_ID" = "ENS\_ID"))

tmp2 = tmp2[!duplicated(tmp2$Gene\_Name),]

tmp2 = as.data.frame(tmp2[complete.cases(tmp2$Gene\_Name), ])

row.names(tmp2) = tmp2$Gene\_Name

RNA = as.matrix(tmp2[,3:19])

remove("tmp","tmp2")

dds <- DESeqDataSetFromMatrix(countData = gcounts\_clean, colData = targets, design = design.cor)

dds.deseq <- DESeq(dds, fitType="local")

ddsresults = as.data.frame(results(dds.deseq, contrast=list("groupControl","groupDown")))

ddsresults <- ddsresults[!is.na(ddsresults$padj),]

ddsresults$ENS\_ID = row.names(ddsresults)

ddsresults <- right\_join(hg38.gtf.df[,c("ENS\_ID", "Gene\_Name")], ddsresults, by = c("ENS\_ID" = "ENS\_ID"))

ddsresults.df = cbind(ddsresults,hg38.gtf.df$start[match(ddsresults$Gene\_Name,hg38.gtf.df$Gene\_Name)])

ddsresults.df = cbind(ddsresults.df,hg38.gtf.df$end[match(ddsresults.df$Gene\_Name,hg38.gtf.df$Gene\_Name)])

ddsresults.df = cbind(ddsresults.df,hg38.gtf.df$chr[match(ddsresults.df$Gene\_Name,hg38.gtf.df$Gene\_Name)])

names(ddsresults.df)[9:11] = c("start", "end","chr")

ddsresults.df = ddsresults.df[complete.cases(ddsresults.df$start), ]

ddsresults.df <- as.data.frame(ddsresults.df[order(ddsresults.df$pvalue),])

row.names(ddsresults.df) = ddsresults.df$ENS\_ID

ddsresults.gr = makeGRangesFromDataFrame(ddsresults.df, keep.extra.columns = T, seqnames.field = "chr")

write.table(ddsresults.df, "~/L-personal/Down-iHD/all/ddsresults\_hg38.txt", sep = "\t", row.names = F)

source("~/L-personal/Down-iHD/Volcanoplot.R")

RNA\_volcano = volcano\_plot(effect\_sizes=ddsresults$log2FoldChange,pvals=ddsresults$pvalue,significance=ddsresults$padj<0.05,int\_effect\_threshold=20,top\_names=10,identifiers=ddsresults$Gene\_Name)

print(RNA\_volcano)

##Print top 10 genes in boxplots and heatmap

require("NDlib")

rld <- rlog(dds)

rld\_counts <- assay(rld)

##Heatmap all and heatmap of reads data combined with q.values from the ddsresults

rld\_cor <- cor(rld\_counts)

pheatmap(rld\_cor)

tmp = cbind(row.names(gcounts\_clean),gcounts\_clean[1:17])

colnames(tmp)[1] = "ENS\_ID"

tmp2 = right\_join(tmp, ddsresults.df[,c("ENS\_ID","padj")], by = c("ENS\_ID" = "ENS\_ID"))

tmp2=tmp2[order(tmp2$padj),]

tmp2=tmp2[1:nrow(tmp2[tmp2$padj<0.05,]),2:18]

tmp2 <- as.matrix(tmp2)

tmp3 <- cor(tmp2)

pheatmap(tmp3)

remove("tmp","tmp2","tmp3")

##For printing top 10 gene hits boxplots

rld\_dm <- rld\_counts-rowMeans(rld\_counts)

rld\_svd <- svd(t(rld\_dm))

rld\_svd\_pct\_var <- round(rld\_svd$d/sum(rld\_svd$d)\*100, 2)

rld\_svd\_df <- data.frame(PC1 = rld\_svd$u[,1],PC2 = rld\_svd$u[,2],Sample\_Group=colData(rld)$Sample\_Group,Cohort = targets$Sample\_Group)# Gender = colData(rld)$Gender, Ethnicity = colData(rld)$Ethnicity)

for(i in 1:10){

 jpg = paste("~/L-personal/Down-iHD/all/Tophit\_dds-",i,"\_",ddsresults.df[i,"Gene\_Name"],".jpg", sep="")

 jpeg(jpg)

 print(transcript\_strip\_plot(id=ddsresults.df[i,"ENS\_ID"],counts=rld\_counts,factor\_interest=colData(rld)$Sample\_Group,title=ddsresults.df[i,"Gene\_Name"],type="SE"))

 dev.off()}

remove("i","jpg")

## RNA PCA plot

PCA\_RNA <- ggplot(rld\_svd\_df, aes(x = PC1, y = PC2, col = Sample\_Group)) +

 geom\_point(size = 3) +

 theme\_bw() +

 xlab(paste0("PC1 (", rld\_svd\_pct\_var[1], "%)")) +

 ylab(paste0("PC2 (", rld\_svd\_pct\_var[2], "%)")) +

 theme(legend.pos = "bottom")

print(PCA\_RNA)

## Interactive version

m <- highlight\_key(rld\_svd\_df)

PCA.RNA.df.plot <- ggplot(rld\_svd\_df, aes(x = PC1, y = PC2, color = Cohort, ID = samplenames, Sex=sex, shape = Cohort)) +

 geom\_point(size = 2) + theme\_bw() + ggtitle("PC1 vs PC2") + xlim(-0.7, 0.7) + ylim(-0.7, 0.7) + ylab("PC2") + xlab("PC1") +

 theme(axis.text=element\_text(size=12),axis.title=element\_text(size=14),legend.title=element\_text(size=14),legend.text=element\_text(size=12),legend.position="right")

gg <- highlight(ggplotly(PCA.RNA.df.plot))

crosstalk::bscols(gg)

remove("gg")

DT::datatable(highlight\_key(rld\_svd\_df))

##Annotation and eQTM analsye

require(eqtm)

eqtm.annot.gr = makeGRangesFromDataFrame(getAnnotation(GMset.noXY.noPOP.noPP.noCH), start.field = "pos", end.field = "pos", keep.extra.columns = T)

eqtm.hg19<- eqtm(dmrs\_gr = dmrcate\_unique.hg19.gr,

 gene\_col = "Gene\_Name", # gene\_col gaat altijd over de gen naam kolom in alleen de dmr regios

 meth\_data = beta,

 expr\_data = RNA,

 meth\_anno\_gr = eqtm.annot.gr, #Gemaakt van beta annotatie

 cor\_method = "pearson",

 aggregation\_method = "median",

 alternative = "two.sided", #One sided is nooit bekeken, werkt misschien ook niet

 N = 100000, #Vergroot aantal iteraties \*10 voor aantal cijfers achter komma van pval te vergroten

 iseed = 531235,

 ncores = 8)

eqtms.hg19 <- data.frame(rowRanges(eqtm.hg19))

eqtms.hg19 <- eqtms.hg19[order(eqtms.hg19$nCpGs, decreasing = T),]

eqtms.topcor.hg19 <- eqtm::topcor(eqtm.hg19, sort.by = "pval")

write.table(eqtms.topcor.hg19, "~/L-personal/Down-iHD/all/eqtm\_dmrcate\_sex\_sec\_N100000fdr0.05\_hg19.txt", sep = "\t", row.names = F)

