Microenvironment-dependent phenotypes of human macrophages

Dissertation zur Erlangung des
Doktorgrades der Naturwissenschaften (Dr. rer. nat)
der Fakultät für Biologie und vorklinische Medizin
der Universität Regensburg

vorgelegt von
Claudia Kiesewetter

aus
Neustadt bei Coburg

im Jahr
2019

Promotionsgesuch wurde eingereicht am:
01.03.2019

Die Arbeit wurde angeleitet von:
Prof. Dr. Michael Rehli

Unterschrift:

. . . .
(Claudia Kiesewetter)

Für Matthias

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Macrophages, Professional Phagocytes

Over 100 years ago Elie Metchnikoff first observed large vacuolar cells, which efficiently phagocytosed foreign particles. He named them macrophages, derived from the greek word for "big eater" [4]. These cells are part of the innate immune system, which provides a first line of defence against infections. The innate system consists of many different cell types, physical barriers and soluble proteins, which continually try to prevent an infection [5].

Innate immune cells derive from haematopoietic stem cells in the bone marrow, which give rise to the lymphoid, erythroid and myeloid lineages. The latter gives rise to mature monocytes amongst others, which exit the bone marrow in order to patrol the blood stream. Afterwards they can migrate into tissues where they are capable of differentiating into either dendritic cells or macrophages depending on the cytokine milieu inside the tissue [6].

For a long time the differentiation of monocytes, derived form adult haematopoiesis, was the prevailing view of macrophage origin, despite some early reports stating that tissue macrophage populations are independent of blood monocytes [7, 8]. Technical advances have made it possible to trace the embryonic seeding of macrophages in mice and track the population during embryogenesis and beyond. These fate-mapping analyses have shown that resident tissues are seeded by yolk sac and foetal liver progenitors during early development and that the derived macrophage populations persist unto adulthood by slow proliferation [9, 10, 11, 12, 13, 14, 15, 4]. Microglia, the resident macrophage population of the central nervous system, are exclusively derived from yolk sac c-kit^{+} erythromyeloid progenitor cells [16], whereas all other tissue macrophage populations are also seeded by foetal liver monocytes [4]. An exception to this rule are intestinal macrophages, as these are replenished by peripheral blood monocytes [4].

Using a neo-natal empty alveolar niche in mice, it was shown that yolk sac progenitors, foetal monocytes and bone marrow derived monocytes are capable of seeding the niche and differentiating into functional tissue resident macrophages. Foetal monocytes seemed to have a higher proliferative capacity and therefore a competitive advantage over the other possible progenitors. On a molecular level all three cell types showed a specific transcriptome, but after transplantation the cells differentiated into highly similar alveolar macrophages, which are able to perform all needed functions [17]. Similarly, analysing the transcriptome of alveolar macrophages from chimeric mice, which were set up to contain both adult and embryonic derived cells, both gave rise to highly similar alveolar macrophages [18]. The strong impact of the environment on macrophage phenotypes has also been shown for liver Kupffer cells. Peripheral monocytes efficiently migrated into the empty tissue and differentiated into functional liver macrophages [19]. According to these studies, we can infer that the cellular origin seems to have a small impact on the phenotype of tissue macrophages. Instead, they suggest that the local environment dictates macrophage phenotype and function.

Resident macrophages are sentinels of the immune system positioned at strategic points to guard against invading pathogens. To fulfil their role, they express many surface receptors, which enable them to sense minute changes in their environment. These encompass amongst others immunoglobulin receptors, which recognize antigen-antibody complexes, and toll like receptors (TLR). This evolutionary conserved protein family recognizes molecular patterns produced by microbes. Binding of their ligands leads to the activation of several signalling pathways and pro-inflammatory activation of the cell [5].

In addition to immune defence, resident macrophages perform homoeostatic functions which depend on the respective tissue, as does their morphology. Some resident tissue macrophage populations were named after their discoverer, for example the Langerhans cells, which reside in the skin [20]. There they are responsible for antigen presentation [21], whereas intestinal macrophages phagocytose microbiota without inducing an inflammatory response [22]. Also in the lung, alveolar macrophages remove inhaled debris, while maintaining a low inflammatory context [23]. In contrast, microglia residing in the central nervous system (CNS), are important for the refinement of synaptic networks and maintaining homoeostasis [24]. Appropriately, they have a unique morphology ranging from ramified, pseudopodic to amoeboid depending on their location within the tissue [25]. These are just examples to show how each resident population has their own name, morphology and function [21].

1.1.1 Regulation of Tissue Resident Macrophage Phenotypes

In order to analyse different macrophage populations more thoroughly, Lavin and colleagues profiled seven mouse tissue macrophage populations: microglia, Kupffer cells red pulp-, alveolar-, peritoneal cavity- and intestinal-macrophages. They showed that each population has an unique transcriptome and open chromatin features. On top, they found transcription factors specifically expressed in different populations as Gata6 in peritoneal macrophages, Sall1 in microglia and Spi-C in spleen macrophages [26, 27]. The importance of Sall1 for microglia has been backed up by another report [28]. Here it was shown that microglia are dependent on CSF1R and TGF-β signalling to maintain their specific phenotype. Upon deletion of the Sall1 locus in mice, microglia turned from homoeostatic to inflammatory macrophages [28].

Gata6, the essential factor for peritoneal macrophages, has been studied extensively. It coordinates the steady state proliferation and inflammatory response of the cells [29]. It is reversibly induced by retinoic acid, which is a derivative of vitamin A. Retinoic acid converting enzymes are abundantly expressed in the peritoneal cavity by mesothelial cells, which means that only macrophages present in the peritoneal cavity receive this signal. Gata6 deletion in mice leads to drastic changes in their transcriptome, which in turn induces a longer inflammatory response by these cells when stimulated by bacteria [29]. Furthermore, Gata6 deficient peritoneal macrophages were not able to induce IgA producing B-1 cells, which are important for mucosal immunity [30]. A recent study showed that human peritoneal macrophages also express Gata6 in contrast to peripheral blood monocytes, indicating that similar regulatory mechanisms could control their phenotype [31].

Another tissue macrophage population that has been extensively studied in mice, are alve-

olar macrophages. Signalling via GM-CSF is essential for the correct development of these macrophages [14]. Both GM-CSF and TGF-β signalling induce expression of PPAR-γ, an essential transcription factor of the cells [32, 33].
Additionally, TGF-β signalling is important for monocyte differentiation in the intestine [34]. Intestinal macrophages are vital for maintaining gut homoeostasis by disposing of microbes without inducing an inflammatory response [34, 22]. Analysis of the transcriptome from mature intestinal macrophages, in comparison to recently arrived monocytes, shows an up-regulation of genes involved with phagocytosis (Mertk, Gas6, Itgb5, Cd9, C1qa-c), monocyte chemoattractants, components of the complement cascade and members of the TGFβR signalling pathway. It has also been shown that the last element is essential to attract the correct amount of monocytes to the intestine and control their differentiation [34]. These regulatory mechanisms have been uncovered using murine macrophages, whether the same pattern is active in human tissue macrophages has to be established.

Hofbauer Cells, Tissue Macrophages of the Placenta

Another fascinating example are Hofbauer cells, which are macrophages inside the placenta. This tissue has several functions that are essential for a successful pregnancy. It facilitates gas and nutrient exchange between mother and child, but also protects the foetus from infections [35]. The tissue is made up of treelike structures, the so called villi, which are in direct contact with circulating maternal blood. Hofbauer cells are located inside the stroma of these, close to the foetal capillaries [36].
These macrophages are of embryonic origin, deriving from several progenitors, depending on the stage of development. Before the onset of foetal circulation they derive from mesenchymal stem cells, afterwards from foetal monocytes [37, 38, 39]. Discovered over 100 years ago, the cells were described by Hofbauer and several other investigators as large, pleiomorphic and highly vacuolated cells [39]. They seem to maintain their population by slow proliferation, since mitotic cells have been observed [40].
Hofbauer cells are important for regulating vasculogenesis and angiogenesis in the developing placenta [41], since they are the major source of sprouty proteins, important regulators of branching and growth factor signalling, in the placenta [42]. Additionally it has been shown that Hofbauer cells, isolated from third trimester placenta, express and secrete pro-angiogenic factors such as vascular endothelial growth factor (VEGF) and fibroblast growth factor 2 (FGF2). In an *in vitro* model, they were also able to induce angiogenesis of isolated primary feto-placental endothelial cells (fpEC) [43].
Furthermore, they play an important role in suppressing the maternal immune system to accept the semi-allogenic foetus. To achieve this genes involved in immune response were found to be hypermethylated and therefore silenced in foetal cells (foetal monocytes and Hofbauer cells) compared to the maternal monocytes and decidual macrophages [44]. In contrast, genes belonging to immune suppressive categories were hypomethylated, thus activated in foetal cells. Moreover the expression of DNA methyltransferases DNMT1, 3A and 3B were found to be lower expressed in decidual macrophages compared to Hofbauer

cells. These results indicate an epigenetic control of Hofbauer cells to a more immunosuppressive phenotype [44]. Additionally, they express the anti-inflammatory cytokine IL-10, as well as the folate receptor β and are thus classified as immuno-suppressive macrophages [45, 46]. However, when stimulated with lipopolysaccharide (LPS) they are capable of responding with the production of pro-inflammatory interleukin-6 and -8 [45].

1.1.2 Macrophage Polarization

Historically, macrophage polarization has been viewed in the context of T cell activation. T helper cells type 1 ($\mathrm{T_H}$1) produce inflammatory cytokines for example interferon-γ, whereas type 2 T helper cells ($\mathrm{T_H}$2) produce anti-inflammatory cytokines, such as interleukin-4 [47]. Accordingly, macrophage polarization was divided into two opposing states, a pro-inflammatory (M1) and an anti-inflammatory (M2) phenotype [48]. Stimulation of resting macrophages with microbial products such as LPS leads to a swift pro-inflammatory polarization, called M1, which is characterized by the production of inflammatory mediators, such as interleukin 6 and reactive oxygen species. In contrast, when treating resting macrophages with interleukin 4 or 13 the cells are converted into an anti-inflammatory phenotype, called M2, which is characterised by the production of arginase and interleukin 10 (Reviewed in [49]). *In vitro* differentiation of monocytes to macrophages can be achieved by using one of the essential growth factors, either M-CSF or GM-CSF. Stimulation with M-CSF has been reported to result in a slightly anti-inflammatory (M2) phenotype, whereas GM-CSF induces a more pro-inflammatory phenotype in the resulting macrophages [50]. GM-CSF differentiated macrophages express higher levels of IRF4, TNF and IL-23, whereas M-CSF differentiated cells express higher levels of IL-10 [51].

This dichotomy of macrophage activation has been useful in the case of acute infections [52].Subsequent studies on macrophage activation revealed an increased complexity on the nature of their different polarization states, which ended up being integrated into the existing system of M1-M2 integrated into the existing system of M1-M2 by creating M2 subclasses, M2a (IL-4 or IL-13 treatment), M2b (immune complexes with IL-1β) and M2c (TGF-β or glucocorticoids) [53, 43]. Tumour associated macrophages could not be classified to either M1 or M2 [54]. After several transcriptome studies of macrophage activation with immune-related stimuli, the M1-M2 system slowly gave way to a spectrum model of macrophage polarization [55, 56, 57, 58]. In order to shed light into this matter, Xue et. al. looked at the reaction of macrophages to several stimuli and analysed their transcriptome. The factors used were immune related, for example IL-4, IL-13, TNF, or LPS and combinations of these. Transcriptome analysis revealed that every stimulus induced a specific phenotype and when such trancriptomic programs were applied to alveolar to alveolar macrophages from smokers and COPD patients, they could see that the latter lost inflammatory signatures. This study clearly shows that macrophages have a broad polarization potential, not limited to the classical pro- and anti-inflammatory dichotomy [59].

Taken together with the fact that tissue resident macrophages also possess a tissue specific phenotype, macrophage polarization should be viewed as a spectrum of possibilities, with each stimulus potentially inducing a distinct phenotype.

1.1.3 Transcription Factors involved in Macrophage Differentiation and Polarization

Lineage commitment and differentiation involves the so-called master transcription factors, which are capable of binding their recognition site in closed chromatin conformation. They recruit chromatin remodelers to open up the genomic locus for cell type or signal specific transcription [60, 61]. In the myeloid lineage, PU.1 has been identified as an essential master transcription factor, since the homozygous deletion of PU-1 in mice led to a complete depletion of mature macrophages, culminating in the death of newborns within a period of 48 hours [62]. PU-1 is slightly expressed in stem cells, mature erythroid cells, megakaryocytes and T cells, whereas macrophages and B cells contain high levels of PU.1 [63, 64, 65].

The human genome contains up to a million binding sites for PU-1, from which approximately 45000 are actually bound [66]. In order to select the correct cell specific binding sites, PU-1 collaboratively binds with other signal dependent transcription factors, including CCAAT-enhancer-binding proteins beta (C/EBPβ) [66]. In order to establish the microglia phenotype, PU-1 seems to work with SMAD transcription factors to activate a specific subset of enhancers [67, 24].

Furthermore PU-1 binds together with two members of the interferon regulatory factor protein family (IRF) 4 and 8, an ETS:IRF composite motif (EICE) to regulate gene expression [68, 69, 70]. Both IRF4 and IRF8 are expressed in the myeloid lineage and perform separate functions. Double knock out studies in mice led to a chronic myeloid leukemia (CML)-like disease, characterized by an increase of granulocytes and a concomitant decrease of monocytes and macrophages [71]. In addition IRF8 and PU-1 are essential for the development of microglia in the CNS, further confirming the role of IRFs during macrophage differentiation [16]. In contrast, the development of murine bone macrophages depends on the transcription factor Nuclear factor of activated T-cells (NFATC2) [72]. NFATC2 expression can be induced via c-Jun, which forms a heterodimer, sometimes called AP-1, with the transcription factor FOS. [73, 74]. Fos and Jun are known to influence bone development [75], but they are also involved in the differentiation of all haematopoietic cells [76].

In addition to PU1, macrophages express various other transcription factors, including MafB a member of the Maf bZIP transcription factor family, which is a subgroup of the AP-1 family. It can form homo- or heterodimers, with either c-Maf or FOS, which recognize a special DNA sequence, the so called MARE element [77]. MafB is successively up-regulated during myeloid lineage commitment, reaching it's highest expression in macrophages [78], which can be further increased *in vitro* by stimulation with interleukin 4 and 10, or reduced by LPS or GM-CSF [79]. Macrophages are usually non-proliferating cells, but deletion of both MafB and c-MAF in murine macrophages led to proliferation

without loss of function or phenotype [80]. Both c-MAF and MAFB belong to the subgroup of large MAF proteins, which contain a trans-activation domain and are thereby capable of activating transcription by forming a homodimer [81].
Another factor collaborating with PU-1 is EGR2, a member of the early response genes (EGR). At the intronic enhancer site of the M-CSF receptor gene (*CSF1R*), binding of both PU-1 and EGR2 is required for efficient transcription of this receptor during macrophage differentiation [82]. *In vitro* EGR2 can be induced by M-CSF treatment [83], or by adhesion to plastic [84, 85], but not when monocytes are cultured on either collagen or fibronectin [85].

1.2 Overview of the Extracellular Matrix

In vivo all tissue macrophages come into contact with the extracellular matrix (ECM), but also monocytes encounter it during extravasation from the blood stream [5].
Functionally, the ECM provides physical scaffolding for the cells, as well as important biomechanical cues for tissue morphogenesis and differentiation. All extracellular matrices are build from the same components, but the topology can differ between tissues [86]. The main component of these matrices are collagen fibres, which are composed of three polypeptide chains, see Figure 1.1. In this structure, the amino acid sequence of Prolin-Prolin-Glycin is very important for the twisting and stability of the helix [87]. This helix is further condensed into larger fibres, which have a diameter of 50-200nm. Within this strand, each collagen molecule is staggered by 67nm leading to the characteristic pattern observed by electron microscopy. Several collagen types form this structure including type I, II, III and V [88].

Though lower in quantity, fibronectin also plays an important role inside the extracellular matrix. Structurally it is composed of 2 amino acid chains that are connected by a disulfide bond (Figure 1.2). Fibronectin acts as a structural organizer, because of its ability to bind several components of the ECM. It has binding sites for both collagen and integrins [89]. Inside tissues fibronectin is incorporated into the matrix as cellular protein, but it is also present in plasma. This form is mainly produced by hepatocytes and secreted in a non-fibrillar, soluble form (plasma fibronectin).

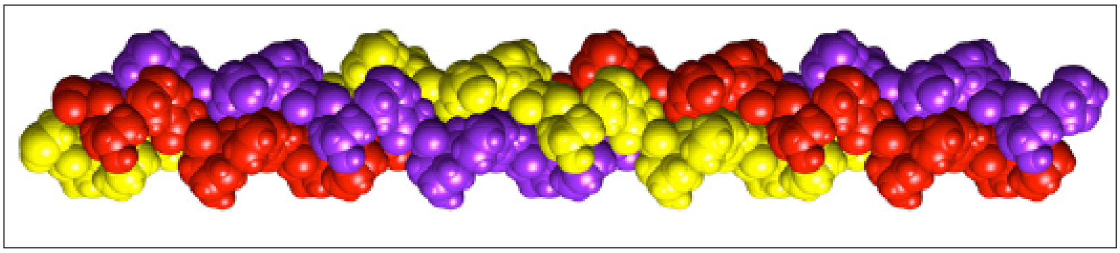

Figure 1.1: Molecular Structure of the Collagen Triple Helix, adapted from [1]

These fibrous proteins of the ECM are embedded in a hydrogel consisting of glycosaminoglycan containing proteoglycans, which are necessary for the matrix to adjust to tensile stresses [86].

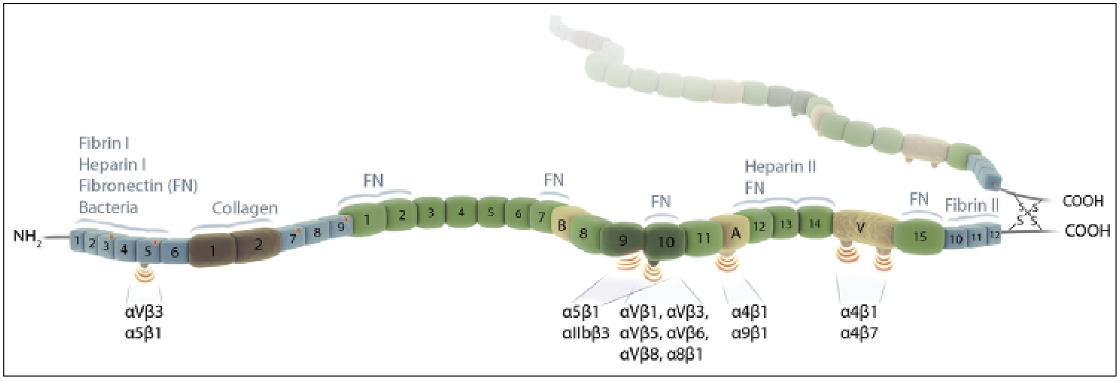

Figure 1.2: Overview of Fibronectin protein domains [2]

A specialized form of ECM is the basement membrane, which forms a thin layer around all epithelial and endothelial cells. It is normally between 50-100nm thick and consists of two connected sheets of collagen type IV and laminin networks, which results in an average pore size of 10nm [90]. Inside this layer collagen forms a two dimensional structure, that resembles honey-combs, requiring both laminin and glycoproteins to obtain a functional membrane [90].