##cate 0.1

eqtm\_01.hg19 <- eqtm(dmrs\_gr = dmrcate\_01\_unique.hg19.gr,

 gene\_col = "Gene\_Name", # gene\_col gaat altijd over de gen naam kolom in alleen de dmr regios

 meth\_data = beta,

 expr\_data = RNA,

 meth\_anno\_gr = eqtm.annot.gr, #Gemaakt van beta annotatie

 cor\_method = "pearson",

 aggregation\_method = "median",

 alternative = "two.sided", #One sided is nooit bekeken, werkt misschien ook niet

 N = 100000, #Vergroot aantal iteraties \*10 voor aantal cijfers achter komma van pval te vergroten

 iseed = 531235,

 ncores = 16)

eqtms\_01.hg19 <- data.frame(rowRanges(eqtm\_01.hg19))

eqtms\_01.hg19 <- eqtms\_01.hg19[order(eqtms\_01.hg19$nCpGs, decreasing = T),]

eqtms\_01.topcor.hg19 <- eqtm::topcor(eqtm\_01.hg19, sort.by = "pval")

write.table(eqtms\_01.topcor.hg19, "~/L-personal/Down-iHD/all/eqtm\_dmrcate\_sex\_sec\_N100000fdr0.1\_hg19.txt", sep = "\t", row.names = F)

##cate 0.5

eqtm\_05.hg19 <- eqtm(dmrs\_gr = dmrcate\_05\_unique.hg19.gr,

 gene\_col = "Gene\_Name", # gene\_col gaat altijd over de gen naam kolom in alleen de dmr regios

 meth\_data = beta,

 expr\_data = RNA,

 meth\_anno\_gr = eqtm.annot.gr, #Gemaakt van beta annotatie

 cor\_method = "pearson",

 aggregation\_method = "median",

 alternative = "two.sided", #One sided is nooit bekeken, werkt misschien ook niet

 N = 100000, #Vergroot aantal iteraties \*10 voor aantal cijfers achter komma van pval te vergroten

 iseed = 531235,

 ncores = 32)

eqtms\_05.hg19 <- data.frame(rowRanges(eqtm\_05.hg19))

eqtms\_05.hg19 <- eqtms\_05.hg19[order(eqtms\_05.hg19$nCpGs, decreasing = T),]

eqtms\_05.topcor.hg19 <- eqtm::topcor(eqtm\_05.hg19, sort.by = "pval")

write.table(eqtms\_05.topcor.hg19, "~/L-personal/Down-iHD/all/eqtm\_dmrcate\_sex\_sec\_N100000fdr0.5\_hg19.txt", sep = "\t", row.names = F)

##LMfit

lmfit.cor = lmFit(beta, design.cor, method="ls") #Lineair model correction, set method as desired (or use ls)

lmfit.cor.contrast = contrasts.fit(lmfit.cor,cont.matrix)

lmfit.cor.contrast.eB = eBayes(lmfit.cor.contrast) #Bayes to moderate the standard deviations between genes

lmfit.cor.contrast.eB.fdr = topTable(lmfit.cor.contrast.eB, n=nrow(beta), adjust.method="fdr")

lmfit.cor.contrast.eB.BH = topTable(lmfit.cor.contrast.eB, n=nrow(beta), adjust.method="BH")

lmfit = cbind(lmfit.cor.contrast.eB.fdr, lmfit.cor.contrast.eB.BH[5])

colnames(lmfit)[7] = "BH"

beta.df = as.data.frame(beta)

colnames(beta.df) = targets$Sample\_Group

beta.df$Delta = c(rowMeans(beta.df[,colnames(beta.df)=="Down"]) - rowMeans(beta.df[,colnames(beta.df)=="Control"])) #Calculate Delta

colnames(beta.df) = as.character(c(targets$Sample\_Name,"Delta"))

tmp = data.frame(beta.df[match(row.names(lmfit), row.names(beta.df)),])

lmfit = cbind(lmfit,tmp$Delta)

colnames(lmfit)[colnames(lmfit)=="tmp$Delta"] = "Delta"

tmp = featu.df[match(row.names(lmfit), row.names(featu.df)),]

lmfit = cbind(lmfit,tmp)

lmfit.gr = makeGRangesFromDataFrame(lmfit, keep.extra.columns = TRUE, seqnames.field = "chr", strand.field = "strand", start.field = "pos", end.field = "pos")

write.table(lmfit,"~/L-personal/Down-iHD/all/lmfit\_corr-sex-sec.txt", sep="\t",col.names=T,row.names=T) # write data to file

remove("tmp")

##Betas visualized

plot(x = beta.df$Delta, y = -log10(beta.df$P.Value), #Specify q-value or P.Value

 cex = 0.5, pch = 21, bg = "red", xlab = "Delta", ylab = "-log10(pVal)", #Depending on te above, enter p/q-val

 xlim = c(-1,1), ylim = c(0, 10))

abline(h = -log10(exp(-10)), lty = 1) #Specify threshold for p/q-value

abline(v = 0.3, lty = 1)

abline(v = -0.3, lty = 1)

##Volcanoplot

volcano\_plot(lmfit$logFC, lmfit$P.Value, lmfit$adj.P.Val < 0.05, lmfit$Delta, int\_effect\_threshold=-0.7)

##Heatmap of beta data combined with lmfit q.values

library(heatmap3)

tmp = data.frame(lmfit[match(row.names(beta.df), row.names(lmfit)),])

beta.df=cbind(as.data.frame(beta.df),tmp$P.Value, tmp$adj.P.Val,tmp$chr)

colnames(beta.df)[19:21] = c("P.Value","q.value","chr")

tmp2=beta.df[order(beta.df$q.value),]

tmp2=tmp2[1:nrow(tmp2[tmp2$q.value<0.05,]),1:17]

my\_palette <- colorRampPalette(c("red", "orange", "yellow", "white", "lightblue","blue","violet"))(n = 7)

heatmap3(tmp2, ColSideLabs=colnames(tmp2), Colv = NULL, showColDendro = T, showRowDendro = F,

 col = my\_palette, method = "ward.D", RowSideLabs="", cexRow = 0.001, breaks=seq(-5, 5, by=1.4),

 ylab = "CpG's with qValue < 0.1")

tmp2 <- as.matrix(tmp2)

tmp3 <- cor(tmp2)

pheatmap(tmp3)

remove("tmp","tmp2","tmp3")

##Pathway analysis RNA

require(qusage)

library(fgsea)

stats\_RNA = ddsresults.df$stat

names(stats\_RNA) <- ddsresults.df$Gene\_Name

stats\_RNA = stats\_RNA[order(-stats\_RNA)]

## Let op, er zijn meerdere ENS\_IDs voor 1 gen, dus 1 gen komt meerdere keren voor. Hierdoor moet gefiterd worden op de meest significante

stats\_RNA = stats\_RNA[!duplicated(names(stats\_RNA))]

GO\_file = "~/L-personal/Down-iHD/GSEA/h.all.v7.0.symbols.gmt"

hallmark = read.gmt(GO\_file)

fgseaRNA <- fgsea(pathways=hallmark, stats=stats\_RNA, nperm=1000000)

fgseaRNA <- fgseaRNA[order(fgseaRNA$pval),]

fgseaRNA.sig <- fgseaRNA[which(fgseaRNA$padj<0.05),]$pathway

fgseaRNA$leadingEdge = gsub('c\\(',"",fgseaRNA$leadingEdge)

fgseaRNA$leadingEdge = gsub('\\)',"",fgseaRNA$leadingEdge)

fgseaRNA$leadingEdge = gsub('\\"',"",fgseaRNA$leadingEdge)

colnames(fgseaRNA)[8] = "Gene\_Name"

write.table(fgseaRNA,"~/L-personal/Down-iHD/all/Pathways\_RNA.txt", sep="\t",col.names=T,row.names=F) # write data to file

plotGseaTable(pathways = hallmark[fgseaRNA.sig], stats = stats\_RNA, fgseaRes = fgseaRNA, gseaParam = 0.5)

##Pathways analysis lmfit

tmp = lmfit[!duplicated(lmfit$Gene\_Name),]

stats\_lmfit= unique(tmp$t)

stats\_lmfit = as.vector(stats\_lmfit[order(-stats\_lmfit)])

names(stats\_lmfit) = tmp$Gene\_Name

fgsea\_lmfit = fgsea(pathways=hallmark, stats=stats\_lmfit, nperm=1000000)

fgsea\_lmfit <- fgsea\_lmfit[order(fgsea\_lmfit$pval),]

fgsea\_lmfit.sig <- fgsea\_lmfit[which(fgsea\_lmfit$padj<0.00001),]$pathway

fgsea\_lmfit$leadingEdge = gsub('c\\(',"",fgsea\_lmfit$leadingEdge)

fgsea\_lmfit$leadingEdge = gsub('\\)',"",fgsea\_lmfit$leadingEdge)

fgsea\_lmfit$leadingEdge = gsub('\\"',"",fgsea\_lmfit$leadingEdge)

colnames(fgsea\_lmfit)[8] = "Gene\_Name"

write.table(fgsea\_lmfit,"~/L-personal/Down-iHD/all/Pathways\_lmfit.txt", sep="\t",col.names=T,row.names=F) # write data to file

remove("tmp")

plotGseaTable(pathways = hallmark[fgsea\_lmfit.sig], stats = stats\_lmfit, fgseaRes = fgsea\_lmfit, gseaParam = 0.5)

##Pathway analysis dmrcate

tmp = DMRcate\_unique.hg19[!duplicated(DMRcate\_unique.hg19$Gene\_Name),]

stats\_epic = tmp$HMFDR

names(stats\_epic) = tmp$Gene\_Name

stats\_epic = stats\_epic[order(-stats\_epic)]

fgseaEPIC <- fgsea(pathways=hallmark, stats=stats\_epic, nperm=1000000) #There are ties in the preranked stats (34.27% of the list).

fgseaEPIC <- fgseaEPIC[order(fgseaEPIC$pval),]

fgseaEPIC.sig <- fgseaEPIC[which(fgseaEPIC$padj<0.5),]$pathway

fgseaEPIC$leadingEdge = gsub('c\\(',"",fgseaEPIC$leadingEdge)

fgseaEPIC$leadingEdge = gsub('\\)',"",fgseaEPIC$leadingEdge)

fgseaEPIC$leadingEdge = gsub('\\"',"",fgseaEPIC$leadingEdge)

colnames(fgseaEPIC)[8] = "Gene\_Name"

write.table(fgseaEPIC,"~/L-personal/Down-iHD/all/Pathways\_Cate\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

remove("tmp")

plotGseaTable(pathways = hallmark[fgseaEPIC.sig], stats = stats\_epic, fgseaRes = fgseaEPIC, gseaParam = 0.5)

##cate 0.1

tmp = DMRcate\_01\_unique.hg19[!duplicated(DMRcate\_01\_unique.hg19$Gene\_Name),]

stats\_epic\_01 = tmp$HMFDR

names(stats\_epic\_01) = tmp$Gene\_Name

stats\_epic\_01 = stats\_epic\_01[order(-stats\_epic\_01)]

fgseaEPIC\_01 <- fgsea(pathways=hallmark, stats=stats\_epic\_01, nperm=1000000) #There are ties in the preranked stats (28.38% of the list).

fgseaEPIC\_01 <- fgseaEPIC\_01[order(fgseaEPIC\_01$pval),]

fgseaEPIC\_01.sig <- fgseaEPIC\_01[which(fgseaEPIC\_01$pval<0.05),]$pathway

fgseaEPIC\_01$leadingEdge = gsub('c\\(',"",fgseaEPIC\_01$leadingEdge)

fgseaEPIC\_01$leadingEdge = gsub('\\)',"",fgseaEPIC\_01$leadingEdge)

fgseaEPIC\_01$leadingEdge = gsub('\\"',"",fgseaEPIC\_01$leadingEdge)

colnames(fgseaEPIC\_01)[8] = "Gene\_Name"

write.table(fgseaEPIC\_01,"~/L-personal/Down-iHD/all/Pathways\_Cate\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

remove("tmp")

plotGseaTable(pathways = hallmark[fgseaEPIC\_01.sig], stats = stats\_epic\_01, fgseaRes = fgseaEPIC\_01, gseaParam = 0.5)

##cate 0.5

tmp = DMRcate\_05\_unique.hg19[!duplicated(DMRcate\_05\_unique.hg19$Gene\_Name),]

stats\_epic\_05 = tmp$HMFDR

names(stats\_epic\_05) = tmp$Gene\_Name

stats\_epic\_05 = stats\_epic\_05[order(-stats\_epic\_05)]

fgseaEPIC\_05 <- fgsea(pathways=hallmark, stats=stats\_epic\_05, nperm=1000000) # There are ties in the preranked stats (24.8% of the list).