Receptors for extracellular matrix proteins, called integrins, are expressed on the surface of leukocytes [91]. In total the protein family consists of 18 α- and 8 β-subunits, which form heterodimers with each other [92]. Fibronectin provides binding sites for 11 integrin dimers [2], whereas the protein dimers $\alpha3\beta1$, $\alpha6\beta1$, $\alpha7\beta1$, $\alpha6\beta4$ enable cells to bind to laminin fibres [93, 94]. Leukocytes are able to adhere to collagen via several integrin β_2 heterodimers [95, 96]. Adherence to collagen type I was shown to increase monocyte phagocytosis by up to 12 fold [97] and increased expression of MMP9 [98]. This protein belongs to the family of metalloproteinases, which are enzymes responsible for degrading extracellular matrix proteins such as collagen. Indeed, macrophages are not just inhabitants of the matrix, but they can also shape it. During wound repair macrophage derived MMP-10 is important to induce also MMP-13 expression and avoid skin stiffness and excess collagen deposition [99]. In contrast, macrophages are also able to build up the extracellular matrix by stimulating fibroblasts to proliferate and secrete matrix components, but also by synthesizing and secreting collagen themselves [100]. An important pathway in this process is interleukin-4 receptor α signalling, as it induces lysyl hydroxylase 2 an enzyme responsible for producing collagen cross-links [101]. Macrophages are also able to synthesize another matrix component already mentioned above, namely fibronectin [102]. Monocytes are known to tightly adhere on fibronectin coated surfaces *in vitro* [103], which leads to an increased expression of metalloproteinases two and nine, as well as their endocytosis [104]. *In vitro* generated fibronectin peptides induce a strong chemotactic response in monocytes. This is probably due to the fact that fibronectin fragments are present in sites of injury and inflammation [105]. Further evidence for this observation is that small fibronectin fragments induced a pro-inflammatory activation of macrophages during periodontitis, which was assessed by their production of TNF-α, IL-1β and IL-8 [106, 107].

1.3 Epigenetic Control of Cellular Phenotypes

In addition to the genetic information encoded by the DNA sequence, a second layer of information is used to determine cellular phenotypes. It is imprinted on the DNA sequence and called epigenetics. The term was coined in 1942 by Waddington and described as “the causal interactions between genes and their products which bring the phenotype into being” [108]. By now, several mechanisms are included in this term, which are schematically depicted in Figure 1.3. Cytosine nucleotides can be methylated by DNA methyltransferases (DNMT), usually in the context of CpG dinucleotides [109]. This modification conveys a signal for chromatin compaction and gene silencing [110]. DNA methylation is not a permanent mark, it can be actively removed by methylcytosine dioxygenases, called Ten-Eleven Translocation (TET) enzymes. This process is important for monocyte to dendritic cell differentiation [111]. During lineage commitment from haematopoietic progenitors, specific genomic regions are methylated and silenced, which is important for correct development [112].

Furthermore, the DNA double strand is tightly wrapped around protein complexes called nucleosomes, which are build from nine histone proteins (2x (H2A, H2B, H3, H4) and H1). This structure is strongly conserved across species and used to build up chromatin [113]. The tails of histone proteins are standing out from the complex and are therefore relatively accessible and mobile. They also contain a lot of positively charged amino acids that do not favour a known secondary structure. This part of the protein is used by the cells to imprint a genomic region with certain marks [114]. Chromatin can be further condensed into higher order structures, so called heterochromatin. This formation also plays an important role in the regulation of cellular phenotypes, as it decides whether cis-regulatory elements are accessible for transcription factor binding or not.

A third epigenetic element is constituted by non-coding RNAs [3]. These are divided into two main classes, long non coding RNAs (longer than 200 nucleotides) and short non coding RNAs (shorter than 200 nucleotides). In mammals there several groups of small non-coding RNAs, namely micro RNAs, enhancer RNAs, small nucleolar RNAs and Piwi-interacting RNAs (piRNAs) [115, 116, 117, 118, 119].

1.3.1 Histone Modifications

There are several modifications associated with open and active chromatin regions. Histone 3 lysine 27 acetylation (H3K27ac) is a general mark of active regions, as it is present at active promoters and enhancers [120]. Histone acetyltransferases (HAT) place the acetylation marks, whereas histone deacetylases (HDAC) are able to remove them again [121, 122]. H3K27ac is usually placed at regions already containing histone 3 lysine 4 monomethylation (H3K4me1) [123]. Methylation at H3K4, including mono-, di- and trimethylation, is mediated by the histone methyltransferase family SET/MLL [124] and removed by the lysine-specific demethylase 1A [125]. The combination of H3K4me1 and H3K27ac can be reliably used to identify cis regulatory regions [126, 127]. In contrast, promoters of actively transcribed genes are associated with trimethylation of histone 3 at lysine 4 (H3K4me3).

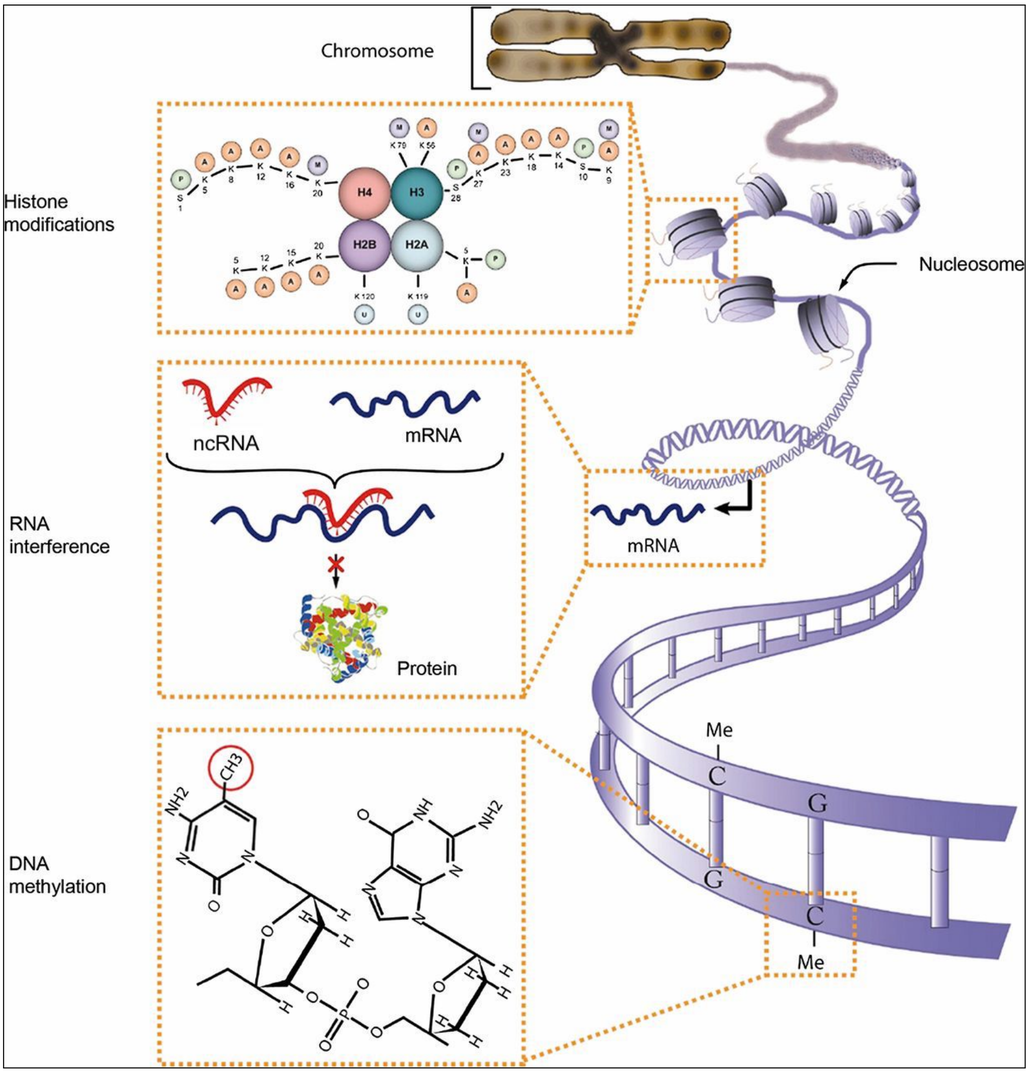

Figure 1.3: Scheme of epigenetic control, adapted from [3]

Removal of this mark is performed by the H3K4 demethylase KDM5B [128]. However there are also several marks for silenced, inactive regions. Heterochromatin is associated with histone 3 lysine 9 trimethylation (H3K9me3) [129], placed by the histone methyltransferase G9a and removed by JMJD2 [121]. These histone marks represent only a few examples of known histone modifications A schematic view of these modifications is shown in figure 1.3. In addition to enzymes setting or removing histone modifications, the cells express several Bromodomain (BRD) proteins, which recognize ϵ-N-lysine acetylation motifs on histone tails [130]. For example, BRD4 binds to acetylated lysine residues and recruits the P-TEFb complex to phosphorylate RNA polymerase II and enable transcription elongation [131].

1.3.2 Small non-coding RNAs

In recent years, the importance of small non-coding RNAs has been recognized. Several types of RNAs are counted as small non-coding RNAs, including miRNAs, piRNAs, snoRNAs and eRNAs. All of them perform their functions in the context of RNA-protein complexes [115]. Small nucleolar RNAs are another class of non coding RNAs, which are responsible for modifying other non coding RNAs, especially ribosomal RNAs. Canonical snoRNAs accumulate in the nucleolus, the main site of ribosome production. The nucleotide os the ribossomal RNA that is to be modified, depends on sequence complementarity to the snoRNA [116]. Recently it was discovered, that they can also give rise to further small non coding RNAs, which function like miRNAs [132].

MiRNAs are important for the regulation of mRNA stability and translation. MiRNA biogenesis starts with transcription of the miRNA gene by RNA polymerase II. The resulting pri-miRNA is cleaved by Drosha, which leads to the precursor miRNA. This molecule contains a hairpin structure, in which both strands of the miRNA are present. The precusor is exported from the nucleus to the cytoplasm, where it is processed by Dicer to create a duplex miRNA. One of the strands is incorporated into the argonaute protein complex, which gives rise to a functional miRNA-Protein complex that can target complementary and semi-complementary mRNAs [133]. Up to 70% of miRNA genes are located within introns and are expressed regulated like their host genes [133]. Binding of miRNAs to their targets leads either to it's degradation, destabilization, or to translational inhibition. Nowadays, there are several algorithms that try to predict targets for miRNAs. It is important that these programs take into account the conservation of the seed sequence, which is located in the nucleotides two to seven [134].

Closely related to miRNAs are the so-called piRNAs. These are characterized by their interaction with argonaute proteins of the PIWI-class and their independence from Dicer [135]. PiRNAs are responsible for silencing transposable elements in the genome. They are loaded into the piRISC complex, which mediates the endonucleolytic cleavage of transposon transcripts based on sequence complementarity. In murine germline cells, piRNAs mediate silencing also by inducing *de novo* DNA methylation [136]. piRNAs were first discovered in Drosophila germline cells, but they are also present in somatic tissues to repress retroviral elements [137].

The discovery that active enhancers are also transcribed by RNA polymerases, led to the identification of a new class of non coding RNAs, so called enhancer RNAs [117]. They are transcribed bidirectionally from the transcription start site (TSS) of enhancer sites [138]. ERNAs are short, nuclear and possess a 5' cap, but are generally not spliced or polyadenylated [117, 138]. They have a shorter half life than mRNA and are degraded by the exosome [138]. Whether these eRNAs are just a by-product of active enhancers or if they take part in regulating gene transcription is still unclear [139]. Signal dependent induction of eRNAs has been correlated to signal dependent expression levels of mRNAs [140, 141]. A few eRNAs have been shown to influence the expression levels of protein coding genes [142, 143, 144], indicating that at least some eRNAs perform specific functions.

Chapter 2

Objectives

Macrophages are professional phagocytes and play an important role in innate immunity. Resident tissue macrophages are strategically positioned sentinels to detect pathogen invasions, however they also perform many homoeostatic functions. Therefore, the deregulation of macrophages has been implicated in the development and progression of many diseases, including cancer. Their highly plastic nature allows them to adjust to their environment and perform needed functions accordingly. This characteristic is exploited, for example by cancer cells, to induce an anti-inflammatory, pro-angiogenic phenotype in tumour associated macrophages, however macrophages play an important role in many other diseases too. Therefore, understanding the reaction of macrophages to various stimuli is important for further characterization of diseased states and possible therapeutic responses.

Consequently, the aim of this dissertation was to investigate the plasticity of human monocytes and macrophages *in vitro* and *ex vivo*. Hence, the impact of several conditions on human monocytes during their differentiation to macrophages was analysed. In order to achieve this monocytes were obtained by leukapharesis and elutriation from the peripheral blood of healthy, voluntary donors. *In vitro* differentiation was induced using either recombinant M-CSF or human serum in RPMI medium on teflon bags or tissue culture treated plastic dishes. To this basic differentiation, several soluble mediators were added to stimulate the cells, for example TGF-β or salt. Another set of conditions included differentiation on plastic dishes coated with proteins from the ECM, such as collagen or fibronectin. A total of sixteen different settings were used.

In addition to these *in vitro* analysis, also primary human tissue macrophages were analysed. By collaborating with the St. Josef clinic in Regensburg, we were able to obtain placentas shortly after selective caesarian dissections, in order to isolate Hofbauer Cells, which are foetal macrophages located inside the chorionic villi of the tissue.

In order to gain a global overview of the different phenotypes, we analysed the transcriptome using RNA-sequencing, transcription factor binding with the ChIP-seq technique and the open chromatin landscape by an aassay of transposase accessible chromatin (ATAC). The latter protocol was established in the lab using several published methods [145, 146]. Additionally we used a special protocol of small RNA sequencing called miRNA-seq to check levels of mature miRNAs. In addition to these high throughput methods, a nine-colour FACS panel was established in order to determine known macrophage surface marker expression.

After identification of transcription factors important for specific phenotypes, we conducted knock down experiments of these in monocytes using siRNAs. Since these are non-proliferating cells the siRNA concentration remained constantly high over the seven day differentiation process to macrophages. Afterwards the knock down was checked using qPCR and Western Blot. If successful the cells were analysed again using the methods mentioned above.

Chapter 3

Materials and Methods

3.1 Materials

3.1.1 Equipment

Table 3.1: List of Instruments

Autoclave	Technomara, Fernwald, Germany
Avanti J25	Beckman Coulter, Munich Germany
BD LSRFortessa	BD Biosciences, Heidelberg, Germany
Biofuge Fresco	Heraeus, Hanau; Eppendorf, Hamburg, Germany
CASY	Innovatis/Roche, Mannheim, Germany
Centrifuge 5904 R	Eppendorf, Hamburg, Germany
Covaris S2	Covaris, Woburn, USA
Electrophoresis chamber	Biometra, Göttingen, Germany
Evos Digital Microscope	Thermo Fisher Scientific, Waltham, MA USA
Fusion Pulse	Vilber Lourmat, Frankfurt, Germany
GenePulser Xcell (Electroporator)	Bio-Rad, Munich, Germany
Incubator	Heraeus, Hanau; Eppendorf, Hamburg, Germany
J6M-E centrifuge	Beckman Coulter, Munich Germany
LightCycler480	Roche, Mannheim, Germany
Mastercycler Nexus X2	Eppendorf, Hamburg, Germany
Megafuge 3,0 R	Heraeus, Hanau; Eppendorf, Hamburg, Germany
Microscopes	Leitz, Jena, Germany
Mini Centrifuge	Carl Roth, Karlsruhe, Germany
Mini-PROTEAN Tetra System	Bio-Rad, Munich, Germany
Multifuge 3S-R	Heraeus, Hanau; Eppendorf, Hamburg, Germany
NanoDrop	Peqlab, Erlangen, Germany
Neubauer improved chamber	Marienfeld, Lauda Königshofen, Germany
NextSeq550	Illumina, San Diega, USA
Qubit 2.0 Fluorometer	Life Technologies/ Thermo Fisher Scientific, Waltham, MA USA
Realplex Mastercycler epGradient S	Eppendorf, Hamburg, Germany
sterile hood	Heraeu; Eppendorf, Hamburg, Germanys
TapeStation 2200	Agilent Technologies, Böblingen, Germany
ThermoMixer C	Eppendorf, Hamburg, Germany
Typhoon 9200	Amersham Biosciences, Germany
pH-Meter	Knick, Berlin, Germany
Picofuge	Heraeus, Hanau; Eppendorf, Hamburg, Germany
PowerPac Basic	Bio-Rad, Munich, Germany

Table 3.1: List of Instruments

Vortex Genie	Scientific Industries, New York, USA
water baths	Julabo, Seelbach, Germany
Whatman Fastblot B44	Biometra, Göttingen, Germany

3.1.2 Plastic Consumables

Table 3.2: List of Consumables

Cell culture Flasks	Costar corning, Wiesbaden, Germany
Cellcoat Collagen Type I 100mm	Greiner Bio-One, Kremsmünster, Austria
Cellcoat Fibronektin 100mm	Greiner Bio-One, Kremsmünster, Austria
Cellcoat Laminin 100mm	Greiner Bio-One, Kremsmünster, Austria
Cellstar cell culture dish 100mm	Greiner Bio-One, Kremsmünster, Austria
Cell strainer 100μM	Falcon, Heidelberg, Germany
Elektroporationsküvette 4mm kurze Elektrode	Biolab products, Bebensee, Germany
Falcon tubes (15, 50, 250ml)	Falcon, Heidelberg, German
Heat sealing Film	Eppendorf, Hamburg, Germany
LightCycler® 480 Multiwell Plate 384, clear	Roche, Mannheim, Germany
Loading tips for TapeStation 2200	Agilent Technologies, Böblingen, Germany
Optical tube strips (8x Strip)	Agilent Technologies, Böblingen, Germany
Optical tube strip caps (8x Strip)	Agilent Technologies, Böblingen, Germany
Qubit assay tubes	Invitrogen, Carlsbad, USA
Reagiergefäß 1,5ml DNA LowBind	Sarstedt, Nürnbrecht, Germany
Reagiergefäß 0,5ml DNA LowBind	Sarstedt, Nürnbrecht, Germany
Reagiergefäß 2ml PP	Sarstedt, Nürnbrecht, Germany
Safe Seal SurPhob Spitzen (10μl, 100μl, 1250 μl)	Biozym, Hessisch Oldendorf, Germany
SpinX Centrifuge Filter 0.45μm Cellulose Acetate	Costar Corning, Wiesbaden, Germany
Stabpipette (2ml, 5ml, 10ml, 25ml, 50ml)	Costar Corning, Wiesbaden, Germany
twin.tec® real-time PCR-Platten	Eppendorf, Hamburg, Germany
Zetuvit E Saugkompressen	HARTMANN International, Heidenheim Germany

3.1.3 Chemicals

All chemicals were ordered from Sigma Aldrich (Taufkirchen, Germany) or Merck (Darmstadt, Germany), unless noted differently.