fgseaEPIC\_05 <- fgseaEPIC\_05[order(fgseaEPIC\_05$pval),]

fgseaEPIC\_05.sig <- fgseaEPIC\_05[which(fgseaEPIC\_05$pval<0.05),]$pathway

fgseaEPIC\_05$leadingEdge = gsub('c\\(',"",fgseaEPIC\_05$leadingEdge)

fgseaEPIC\_05$leadingEdge = gsub('\\)',"",fgseaEPIC\_05$leadingEdge)

fgseaEPIC\_05$leadingEdge = gsub('\\"',"",fgseaEPIC\_05$leadingEdge)

colnames(fgseaEPIC\_05)[8] = "Gene\_Name"

write.table(fgseaEPIC\_05,"~/L-personal/Down-iHD/all/Pathways\_Cate\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

remove("tmp")

plotGseaTable(pathways = hallmark[fgseaEPIC\_05.sig], stats = stats\_epic\_05, fgseaRes = fgseaEPIC\_05, gseaParam = 0.5)

##Overlap DMRcate and expression data with rightjoin script, links gene name

overlap = right\_join(ddsresults.df, DMRcate\_unique.hg19[,c("Gene\_Name","Stouffer")], by = c("Gene\_Name" = "Gene\_Name"))

overlap = overlap[complete.cases(overlap$pvalue), ]

overlap.unsig = subset(overlap, overlap$pvalue > 0.05 & overlap$Stouffer > 0.05)

overlap.sig.expr = subset(overlap, overlap$pvalue < 0.05 & overlap$Stouffer > 0.05)

overlap.sig.meth = subset(overlap, overlap$Stouffer < 0.05 & overlap$pvalue > 0.05)

overlap.sig.expr.meth = subset(overlap, overlap$pvalue < 0.05 & overlap$Stouffer < 0.05)

##cate 0.1

overlap0.1 = right\_join(ddsresults.df, DMRcate\_01\_unique.hg19[,c("Gene\_Name","Stouffer")], by = c("Gene\_Name" = "Gene\_Name"))

overlap0.1 = overlap0.1[complete.cases(overlap0.1$pvalue), ]

overlap.unsig0.1 = subset(overlap0.1, overlap0.1$pvalue > 0.05 & overlap0.1$Stouffer > 0.05)

overlap.sig.expr0.1 = subset(overlap0.1, overlap0.1$pvalue < 0.05 & overlap0.1$Stouffer > 0.05)

overlap.sig.meth0.1 = subset(overlap0.1, overlap0.1$Stouffer < 0.05 & overlap0.1$pvalue > 0.05)

overlap.sig.expr.meth0.1 = subset(overlap0.1, overlap0.1$pvalue < 0.05 & overlap0.1$Stouffer < 0.05)

##cate 0.5

overlap0.5 = right\_join(ddsresults.df, DMRcate\_05\_unique.hg19[,c("Gene\_Name","Stouffer")], by = c("Gene\_Name" = "Gene\_Name"))

overlap0.5 = overlap0.5[complete.cases(overlap0.5$pvalue), ]

overlap.unsig0.5 = subset(overlap0.5, overlap0.5$pvalue > 0.05 & overlap0.5$Stouffer > 0.05)

overlap.sig.expr0.5 = subset(overlap0.5, overlap0.5$pvalue < 0.05 & overlap0.5$Stouffer > 0.05)

overlap.sig.meth0.5 = subset(overlap0.5, overlap0.5$Stouffer < 0.05 & overlap0.5$pvalue > 0.05)

overlap.sig.expr.meth0.5 = subset(overlap0.5, overlap0.5$pvalue < 0.05 & overlap0.5$Stouffer < 0.05)

## Plots made with rightjoin

plot(x = -log10(overlap.unsig$Stouffer), y = -log10(overlap.unsig$pvalue), cex = 0.8, pch = 21, bg = "#F8766D", xlab = "-log10 Stouffer, DMRCate", ylab = "-log10 pVal, RNA Data", xlim = c(-0.1,10), ylim = c(-0.1, 10))

points(x = -log10(overlap.sig.expr$Stouffer), y = -log10(overlap.sig.expr$pvalue), cex = 0.8, pch = 21, bg = "#00BFC4", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlap.sig.meth$Stouffer), y = -log10(overlap.sig.meth$pvalue), cex = 0.8, pch = 21, bg = "#7CAE00", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlap.sig.expr.meth$Stouffer), y = -log10(overlap.sig.expr.meth$pvalue), cex = 0.8, pch = 21, bg = "#C77CFF", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

write.table(overlap.unsig,"~/L-personal/Down-iHD/all/Quadrant\_unsig\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlap.sig.expr,"~/L-personal/Down-iHD/all/Quadrant\_sig.expr\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlap.sig.meth,"~/L-personal/Down-iHD/all/Quadrant\_sig.meth\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlap.sig.expr.meth,"~/L-personal/Down-iHD/all/Quadrant\_sig.expr.meth\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

##cate 0.1

plot(x = -log10(overlap.unsig0.1$Stouffer), y = -log10(overlap.unsig0.1$pvalue), cex = 0.8, pch = 21, bg = "#F8766D", xlab = "-log10 Stouffer, DMRCate", ylab = "-log10 pVal, RNA Data", xlim = c(-0.1,10), ylim = c(-0.1, 10))

points(x = -log10(overlap.sig.expr0.1$Stouffer), y = -log10(overlap.sig.expr0.1$pvalue), cex = 0.8, pch = 21, bg = "#00BFC4", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlap.sig.meth0.1$Stouffer), y = -log10(overlap.sig.meth0.1$pvalue), cex = 0.8, pch = 21, bg = "#7CAE00", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlap.sig.expr.meth0.1$Stouffer), y = -log10(overlap.sig.expr.meth0.1$pvalue), cex = 0.8, pch = 21, bg = "#C77CFF", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

write.table(overlap.unsig0.1,"~/L-personal/Down-iHD/all/Quadrant\_unsig\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlap.sig.expr0.1,"~/L-personal/Down-iHD/all/Quadrant\_sig.expr\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlap.sig.meth0.1,"~/L-personal/Down-iHD/all/Quadrant\_sig.meth\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlap.sig.expr.meth0.1,"~/L-personal/Down-iHD/all/Quadrant\_sig.expr.meth\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