3.1.4 Enzymes, Reagents and Molecular Kits

Table 3.3: List of Enzymes, Reagents and molecular Kits

2-Mercaptoethanol	Gibco/ Life Technologies, Carlsbad, USA; GE Healthcare, Chalfont St. Hiles, UK
50bp Ladder	New England Biolabs (NEB) , Ipswich, Massachusetts, USA
Accutase solution	Sigma Aldrich, Taufkirchen, Germany
Active human adipponectin full length protein	Abcam, Cambridge, United Kingdom
AMPure XP Beads	Beckman Coulter, Munich Germany
Anti-Mouse Ig, κ/Negative Control Compensation Particles Set	BD Biosciences, Heidelberg, Germany
Anti-Rat and Anti-Hamster Ig κ/Negative Control Compensation Particles Set	BD Biosciences, Heidelberg, Germany
Ammonium peroxodisulfate (APS)	MERCK, Darmstadt, Germany
BD FACSFlow™Sheath Fluid	BD Biosciences, Heidelberg, Germany
BD Horizon Brilliant Stain Buffer	BD Biosciences, Heidelberg, Germany
Bovine Serum Albumin (BSA)	Sigma Aldrich, Taufkirchen, Germany
Betaine Solution	Sigma Aldrich, Taufkirchen, Germany
Calcium chloride solution	Sigma Aldrich, Taufkirchen, Germany
Collagen Type IV from human placenta	Sigma Aldrich, Taufkirchen, Germany
Collagenase A from Clostridium Histolyticum	Sigma Aldrich, Taufkirchen, Germany
Deoxyribonuclease	Sigma Aldrich, Taufkirchen, Germany
Deoxyribonucleoside Trisphosphate Set PCR grade (dNTPs)	Roche, Mannheim, Germany
Digitonin	Promega, Madison, USA
Disuccinimide Glutarate	Thermo Fisher Scientific, Waltham, USA
DMSO	Sigma Aldrich, Taufkirchen, Germany
Dulbecco's phosphate buffered saline (PBS)	Sigma Aldrich, Taufkirchen, Germany
ECL Prime Western Blotting Detection Reagent	GE Healthcare, Chalfont St. Giles, UK
FACS Clean Solution	BD Biosciences, Heidelberg, Germany
FACS Rinse Solution	BD Biosciences, Heidelberg, Germany
FcR Blocking Reagent, human	Miltenyi Biotec, Bergisch Gladbach, Germany
Fetal Calf Serum	Gibco/ Life Technologies, Carlsbad, USA; GE Healthcare, Chalfont St. Hiles, UK
Fibronectin from human plasma	Sigma Aldrich, Taufkirchen, Germany
Formaldehyde solution (16%) methanol free	Thermo Fisher Scientific, Waltham, MA USA

Table 3.3: List of Enzymes, Reagents and molecular Kits

GM-CSF	Berlex, Seattle, USA
Glycogen	Ambion/Life Technologies, Carlsbad, USA
Hanks' balanced salt solution (Hank)	Sigma Aldrich, Taufkirchen, Germany
Hemin bovine	Sigma Aldrich, Taufkirchen, Germany
HEPES	Gibco/7 Life Technologies, Carlsbad, USA
High Sensitivity D1000 ScreenTape	Agilent Technologies, Böblingen, Germany
High Sensitivity D1000 Ladder	Agilent Technologies, Böblingen, Germany
High Sensitivity D1000 Sample Buffer	Agilent Technologies, Böblingen, Germany
Human AB-Serum	Deutsches Rotes Kreutz
Hydrogen Peroxide	Merck, Darmstadt, Germany
IL-4 (Human)	Promokine, Heidelberg, Germany
Immobilon Transfer Membranse	Merck, Darmstadt, Germany
Kapa Library Quant Illumina	Kapa Biosystems/ Roche, Mannheim, Germany
Laminin from human placenta	Sigma Aldrich, Taufkirchen, Germany
L-Glutamine	Biochrom , Berlin, Germany
Magnesium Sulfate Solution	Sigma Aldrich, Taufkirchen, Germany
Methanol	Merck, Darmstadt, Germany
miRIDIAN microRNA has-miR-148a-3p hairpin inhibitor	Dharmacon, Lafayette, Colorado, USA
miRIDIAN microRNA Hairpin Inhibitor negative control 1	Dharmacon, Lafayette, Colorado, USA
miRNeasy Mini Kit	Qiagen, Hilden, Germany
M-MLV Reverse Transkriptase, RNase H minus, point mutant	Promega, Madison, USA
M-CSF recombinant protein	Cetus (Australia)
Monarch PCR & DNA Cleanup Kit	New England Biolabs (NEB), Ipswich, Massachusetts, USA
NEBNext® Ultra II DNA Library Prep Kit for Illumina®	New England Biolabs (NEB), Ipswich, Massachusetts, USA
Nextera DNA Library Prep Kit	Illumina, San Diega, USA
Nextera Index Kit	Illumina, San Diega, USA
Non-Essential Amino Acids	Sigma Aldrich, Taufkirchen, Germany
Nonidet P-40 substitute	Roche, Mannheim, Germany
Opti-Mem	Life Technologies, Böblingen, Germany
O'RangeRuler 20bp DNA ladder	Thermo Fisher Scientific, Waltham, MA USA
Pancoll human	PAN Biotech, Aidenbach, Germany
Percoll	GE Healthcare, Chalfont St. Giles, UK
Gel Loading Dye Purple (6x)	New England Biolabs (NEB), Ipswich, Massachusetts, USA
Penicillin	Gibco/Life Technologies, Carlsbad, USA

Table 3.3: List of Enzymes, Reagents and molecular Kits

Phusion Hot Start High-Fidelity DNA Polymerase	Fermentas, Walthamm USA
Precision Plus Standard	Bio-Rad, Munich, Germany
Protease Inhibitor	Roche, Basel, Switzerland
Protein G Sepharose 4 Fast Flow	GE Healthcare, Freiburg, Germany
Proteinase K	Roche, Basel, Switzerland
QuantiFast SYBR® Green Kit	Qiagen, Hilden, Germany
Qubit dsDNA HS Assay Kit	Invitrogen, Carlsbad, USA
Random Decamers	Life Technologies/ Thermo Fisher Scientific, Waltham, MA USA
Re-Blot Plus Mild Solution (10x)	Merck, Darmstadt, Germany
Ribo-Zero Gold rRNA removal kit	Illumina, San Diega, USA
RNA ScreenTape	Agilent Technologies, Böblingen, Germany
RNA ScreenTape Sample Buffer	Agilent Technologies, Böblingen, Germany
RNA ScreenTape Ladder	Agilent Technologies, Böblingen, Germany
RNase für ChIP	Qiagen, Hilden, Germany
RNeasy Mini Kit	Qiagen, Hilden, Germany
Rotiphorese Gel 30 Acrylamid Solution	Carl Roth, Karlsruhe, Germany
RPMI-Medium 1640	Life Technologies/ Thermo Fisher Scientific, Waltham, MA USA
RPMI-Medium 1640 without Phenolred	Life Technologies , Böblingen, Germany
Sandoglobulin	Octapharma, Lachen SZ, Switzerland
Serotonin hydrochloride	Sigma Aldrich, Taufkirchen, Germany
Sepharose CL-4B beads	Sigma Aldrich, Taufkirchen, Germany
SequaGel -UreaGel Conentrate	National Diagnostics, Atlanta, USA
SequaGel Diluent	National Diagnostics, Atlanta, USA
SequaGel UreaGel Buffer	National Diagnostics, Atlanta, USA
skimmed Milk Powder easily soluble	Sucofin, Germany
Sodium dodecylsulfate (SDS)	Sigma Aldric, Taufkirchen, Germanyh
Sodium pyruvate	Sigma Aldrich, Taufkirchen, Germany
Streptomycin	Gibco, Life Technologies, Carlsbad, USA
SuperScript II RNase H-Reverse Transcriptase	Invitrogen, Carlsbad, USA
SuperScript III First Strand cDNA Super Mix	Invtirogen, Carlsbad, USA
T4 RNA-Ligase 1	New England Biolabs (NEB), Ipswich, Massachusetts, USA
TEMED	Sigma Aldrich, St. Louis, USA
Transforming Growth Factor beta (TGF-beta)	PeproTech
TRIZMA Base	Sigma Aldrich, Taufkirchen, Germany

Table 3.3: List of Enzymes, Reagents and molecular Kits

TRIzol® Reagent	Life Technologies/ Thermo Fisher Scientific, Waltham, MA USA
Truseq Stranded Total RNA Library Prep Kit with Ribo-Zero Human/-Mouse/Rat (49samples)	Illumina, San Diega, USA
Trypanblu solution 0.4%	Sigma Aldrich, Taufkirchen, Germany
Tween20 solution	Sigma Aldrich, Taufkirchen, Germany
UltraPure ™ RNase/DNase-Free Distilled Water	Life Technologies/ Thermo Fisher Scientific, Waltham, MA USA
Vitamins	Sigma Aldrich, Taufkirchen, Germany
Whatman 3MM CHR	GE Healthcare, Chalfont St. Giles, UK
7-AAD	BD Pharmingen, Heidelberg, Germany

3.1.5 Buffers and Solutions

Adiponectin Solution

Lyophilized protein was dissolved in 50μl water, diluted with 150μl RPMI medium (which included 2% AB-Serum) and aliquoted (25ng/μl). The protein solution was stored at -20°C.