##cate 0.5

plot(x = -log10(overlap.unsig0.5$Stouffer), y = -log10(overlap.unsig0.5$pvalue), cex = 0.8, pch = 21, bg = "#F8766D", xlab = "-log10 Stouffer, DMRCate", ylab = "-log10 pVal, RNA Data", xlim = c(-0.1,10), ylim = c(-0.1, 10))

points(x = -log10(overlap.sig.expr0.5$Stouffer), y = -log10(overlap.sig.expr0.5$pvalue), cex = 0.8, pch = 21, bg = "#00BFC4", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlap.sig.meth0.5$Stouffer), y = -log10(overlap.sig.meth0.5$pvalue), cex = 0.8, pch = 21, bg = "#7CAE00", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlap.sig.expr.meth0.5$Stouffer), y = -log10(overlap.sig.expr.meth0.5$pvalue), cex = 0.8, pch = 21, bg = "#C77CFF", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

write.table(overlap.unsig0.5,"~/L-personal/Down-iHD/all/Quadrant\_unsig\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlap.sig.expr0.5,"~/L-personal/Down-iHD/all/Quadrant\_sig.expr\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlap.sig.meth0.5,"~/L-personal/Down-iHD/all/Quadrant\_sig.meth\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlap.sig.expr.meth0.5,"~/L-personal/Down-iHD/all/Quadrant\_sig.expr.meth\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

##overlap findOverlaps script, links by genomic ranges

overlaps.fo = as.data.frame(findOverlaps(dmrcate\_unique.hg19.gr,ddsresults.gr, maxgap = 2000))

idx <- overlaps.fo$subjectHits

idx2 = overlaps.fo$queryHits

tmp <- as.data.frame(ddsresults.gr$pvalue[idx], ddsresults.gr$ENS\_ID[idx])

tmp2 <- as.data.frame(cbind(dmrcate\_unique.hg19.gr$Gene\_Name[idx2], dmrcate\_unique.hg19.gr$Stouffer[idx2]),ddsresults.gr$Gene\_Name[idx2])

overlaps.fo = data.frame(Gene\_Name = rownames(tmp),pvalue = tmp[,1], Stouffer = as.character(tmp2[,2]), stringsAsFactors = F)

overlaps.fo$pvalue=as.numeric(overlaps.fo$pvalue)

overlaps.fo$Stouffer=as.numeric(overlaps.fo$Stouffer)

overlaps.fo = unique(overlaps.fo)

remove("idx","idx2","tmp","tmp2")

##cate 0.1

overlaps.fo\_01 = as.data.frame(findOverlaps(dmrcate\_01\_unique.hg19.gr,ddsresults.gr, maxgap = 2000))

idx <- overlaps.fo\_01$subjectHits

idx2 = overlaps.fo\_01$queryHits

tmp <- as.data.frame(ddsresults.gr$pvalue[idx], ddsresults.gr$ENS\_ID[idx])

tmp2 <- as.data.frame(cbind(dmrcate\_01\_unique.hg19.gr$Gene\_Name[idx2], dmrcate\_01\_unique.hg19.gr$Stouffer[idx2]),ddsresults.gr$Gene\_Name[idx2])

overlaps.fo\_01 = data.frame(Gene\_Name = rownames(tmp),pvalue = tmp[,1], Stouffer = as.character(tmp2[,2]), stringsAsFactors = F)

overlaps.fo\_01$pvalue=as.numeric(overlaps.fo\_01$pvalue)

overlaps.fo\_01$Stouffer=as.numeric(overlaps.fo\_01$Stouffer)

overlaps.fo\_01 = unique(overlaps.fo\_01)

remove("idx","idx2","tmp","tmp2")

##cate 0.5

overlaps.fo\_05 = as.data.frame(findOverlaps(dmrcate\_05\_unique.hg19.gr,ddsresults.gr, maxgap = 2000))

idx <- overlaps.fo\_05$subjectHits

idx2 = overlaps.fo\_05$queryHits

tmp <- as.data.frame(ddsresults.gr$pvalue[idx], ddsresults.gr$ENS\_ID[idx],)

tmp2 <- as.data.frame(cbind(dmrcate\_05\_unique.hg19.gr$Gene\_Name[idx2], dmrcate\_05\_unique.hg19.gr$Stouffer[idx2]),ddsresults.gr$Gene\_Name[idx2],)

overlaps.fo\_05 = data.frame(Gene\_Name = rownames(tmp),pvalue = tmp[,1], Stouffer = as.character(tmp2[,2]), stringsAsFactors = F)

overlaps.fo\_05$pvalue=as.numeric(overlaps.fo\_05$pvalue)

overlaps.fo\_05$Stouffer=as.numeric(overlaps.fo\_05$Stouffer)

overlaps.fo\_05 = unique(overlaps.fo\_05)

remove("idx","idx2","tmp","tmp2")

## Plots made with findoverlap

overlaps.fo.unsig = subset(overlaps.fo, overlaps.fo$pvalue > 0.05 & overlaps.fo$Stouffer > 0.05)

overlaps.fo.sig.expr = subset(overlaps.fo, overlaps.fo$pvalue < 0.05 & overlaps.fo$Stouffer > 0.05)

overlaps.fo.sig.meth = subset(overlaps.fo, overlaps.fo$Stouffer < 0.05 & overlaps.fo$pvalue > 0.05)

overlaps.fo.sig.expr.meth = subset(overlaps.fo, overlaps.fo$pvalue < 0.05 & overlaps.fo$Stouffer < 0.05)

plot(x = -log10(overlaps.fo.unsig$Stouffer), y = -log10(overlaps.fo.unsig$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#F8766D", xlab = "-log10 Stouffer, DMRCate", ylab = "-log10 pVal, RNA Data", #Depending on te above, enter p/q-val

 xlim = c(-0.1,10), ylim = c(-0.1, 10))

points(x = -log10(overlaps.fo.sig.expr$Stouffer), y = -log10(overlaps.fo.sig.expr$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#00BFC4", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlaps.fo.sig.meth$Stouffer), y = -log10(overlaps.fo.sig.meth$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#7CAE00", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlaps.fo.sig.expr.meth$Stouffer), y = -log10(overlaps.fo.sig.expr.meth$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#C77CFF", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

write.table(overlaps.fo.unsig,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_unsig\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlaps.fo.sig.expr,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_sig.expr\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlaps.fo.sig.meth,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_sig.meth\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlaps.fo.sig.expr.meth,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_sig.expr.meth\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

##cate 0.1

overlaps.fo\_01.unsig = subset(overlaps.fo\_01, overlaps.fo\_01$pvalue > 0.05 & overlaps.fo\_01$Stouffer > 0.05)

overlaps.fo\_01.sig.expr = subset(overlaps.fo\_01, overlaps.fo\_01$pvalue < 0.05 & overlaps.fo\_01$Stouffer > 0.05)

overlaps.fo\_01.sig.meth = subset(overlaps.fo\_01, overlaps.fo\_01$Stouffer < 0.05 & overlaps.fo\_01$pvalue > 0.05)

overlaps.fo\_01.sig.expr.meth = subset(overlaps.fo\_01, overlaps.fo\_01$pvalue < 0.05 & overlaps.fo\_01$Stouffer < 0.05)

plot(x = -log10(overlaps.fo\_01.unsig$Stouffer), y = -log10(overlaps.fo\_01.unsig$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#F8766D", xlab = "-log10 Stouffer, DMRCate", ylab = "-log10 pVal, RNA Data", #Depending on te above, enter p/q-val

 xlim = c(-0.1,10), ylim = c(-0.1, 10))

points(x = -log10(overlaps.fo\_01.sig.expr$Stouffer), y = -log10(overlaps.fo\_01.sig.expr$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#00BFC4", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlaps.fo\_01.sig.meth$Stouffer), y = -log10(overlaps.fo\_01.sig.meth$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#7CAE00", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlaps.fo\_01.sig.expr.meth$Stouffer), y = -log10(overlaps.fo\_01.sig.expr.meth$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#C77CFF", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

write.table(overlaps.fo\_01.unsig,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_unsig\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlaps.fo\_01.sig.expr,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_sig.expr\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlaps.fo\_01.sig.meth,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_sig.meth\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlaps.fo\_01.sig.expr.meth,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_sig.expr.meth\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

##cate 0.5

overlaps.fo\_05.unsig = subset(overlaps.fo\_05, overlaps.fo\_05$pvalue > 0.05 & overlaps.fo\_05$Stouffer > 0.05)

overlaps.fo\_05.sig.expr = subset(overlaps.fo\_05, overlaps.fo\_05$pvalue < 0.05 & overlaps.fo\_05$Stouffer > 0.05)

overlaps.fo\_05.sig.meth = subset(overlaps.fo\_05, overlaps.fo\_05$Stouffer < 0.05 & overlaps.fo\_05$pvalue > 0.05)

overlaps.fo\_05.sig.expr.meth = subset(overlaps.fo\_05, overlaps.fo\_05$pvalue < 0.05 & overlaps.fo\_05$Stouffer < 0.05)

plot(x = -log10(overlaps.fo\_05.unsig$Stouffer), y = -log10(overlaps.fo\_05.unsig$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#F8766D", xlab = "-log10 Stouffer, DMRCate", ylab = "-log10 pVal, RNA Data", #Depending on te above, enter p/q-val

 xlim = c(-0.1,10), ylim = c(-0.1, 10))

points(x = -log10(overlaps.fo\_05.sig.expr$Stouffer), y = -log10(overlaps.fo\_05.sig.expr$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#00BFC4", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlaps.fo\_05.sig.meth$Stouffer), y = -log10(overlaps.fo\_05.sig.meth$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#7CAE00", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

points(x = -log10(overlaps.fo\_05.sig.expr.meth$Stouffer), y = -log10(overlaps.fo\_05.sig.expr.meth$pvalue), #Specify q-value or P.Value

 cex = 0.8, pch = 21, bg = "#C77CFF", xlab = "", ylab = "", xlim = c(-0.1,10), ylim = c(-0.1, 10)) #Depending on te above, enter p/q-val

write.table(overlaps.fo\_05.unsig,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_unsig\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlaps.fo\_05.sig.expr,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_sig.expr\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlaps.fo\_05.sig.meth,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_sig.meth\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

write.table(overlaps.fo\_05.sig.expr.meth,"~/L-personal/Down-iHD/all/Quadrant\_findoverlap\_sig.expr.meth\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

#Pathway analysis overlapping data according to the rightjoin method as only here we have a stat column.

stats\_overlap = unique(overlap$stat)

names(stats\_overlap) = unique(overlap$Gene\_Name)

stats\_overlap = stats\_overlap[order(-stats\_overlap)]

fgsea\_overlap <- fgsea(pathways=hallmark, stats=stats\_overlap, nperm=1000000)

fgsea\_overlap <- fgsea\_overlap[order(fgsea\_overlap$pval),]

fgsea\_overlap$leadingEdge = gsub('c\\(',"",fgsea\_overlap$leadingEdge)

fgsea\_overlap$leadingEdge = gsub('\\)',"",fgsea\_overlap$leadingEdge)

fgsea\_overlap$leadingEdge = gsub('\\"',"",fgsea\_overlap$leadingEdge)

colnames(fgsea\_overlap)[8] = "Gene\_Name"

write.table(fgsea\_overlap,"~/L-personal/Down-iHD/all/Pathways\_Overlap\_RNA-Cate\_fdr0.05.txt", sep="\t",col.names=T,row.names=F) # write data to file

##cate 0.1

stats\_overlap\_01 = unique(overlap0.1$stat)

names(stats\_overlap\_01) = unique(overlap0.1$Gene\_Name)

stats\_overlap\_01 = stats\_overlap\_01[order(-stats\_overlap\_01)]

fgsea\_overlap\_01 <- fgsea(pathways=hallmark, stats=stats\_overlap\_01, nperm=1000000)

fgsea\_overlap\_01 <- fgsea\_overlap\_01[order(fgsea\_overlap\_01$pval),]

fgsea\_overlap\_01$leadingEdge = gsub('c\\(',"",fgsea\_overlap\_01$leadingEdge)

fgsea\_overlap\_01$leadingEdge = gsub('\\)',"",fgsea\_overlap\_01$leadingEdge)

fgsea\_overlap\_01$leadingEdge = gsub('\\"',"",fgsea\_overlap\_01$leadingEdge)

colnames(fgsea\_overlap\_01)[8] = "Gene\_Name"

write.table(fgsea\_overlap\_01,"~/L-personal/Down-iHD/all/Pathways\_Overlap\_RNA-Cate\_fdr0.1.txt", sep="\t",col.names=T,row.names=F) # write data to file

##cate 0.5

stats\_overlap\_05 = unique(overlap0.5$stat)

names(stats\_overlap\_05) = unique(overlap0.5$Gene\_Name)

stats\_overlap\_05 = stats\_overlap\_05[order(-stats\_overlap\_05)]

fgsea\_overlap\_05 <- fgsea(pathways=hallmark, stats=stats\_overlap\_05, nperm=1000000)

fgsea\_overlap\_05 <- fgsea\_overlap\_05[order(fgsea\_overlap\_05$pval),]

fgsea\_overlap\_05$leadingEdge = gsub('c\\(',"",fgsea\_overlap\_05$leadingEdge)

fgsea\_overlap\_05$leadingEdge = gsub('\\)',"",fgsea\_overlap\_05$leadingEdge)

fgsea\_overlap\_05$leadingEdge = gsub('\\"',"",fgsea\_overlap\_05$leadingEdge)

colnames(fgsea\_overlap\_05)[8] = "Gene\_Name"

write.table(fgsea\_overlap\_05,"~/L-personal/Down-iHD/all/Pathways\_Overlap2\_RNA-Cate\_fdr0.5.txt", sep="\t",col.names=T,row.names=F) # write data to file