Collagen Type IV Solution

5mg protein powder were dissolved in 4ml 0.5M acetic acid for 2h on ice. When there was no powder left, the solution was transferred to an eppendorf cup, and the bottle washed with 1ml 0.5M acetic acid. This results in 5ml of 5mg/ml collagen IV solution. The solution was aliquoted (1ml) and stored at -20°C.

Hermin

100mg were dissolved in 4ml 0.15M NaCl + 10% NH_4OH, sterile filtrated, aliquoted and stored at -20°C.

5M NaCl

14.61g salt were dissolved in 50ml water, steril filtrated, autoclaved and stored at RT.

Serotonin

10mg dissolved in 1ml of 1x PBS, sterile filtrated and immediately used for the experiment.

FACS Wash Buffer

500ml of PBS (Sigma Aldrich) with 10ml FCS (2%).

10% APS

100mg APS were dissolved in 1ml water.

5x Lämmli Buffer

216g Glycin (0.95M), 15g SDS (0.5%) and 15g Tris (40mM) were dissolved in a total volume of 3L water. Store at room temperature protected from light.

2x SDS Buffer

2g SDS were dissolved in water, add 10mg Bromophenol Blue (0.02%), 10ml Glycerin (20%), 5ml 1.07M Tris/HCl pH 6.8 (107mM) and 5ml β-Mercaptoethanol (10%). Store at 4°C.

10% SDS Solution

10g SDS dissolved in total volume of 100ml water.

1.5M Tris/HCl pH 8.8

90.83g Tris were dissolved in 400ml water. pH was set to 8.8 using concentrated HCl and volume was filled with water to 500ml.

0.5M Tris/HCl pH 6.8

30g Tris were dissolved in water, set pH to 6.8 using HCl and fill up to 500ml using water.

1.07M Tris/HCl pH 6.8:

13g were dissolved in water, set pH to 6.8 and fill up to 100ml with distilled water.

Western Blot Buffer A

36.3g Tris were dissolved in 800ml distilled water, afterwards add 200ml methanol under the fume hood. The buffer can be stored at room temperature.

Western Blot Buffer B

3.03g Tris were dissolved in 800ml distilled water, add 200ml methanol under the fume hood. Store the buffer at room temperature.

Western Blot Buffer C:

5.2g caproic acid were dissolved in water in a total volume of 800ml. Add 200ml Methanol to reach end volume of 1L. Store buffer at room temperature.

10x TBS:

45.8g Tris and 175.5g NaCl were dissolved in water. Set pH to 7.4 using HCl, afterwards fill up to 2L using distilled water. Store buffer at room temperature.

ATAC-RSB

Table 3.4: ATAC Resuspension Buffer

Reagents	Final Concentration	Volume needed for 50ml
1M Tris/HCl pH 7.4	10mM	500μl
5M NaCl	10mM	100μl
1M $MgCl_2$	3mM	150μl
Water	NA	49.25ml

ATAC Lysis Buffer

Table 3.5: ATAC Lysis Buffer

Reagents	Final Concentration	Volume for 1ml
ATAC-RSB	NA	970μl
NP-40 (10%)	0.1%	10μl
Tween20 (10%)	0.1%	10μl
Digitonin (1%)	0.01%	10μl

2.625M Glycine for ChIP

9.85g Glycine were dissolved in 50ml sterile H_2O, then the solution was filtered in order to sterilize it.

100mM PMSF Solution

1.714g PMSF were dissolved in 100ml pure isopropoanol.

Cell Buffer Mix (CBM)

Table 3.6: Cell Buffer Mix for ChIP

Reagents	Volume needed for 100ml	Final Concentration
HEPES / KOH (1M), pH 7.9	1ml	10mM
KCL (2M)	4.35ml	85mM
EDTA (500mM, PH 8)	200μl	1mM
Water	81.25ml	NA

Nuclear Lysis Buffer for ChIP (L2)

Table 3.7: Nuclear Lysis Buffer for ChIP (L2)

Reagents	Volume needed for 97ml	Final Concentration
Tris/HCl (1M, pH 7.4)	2ml	20mM
SDS (20%)	5ml	1%
Empigen BB (35%)	1.43ml	0.5%
EDTA (0.5M, pH 8.0)	2ml	10mM
Water	83.57ml	NA

Dilution Buffer for ChIP (DB)

Table 3.8: Dilution Buffer for ChIP (DB)

Reagents	Volume needed for 97ml	Final Concentration
Tris/HCl (1M, pH 7.4)	2ml	20mM
NaCl (5M)	2ml	100mM
EDTA (0.5M, pH 8.0)	400μl	2mM
Triton X-100 (10%)	5ml	0.5%
Water	97ml	NA

Wash Buffer 1 for ChIP

Table 3.9: Wash Buffer 1 for ChIP

Reagents	Volume for 100ml	Final Concentration
Tris/HCl (1M, pH 7.4)	2ml	20mM
NaCl (5M)	3ml	150mM
SDS (20%)	500μl	0.1%
Triton X-100 (10%)	10ml	1%
EDTA (0.5M, pH 8.0)	400μl	2mM
Water	84.1ml	NA

Wash Buffer 2 for ChIP

Table 3.10: Wash Buffer 2 for ChIP

Reagents	Volume for 100ml	Final Concentration
Tris/HCl (1M, pH 7.4)	2ml	20mM
NaCl (5M)	10ml	500mM
SDS (20%)	500μl	0.1%
Triton X-100 (10%)	10ml	1%

EDTA (0.5M, pH 8.0)	400μl	2mM
Water	77.6ml	NA

Wash Buffer 3 for ChIP

Table 3.11: Wash Buffer 3 for ChIP

Reagents	**Volume for 100ml**	**Final Concentration**
Tris/HCl (1M, pH 7.4)	1ml	10mM
LiCl (2.5M)	10ml	250mM
NP-40 (10%)	10ml	1%
Deoxycholate (10%)	10ml	1%
EDTA (0.5M, pH 8.0)	200μl	1mM
Water	68.8ml	NA

TE Buffer pH 8.0

Table 3.12: TE-Buffer for ChIP

Reagents	**Volume for 100ml**	**Final Concentration**
Tris/HCl (1M, pH 8.0)	1ml	10mM
EDTA (0.5M, pH 8.0)	250μl	1%
Water	98.8ml	NA

Elution Buffer for ChIP

Table 3.13: Elution Buffer for ChIP (EB)

Reagents	**Volume for 5ml**	**Final Concentration**
$NaHCO_3$ (1M)	500μl	0.1M
SDS (10%)	500μl	1%
Water	4.25ml	NA

10x Ligation Buffer (miRNA-seq)

Table 3.14: 10x Ligation Buffer for miRNA-seq

Reagent	**Concentration**
$MgCl_2$	0.1M
β-Mercaptoethanol	0.1M
Tris pH 7.6	0.5M
BSA	1mg/ml

Elution Buffer for miRNA-seq

Table 3.15: 10x Elution Buffer for miRNA-seq

Reagent	Concentration
NaCl	300mM
EDTA	2mM

3.1.6 Oligonucleotides

Oligonucleotides were synthesized and purified by Sigma Aldrich.

Table 3.16: List of Olignucleotides

Gene	Primer Sequence
BHLHE41-F	TGCTTTACAGAATGGGGAGCGATCT
BHLHE41-R	TCCCGAGTGGAACGCATCCAA
CHIT1-F	TCATCCAGACGCTACGGCAG
CHIT1-R	TGGCAGAACGTGTCTTGTCCAG
GAPDH-F	GAAATCCCATCACCATCTTCCAGG
GAPDH-R	GAGCCCCAGCCTTCTCCATG
LYVE1-F	TGCTGCAGCTGGTCTTGGATTT
LYVE1-R	TGGTTTCGATCATTTCCTTCTGCTGAT

3.1.7 siRNAs

siRNAs were designed and synthesized by Axolabs. They were delivered as a 50μM solution or as lyophilised powder that was dissolved in ultrapure water at 1mg/ml. These were stored at -80°.
miRIDIAN microRNA Hairpin Inhibitor for the knock down of miRNA-148a-3p were designed and ordered from Dharmacom.

3.1.8 Antibodies

FACS

Table 3.17: List of Antibodies used for FACS Analysis

Antibody	Clone	Company
mouse anti human CD3	UCHT1	BD Biosciences, Heidelberg, Germany
mouse anti human CD11b	561015	BD Biosciences, Heidelberg, Germany
mouse anti human CD14	ICRF44	BD Biosciences, Heidelberg, Germany
mouse anti human CD15	HI98	BD Biosciences, Heidelberg, Germany

Table 3.17: List of Antibodies used for FACS Analysis

mouse anti human CD16	3G8	BD Biosciences, Heidelberg, Germany
mouse anti human CD19	HIB19	BD Biosciences, Heidelberg, Germany
mouse anti human CD64	10.1	BD Biosciences, Heidelberg, Germany
mouse anti human CD71	M-A712	BD Biosciences, Heidelberg, Germany
rat anti human CD115	9-4D2-1E4	BD Biosciences, Heidelberg, Germany
mouse anti human CD163	GHI/61	BD Biosciences, Heidelberg, Germany
mouse anti human HLADR	G46-6	BD Biosciences, Heidelberg, Germany
anti human EGF Receptor	13/EGFR	BD Transduction Laboratories, Heidelberg, Germany
anti human CD10 PE/Cy7	HI10a	Biolegend, San Diego, USA
anti human CD163 APC	eBioGHI/61	eBioscience/Thermo Fisher Scientific, Waltham, MA USA

Western Blot

Table 3.18: Antibodies used for Western Blot

Antibody	**Order Number**	**Company**	**Dilution**
Mouse Anit-DEC2 (E-4)	sc-373763	Santa Cruz Biotechnology, Dallas, USA	1:500
Rabbit Anti-Aktin	A2066-2ML	Sigma Aldrich, Taufkirchen, Gemany	1:2000
Goat Anti-Mouse IgG HRP coupled	P0447	Dako/Agilent, Böblingen, Germany	1:5000
Goat Anti-Rabbit IgG HRP coupled	P0448	Dako/Agilent, Böblingen, German	1:10000

ChIP

Table 3.19: Antibdies used for ChIP-seq

Antibody	**Company**	**Order Number**	**Concentration**
Dec-2 (H-71)	Santa Cruz, Dallas, USA	sc-32853-x	200μg/0.1ml
H3K27ac	Abcam, Cambridge, UK	ab4729	1mg/ml
c-MAF	Santa Cruz, Dallas, USA	sc-7866-X	200μg/0.1ml
MAFB	Santa Cruz, Dallas, USA	sc-10022-X	200μg/0.1ml

3.2 Methods

3.2.1 Cell Culture

Collagen IV coating of tissue culture treated dishes

In order to analyse the effect of collagen type IV on monocytes, tissue culture treated cell culture dishes (from corning) were coated with this protein. The collagen powder was dissolved in 0.5M acetic acid to obtain a 1mg/ml concentrated solution. In order to coat the plates with 5μg/cm^2 protein each 10cm plate was incubated with 392.5μl collagen solution (1mg/ml) in 2ml 1x PBS and incubated in the plate for several hours at 37°C, so that the collagen could form a fibrillar structure at the plate bottom. Subsequently the plate was washed with sterile PBS. The plates were dried in the sterile hood, used immediately or kept at 4° maximal two weeks.

Collagen IV and Laminin coating of tissue culture treated dishes

To determine which of these proteins has the stronger effect on macrophages, we coated plates with a mixture of collagen type iv and laminin. Here 392.5μl collagen was combined with 200μl laminin solution in 2.5ml sterile PBS on a 100mm cell culture dish. The plate was incubated at 37 °C for up to seven hours, afterwards the plates were washed twice with sterile PBS and air dried under the sterile hood. They were used immediately or stored at 4 °C for up to four weeks.

Isolation of Monocytes from peripheral blood by Elutriation

In order to be able to analyse primary human monocytes, mononuclear cells (PBMC) from the peripheral blood of healthy volunteers were enriched and collected by leukapharesis (performed by the compartment Transfusionsmedizin at the university clinic Regensburg). The cells were further purified using a gradient centrifugation over Pancoll, where they formed a layer between pancoll and PBS. PBMCs were separated into their different subsets via elutriation. For this procedure a J6M-E centrifuge, JE 5.0 rotor and a 50ml chamber was used. In this machine the cells were separated according to their size. Shortly, the chamber and tubes were sterilized by peroxide, washed with sterile PBS and calibrated using Hanks solution. The cell suspension flows into the machine at a flow rate of 52ml/min, 4°C. By increasing the speed, different fractions of cells can be harvested (see Table 3.20).

Table 3.20: Fractions obtained by Elutriation

Fraction	Volume(ml)	Flow Rate (ml/min)	Cell type
Ia	1000	52	Thrombocytes
Ib	1000	57	
IIa	1000	64	B- and T-Lymphocytes
IIb	500	74	

Table 3.20: Fractions obtained by Elutriation

IIc	400	82	
IId	400	92	
III	800	130	Monocytes

The fraction containing monocytes was centrifuged (300g, 4°C, 8min) and solved in 40ml of RPMI Medium. The solution was counted using the CASY counter.

In vitro differentiation of monocytes into macrophages/ dendritic cells

Monocytes, obtained by elutriation (see section 3.2.1), were differentiated into macrophages in RPMI Medium containing 2mM L-Glutamine, non-essential amino acids (1x), 1mM sodium pyruvate, 2ml vitamins (1x), 10U/ml penicillin/ streptomycin and 500μM 2-mercaptoethanol for 7d in an incubator set to 37°C, 5%CO_2, and 95% humidity.
Within this scheme of differentiation, monocytes were exposed to 16 different conditions, in order to obtain polarized macrophages, these are described in table 3.21.

Table 3.21: Conditions for macrophage differentiation

Name	**Condition**
M-CSF	Differentiation by 10% FCS and 100ng/ml recombinant M-CSF on TcPlastic
GM-CSF	Differentiation by 10% FCS and 280U/ml recombinant GM-CSF on TcPlastic
iDC	280U/ml GM-CSF, 20U/ml IL-4 and 10% FCS on TcPlastic
AB-Teflon	RPMI-Medium with 2% human AB-Serum in Teflon Bags
TGF	10ng/ml TGF-β, RPMI-Medium with 2% human AB-Serum in Teflon Bags
Hemin	20μM Hermin, RPMI-Medium with 2% human AB-Serum in Teflon Bags
Adiponectin	1μg/ml Adiponectin, RPMI-Medium with 2% human AB-Serum in Teflon Bags
Serotonin	10μM Serotonin, RPMI-Medium with 2% human AB-Serum in Teflon Bags
NaCl	40mM NaCl, RPMI-Medium with 2% human AB-Serum in Teflon Bags
IgG	0.5% Sandoglobulin, RPMI-Medium with 2% human AB-Serum in Teflon Bags
AB-TcPlastic	Tissue culture treated cell culture dishes (TcPlastic) with RPMI, 2% human AB-Serum and 100ng/ml M-CSF
Collagen I	Collagen Type I coated dishes with RPMI, 2% human AB-Serum and 100ng/ml M-CSF
Collagen IV	Collagen Type I coated dishes with RPMI, 2% human AB-Serum and 100ng/ml M-CSF

Table 3.21: Conditions for macrophage differentiation

Fibronectin	Fibronectin coated dishes with RPMI, 2% human AB-Serum and 100ng/ml M-CSF
Laminin	Laminin coated dishes with RPMI, 2% human AB-Serum and 100ng/ml M-CSF
Rolling Bottles	Cells were kept in rolling bottles, so that they could not adhere to the surface, with RPMI, 2% human AB-Serum and 100ng/ml M-CSF

After differentiation, cells were harvested for RNA isolation, ChIP-seq and ATAC-seq analysis. For this the cells were either scraped off the surface (cells cultured in teflon bags) or detached using the Accutase Solution (cells cultured in plates). Macrophages were counted with the CASY (settings: CL 9.6, CR 30, NL 5.4, NR 30, Volume 50μl, Dilution 2*10^{-2}) and distributed for the different experiments.

3.2.2 Isolation of Hofbauer Cells from term placenta

In addition to *in vitro* generated macrophages we wanted to profile primary tissue macrophages. for this we chose hofbauer cells, which are the resident macrophages in placenta. The protocol described herein is an adaptation of previous work published by Tang et al [39]. We received placentas in cooperation with the St. Josef clinic in Regensburg, after informed consent of donors.

Dissection of chorionic villi

The placenta is placed within a sterile hood in a box, in order to minimize its contamination with blood. Most of the blood and clots were removed by wiping the tissue with sterile pads. Afterwards the maternal membrane was removed using forceps and scalpels, so that the underlying villous tissue could be obtained. The villi were immediately transferred to a petri-dish filled with cool PBS (4 °C) and stored on ice and maternal blood was washed away. After enough tissue was collected in this way (approximately 30g), the villous tissue was scraped using a scalpel. Large white vessels were discarded, only the smaller vessels were kept. The scraped villous tissue was washed in a sterilized metal until the PBS ran clear. This scraped, washed tissue was weighed to calculate the amount of digestion mix needed.

Serial enzymatic digestion with trypsin and collagenase A

20g of tissue were taken up in 15ml of the following solution and divided into four 15ml falcon tubes.

- 0.25% Trypsin
- 0.2% DNAseI
- 25mM HEPES
- 2mM $CaCl_2$

- 0.8mM $MgSO_4$

This first digestion was carried out for 15min at 37°C in the Thermomix at 800rpm. Nonetheless the tubes had to be shaken manually every 5min in order to prevent sedimentation. Remaining tissue was recovered using a 100μm sieve. The digestion with the trypsin mix was repeated, but with 35ml solution for 60min at 37°C, at 800rpm. Following this step the remaining tissue was again recovered, washed 3x with PBS and used for the digestion using Collagenase in the following mix:

- 1mg/ml collagenase A
- 0.2mg/ml DNAseI
- RPMI-medium
- 25mM HEPES
- 0.8mM $MgSO_4$
- 5% FCS

The digestion with collagenase A was carried out again at 37°C for 1h at 800rpm. During this incubation step the percoll gradient was prepared in falcon tubes, to enrich the hofbauer cells a gradient of 40% / 20% was used. After the collagenase digest, resulting cell suspension was recovered in falcon tubes on ice by catching the flow-through through on another 100μm sieve. The remaining tissue was washed three times with PBS before it was discarded. Cells were centrifuged at 4°C, 300g for 8 minutes. After washing the cell pellet with RPMI medium they were loaded onto the percoll gradient. This was centrifuged at 700g, 23°C for 30min. Hofbauer cells accumulated between the two percoll phases, where they were harvested using a 10ml stripette and immediately transferred to 4°C cold fresh RPMI medium. The resulting cell suspension was centrifuged again at 4°, 300g for 8 minutes and washed using FACS Buffer. This solution was filtered through a 70μm cell strainer to remove larger clumps of cells before FACS sorting. For that they were stained using an antibody against CD163 conjugated to APC. Hofbauer Cells were sorted for a high expression of CD163, with help of Rüdiger Eder (Working Group of PD. Dr. Petra Hoffmann and Prof. Dr. Matthias Edinger, Internal Medicine III university clinic Regensburg). The sorted population was used for ATAC-seq and RNA Isolation.