##Manhatten chromosome plots for lmfit dataset

lmfit.plot = as.data.frame.matrix(lmfit)

lmfit.plot$chr = sapply(lmfit.plot$chr, gsub, pattern="chr",replacement="")

lmfit.plot$chr = as.numeric(lmfit.plot$chr)

lmfit.plot$pos = as.numeric(lmfit.plot$pos)

lmfit.plot = lmfit.plot[order(lmfit.plot$chr,lmfit.plot$pos),]

for (chr in 1:22) { #cor.fit.pos21 = cor.fit.pos.plot[which(cor.fit.pos.plot$chr == "21"), ] #Manhattan for one chromosome

 jpg = paste("~/L-personal/Down-iHD/all/Manhattan-chr\_", chr, ".jpg", sep="")

 jpeg(jpg)

 cols = c()

 .col = c("blue", "black")

 coor = c()

 p = c()

 q = c()

 newRes = lmfit.plot[which(lmfit.plot$chr == chr), ]

 q = as.numeric(newRes$adj.P.Val)

 p = as.numeric(newRes$P.Value)

 coor = (as.numeric(newRes$pos))

 xmin = min(coor[!is.na(coor)])

 xmax = max(coor[!is.na(coor)])

 cols = c(rep(.col[chr %% 1 + 1], nrow(newRes)))

 plot(coor, -log10(q), type = "p", col = cols, pch = 20, xlab=c("chr ",chr), ylab = "-log10(q)", ylim = c(0, 4), xlim=c(xmin, xmax), xaxt = "n")

 axis(1, at=c(xmin,xmax), pos=-0.5)

 lines(c(-1E10, 1E10), c(-log10(0.05), -log10(0.05)), lty = 3, col = "red")

 dev.off()

}

remove("xmin","xmax","q","p","coor","cols","jpg",".col")

##Chromosome EPIC hypo/hyper beta plot

library(RColorBrewer)

library(ggplot2)

hypo\_chromosome.cpgs = rownames(beta.df[which(beta.df$Delta < 0), ]) #All hypo probes

hypo\_chromosome = table(as.character(seqnames(featu.gr[hypo\_chromosome.cpgs,])))

hyper\_chromosome.cpgs = rownames(beta.df[which(beta.df$Delta > 0),]) #All hyper probes

hyper\_chromosome = table(as.character(seqnames(featu.gr[hyper\_chromosome.cpgs,])))

hypo\_chromosome.perc = hypo\_chromosome/(hypo\_chromosome + hyper\_chromosome)\*100

hyper\_chromosome.perc = hyper\_chromosome/(hypo\_chromosome + hyper\_chromosome)\*100

diff\_chromosome.df = data.frame(Hypo = hypo\_chromosome.perc, Hyper = hyper\_chromosome.perc)[,-3]

colnames(diff\_chromosome.df) = c("Chromosome", "Hypo", "Hyper")

diff\_chromosome.melt = reshape2::melt(diff\_chromosome.df)

colnames(diff\_chromosome.melt) = c("Chromosome", "Direction", "Percentage")

chromosomes.vec = unique(beta.df$chr)

chromosomes.fac = c()

for(i in 1:length(chromosomes.vec)){

 chromosomes.fac[i] = paste0("chr", i)}

diff\_chromosome.melt$Chromosome = factor(diff\_chromosome.melt$Chromosome, levels = chromosomes.fac)

jpeg("~/L-personal/Down-iHD/all/hypo\_hyperplot.jpg",width=1000,height=500)

ggplot(diff\_chromosome.melt, aes(x = Chromosome, y = Percentage, ymax = 1.3), breaks = NULL) +

 stat\_summary(aes(label = round(..y.., 2)), fun = sum, geom = "text", size = 4, color = "white") +

 geom\_col(aes(x = Chromosome, y = Percentage, fill = Direction), position = position\_dodge()) +

 geom\_text(aes(label = round(Percentage, 2), group = Direction), vjust = 0.4, hjust = 1.14, position = position\_dodge(0.9), size = 4, angle = 90) +

 theme\_bw() + ylab("Percentage relative to total probes")+ scale\_fill\_brewer(direction = 1, aesthetics = "fill",palette = "Greys") + labs(title = "Uncorrected probe betas (-XY, -SNPs)") +

 theme(axis.text = element\_text(size = 12), axis.title = element\_text(size = 14), axis.text.x = element\_text(angle = 45, hjust = 1), axis.title.x = element\_blank())

dev.off()

##Chromosome EPIC hypo/hyper beta plot, specified with p- or q-value, as desired

hypo\_chromosome.cpgs.sig = rownames(beta.df[which(beta.df$Delta < 0 & beta.df$q.value < 0.5),]) #All hypo probes, significant

hypo\_chromosome.sig = table(as.character(seqnames(featu.gr[hypo\_chromosome.cpgs.sig,])))

hyper\_chromosome.cpgs.sig = rownames(beta.df[which(beta.df$Delta > 0 & beta.df$q.value < 0.5),]) #All hyper probes, significant

hyper\_chromosome.sig = table(as.character(seqnames(featu.gr[hyper\_chromosome.cpgs.sig,])))

hypo\_chromosome.perc.sig = hypo\_chromosome.sig/(hypo\_chromosome + hyper\_chromosome)\*100 #Compared to all cpg

hyper\_chromosome.perc.sig = hyper\_chromosome.sig/(hypo\_chromosome + hyper\_chromosome)\*100 #Compared to all cpg

diff\_chromosome.df.sig = data.frame(Hypo = hypo\_chromosome.perc.sig, Hyper = hyper\_chromosome.perc.sig)[,-3]

colnames(diff\_chromosome.df.sig) = c("Chromosome", "Hypo", "Hyper")

diff\_chromosome.melt.sig = reshape2::melt(diff\_chromosome.df.sig)

colnames(diff\_chromosome.melt.sig) = c("Chromosome", "Direction", "Percentage")

diff\_chromosome.melt.sig$Chromosome = factor(diff\_chromosome.melt.sig$Chromosome, levels = chromosomes.fac)

jpeg("~/L-personal/Down-iHD/all/Significant\_hypo\_hyperplot.jpg",width=1000,height=500)

ggplot(diff\_chromosome.melt.sig, aes(x = Chromosome, y = Percentage, ymax = 1.3), breaks = NULL) +

 geom\_col(aes(x = Chromosome, y = Percentage)) +

 stat\_summary(aes(label = round(..y.., 2)), fun = sum, geom = "text", size = 4, color = "white", vjust = 1.4) +

 geom\_col(aes(x = Chromosome, y = Percentage, fill = Direction), position = position\_dodge()) +

 geom\_text(aes(label = round(Percentage, 2), group = Direction), vjust = 0.4, hjust = 1.14, position = position\_dodge(0.9), size = 4, angle = 90) +

 theme\_bw() + ylab("Percentage relative to total probes")+ scale\_fill\_brewer(direction = 1, aesthetics = "fill",palette = "Greys") + labs(title = "Uncorrected probe betas (-XY, -SNPs) with q.value < 0.5") +

 theme(axis.text = element\_text(size = 12), axis.title = element\_text(size = 14), axis.text.x = element\_text(angle = 45, hjust = 1), axis.title.x = element\_blank())

dev.off()

##Chromosome RNA under/overexpression readcounts plot

hypo\_chromosome.dds.rna = rownames(ddsresults.df[which(ddsresults.df$log2FoldChange < 0 & !ddsresults.df$chr %in% c("chrX","chrY")), ]) # Alle hypo

hypo\_chromosome.rna = table(as.character(seqnames(ddsresults.gr[hypo\_chromosome.dds.rna,])))

hyper\_chromosome.dds.rna = rownames(ddsresults.df[which(ddsresults.df$log2FoldChange > 0 & !ddsresults.df$chr %in% c("chrX","chrY")), ]) # Alle hyper

hyper\_chromosome.rna = table(as.character(seqnames(ddsresults.gr[hyper\_chromosome.dds.rna,])))

hypo\_chromosome.perc.rna = hypo\_chromosome.rna/(hypo\_chromosome.rna + hyper\_chromosome.rna)\*100

hyper\_chromosome.perc.rna = hyper\_chromosome.rna/(hypo\_chromosome.rna + hyper\_chromosome.rna)\*100

diff\_chromosome.df.rna = data.frame(Hypo = hypo\_chromosome.perc.rna, Hyper = hyper\_chromosome.perc.rna)[,-3]

colnames(diff\_chromosome.df.rna) = c("Chromosome", "Under-expressed", "Over-expressed")

diff\_chromosome.melt.rna = reshape2::melt(diff\_chromosome.df.rna)

colnames(diff\_chromosome.melt.rna) = c("Chromosome", "Direction", "Percentage")

diff\_chromosome.melt.rna$Chromosome = factor(diff\_chromosome.melt.rna$Chromosome, levels = chromosomes.fac)

jpeg("~/L-personal/Down-iHD/all/RNA\_hypo\_hyperplot.jpg",width=1000,height=500)

ggplot(diff\_chromosome.melt.rna, aes(x = Chromosome, y = Percentage, ymax = 1.3), breaks = NULL) +

 stat\_summary(aes(label = round(..y.., 2)), fun = sum, geom = "text", size = 4, color = "white") +

 geom\_col(aes(x = Chromosome, y = Percentage, fill = Direction), position = position\_dodge()) +

 geom\_text(aes(label = round(Percentage, 2), group = Direction), vjust = 0.4, hjust = 1.14, position = position\_dodge(0.9), size = 4, angle = 90) +

 theme\_bw() + ylab("Percentage relative to total probes")+ scale\_fill\_brewer(direction = 1, aesthetics = "fill",palette = "Greys") + labs(title = "RNA counts data (-XY)") +

 theme(axis.text = element\_text(size = 12), axis.title = element\_text(size = 14), axis.text.x = element\_text(angle = 45, hjust = 1), axis.title.x = element\_blank())

dev.off()

##Chromosome RNA under/overexpression plot, specified with p- or q-value, as desired

hypo\_chromosome.dds.rna.sig = rownames(ddsresults.df[which(ddsresults.df$log2FoldChange < 0 & ddsresults.df$padj < 0.5 & !ddsresults.df$chr %in% c("chrX","chrY")),]) #All hypo probes, significant

hypo\_chromosome.rna.sig = table(as.character(seqnames(ddsresults.gr[hypo\_chromosome.dds.rna.sig,])))

hyper\_chromosome.dds.rna.sig = rownames(ddsresults.df[which(ddsresults.gr$log2FoldChange > 0 & ddsresults.df$padj < 0.5 & !ddsresults.df$chr %in% c("chrX","chrY")),]) #All hyper probes, significant

hyper\_chromosome.rna.sig = table(as.character(seqnames(ddsresults.gr[hyper\_chromosome.dds.rna.sig,])))

hypo\_chromosome.perc.rna.sig = hypo\_chromosome.rna.sig/(hypo\_chromosome.rna + hyper\_chromosome.rna)\*100 #Compared to all cpg

hyper\_chromosome.perc.rna.sig = hyper\_chromosome.rna.sig/(hypo\_chromosome.rna + hyper\_chromosome.rna)\*100 #Compared to all cpg

diff\_chromosome.df.rna.sig = data.frame(Hypo = hypo\_chromosome.perc.rna.sig, Hyper = hyper\_chromosome.perc.rna.sig)[,-3]

colnames(diff\_chromosome.df.rna.sig) = c("Chromosome", "Under-expressed", "Over-expressed")

diff\_chromosome.melt.rna.sig = reshape2::melt(diff\_chromosome.df.rna.sig)

colnames(diff\_chromosome.melt.rna.sig) = c("Chromosome", "Direction", "Percentage")

diff\_chromosome.melt.rna.sig$Chromosome = factor(diff\_chromosome.melt.rna.sig$Chromosome, levels = chromosomes.fac)

jpeg("~/L-personal/Down-iHD/all/Significant\_RNA\_hypo\_hyperplot.jpg",width=1000,height=500)

ggplot(diff\_chromosome.melt.rna.sig, aes(x = Chromosome, y = Percentage, ymax = 1.3), breaks = NULL) +

 geom\_col(aes(x = Chromosome, y = Percentage)) +

 stat\_summary(aes(label = round(..y.., 2)), fun = sum, geom = "text", size = 4, color = "white", vjust = 1.4) +

 geom\_col(aes(x = Chromosome, y = Percentage, fill = Direction), position = position\_dodge()) +

 geom\_text(aes(label = round(Percentage, 2), group = Direction), vjust = 0.4, hjust = 1.14, position = position\_dodge(0.9), size = 4, angle = 90) +

 theme\_bw() + ylab("Percentage relative to total probes")+ scale\_fill\_brewer(direction = 1, aesthetics = "fill",palette = "Greys") + labs(title = "RNA counts data (-XY) with q.value < 0.5") +

 theme(axis.text = element\_text(size = 12), axis.title = element\_text(size = 14), axis.text.x = element\_text(angle = 45, hjust = 1), axis.title.x = element\_blank())

dev.off()

#########################################################################