3.2.3 Electroporation of primary monocytes

This method was used to introduce siRNAs into primary monocytes for a transient knock down of specific target genes.

Before electroporation enough cell medium (RPMI with added 2% AB-serum and 100ng/ml M-CSF) for all tests, was warmed to 37°C in cell culture plates. Also colourless RPMI (without indicator) and the Opti-MEM were warmed to room temperature. For each transfection nine million monocytes were electroporated with 9μg siRNA. First step was to centrifuge them at 500g, room temperature for 10min. The supernatant was discarded and the cell pellet was washed first with warm colourless RPMI, then with Opti-MEM. Following the next centrifugation the supernatant was completely removed by inverting the tube on a sterile pad and cell pellet suspended in Opti-Mem, so that 9 million monocytes were contained in 200μl. For each test 200μl cell suspension was combined in one cuvette with siRNA and shocked at 400V, for 5ms, 1 pulse. Immediately afterwards the cells were transferred into the pre-warmed medium. They were kept at 37°C, 95% H_2O, 5% CO_2 for seven days until the differentiation into macrophages was complete. Afterwards they were harvested for RNA isolation and protein lysastes for western blot.

3.2.4 FACS Staining

Flow cytometry was used to assess the surface expression of selected markers.
For each staining 1 million cells were used and all centrifugation steps were done at 4°C, 300g for 8min. In the first step all cell populations were washed using FACS buffer. Afterwards 1 million cells were suspended in 40μl FACS Buffer to which 10μl of FcR Blocking reagent were added. This mixture was incubated on ice for 10min. All antibodies were used according to manufacturers instructions. As several brilliant violet stainings were used, 50μl Brilliant Stain Buffer was added to each sample. The blocked cells were added to the prepared antibody cocktail and incubated for 30min at 4°C. Afterwards they were washed twice using FACS Buffer and in the last step were taken up in 200μl. All samples were measured at the LSR Fortessa of the department of Internal Medicine I at the university clinic of Regensburg. 10 minutes before measuring 5μl of 7-AAD solution were added to each tube.
As a 9 colour panel was used, the fluorescent spillover in different channels had to be compensated for. This was done using compensation beads and the same amount of antibody as in the cell staining. For this 1 drop of beads was taken up in 300μl of 1x PBS, under the sterile hood, and incubated at RT with the selected antibody, in the dark, for 20 minutes. Compensation was measured according to instructions of the BD software and the values calculated by it were used to correct for fluorescent spillover. In all experiments in this work the same values were used.
As negative controls unstained cells and fluorescence minus one controls (FMO) were used. For all of them 50000 events were recorded and the data exported as fcs files. Analysis was done using R in a bash script (see Section 8)
In the first step all fcs-files exported from BD-Diva software imported into R into the flowSet class set. The raw fluorescence values are stored in this way, therefore the compensation, which was calculated during measurement of the samples, had to be applied to the data. Afterwards compensated values were transformed in R using the Logicle transformation. It is an improved version of the hyperbolic sine function that gives appropriate, interpretable representations of the data. This command derives its parameters directly from the distribution of the raw data and leads therefore to optimal display of it [147]. Subsequently, the data was gated on several parameters. At first gating in the forward-, sideward-scatter debris was removed. Afterwards single cell events were selected in the FSC-H, FSC-A scatterplot. In the last gating step dead cells were removed using the dye 7-AAD.
With these gated populations the expression of measured surface markers was analysed, for this the data was displayed in histograms. To do so gated data was transferred from

flowSet environment into data frames. For further analysis the median fluorescence intensity (MFI) was calculated for each sample and marker. For this the MFI of the corresponding FMO control was subtracted from the sample MFI. Furthermore the percentage of positive cells compared to FMO and unstained controls was calculated. For this stained populations were overlaid with controls in scatterplots for each marker. The cut-off, which was used to define the amount of positive cells in the stained sample, was set, so that both controls were negative. All percentages were collected into one file, and used for a dotplot in ggplot2.

3.2.5 Western Blot

Western Blot kwas used to analyse protein levels within cell populations.

Preparing Cell Lysates

For this analysis cells were lysed in 2x SDS Buffer, so that 3 million cells were contained in 200μl buffer. The samples were mixed and incubated at 95°C for 10min. Afterwards the lysates were kept at -80°C.

Preparing the Polyacrylamid Gel

The proteins were separated by size on a 12% Polyacrylamid (PAA) gel, which was prepared as described in the tables 3.22 and 3.23. For this the gel chamber from Bio-Rad was used as instructed by the manufacturer.

Table 3.22: Separation Gel

Reagent	**10ml**
Tris/HCl pH 8.8 (1.5M)	2.5ml
SDS (10%)	200μl
AA (30%)	4ml
Water	3.3ml
TEMED	10μl
APS (10%)	100μl

Table 3.23: Collection Gel

Reagent	**5ml**
Tris/HCl pH 6.8 (1.5M)	630μl
SDS (10%)	50μl
AA (30%)	830μl
Water	3.4ml
TEMED	5μl
APS (10%)	50μl

This solution was poured into the chamber until approximately 1cm below the end of the chamber. On top of this Isopropanol (70%) was added in order to obtain a straight end of the separation gel. The polymerization was allowed to run for 30min. In the meantime the collection gel, as described in Table 3.23, was prepared.
From the chamber the isopropanol was removed and the space carefully dried using small pieces of whatman paper. On top of the separation gel the collection gel was poured, the comb for the sample pockets was inserted and the gel polymerized again for 30min. Afterwards it was kept at 4°C wrapped in wet paper tissues and plastic wrap.

Running the Gel

The gels were run in 1x Lämmli buffer (diluted from the 5x stock using de-ionized water), before loading the samples all slots were flushed with Lämmli buffer. In order to gauge the size of detected proteins, 7.5μl of the Precision Plus standard were also loaded onto the gel. For each sample 20μl were used. At first the gels were run at 80V until the samples entered the separation gel, then the voltage was increased to 100V. It was run until the bromphenolblue band nearly ran out of the gel. Following this size separation, the proteins were transferred from the gel to a hydrophobic membrane.

Blotting

For each blotting, nine pieces of 9x7cm whatman paper and one piece of membrane were needed. Four containers were filled with isopropanol, buffer A, buffer B and buffer C respectively. The membrane was kept in isopropanol until usage. 3 pieces of whatman paper were soaked with buffer A and put onto the blotting chamber. Now three whatman papers were soaked in buffer B and put on top of the pieces soaked in buffer A. The membrane was washed in buffer B before it went on top of the stacked whatman papers. 600μl buffer B were pipetted on top of the membrane, before the gel was carefully mounted on top of the membrane. As the last layer 3 whatman papers were soaked in buffer C and added on top. With a 10ml stripette air bubbles were carefully removed from the layers by exercising pressure on the top. The lid of the blotting machine was closed and weighed down with a large, heavy bottle in order to achieve a nice transfer of the proteins. Blotting was done at 11V for 1h.

Washing and Staining of the Blot

For washing a 1x TBS-T Buffer needed to be prepared. For this the 10x stock buffer was diluted using distilled water and 0.05% Tween20 was added.
After blotting the membrane was separated from the stack of whatman papers and the ladder was marked with a ball pen. The upper right edge of the blot was edged in order to not lose the orientation. Now the membrane was soaked in distilled water for 2-3 minutes. Afterwards the blot was blocked with 3% milk in TBS-T for 30min and then washed twice using 10ml TBS-T. The specified first antibody was diluted in 3% milk as described under section 3.18. The blot was incubated with the antibody over night at 4°C. Following this step the blot was washed three times with TBS-T. Now the secondary antibody was diluted in 3% milk (see section 3.18) and incubated with the membrane for 1h at room temperature. Now the blot was washed again 3 times with TBS-T. In order to detect the proteins, the kit Amersham™ECL™Prime Western Blotting Detection Reagent was used according to manufacturer's instructions. The signal was recorded at the Fusion Pulse with the program Evolution Capture. The pictures were cut and labelled in Libre office draw.

Stripping and Second Staining of the Blot

This step was used, if the same membrane had to be stained for another protein. For this the first staining had to be removed.
First the blot was washed with TBS-T for 5 minutes at room temperature. The stripping solution was diluted 1:10 using distilled water and 10ml were used on the blot for 15min at room temperature in order to strip the blot. Again the blot was washed with TBS-T for 5min, after which the procedure described under section 3.2.5 was performed again to stain the blot with another antibody combination.

Quantification

Quantification of the intensity of Bands in the Western Blots were quantified using the program EvolutionCapt Pulse 6. All signals were normalized to the actin signal. The resulting values were plotted in R, the script can be found in section 8.

3.2.6 RNA Isolation

RNA was isolated from cells using the RNeasy Mini Kit from Qiagen or with the TRIZOL reagent, as described by the user manuals.

3.2.7 Reverse Transcription PCR (RT-PCR)

In order to quantify mRNA transcripts, the RNA was transcribed into cDNA using the following protocol. If not otherwise stated 1μg of total RNA was used with random decamers, as primers, dNTP's and the MMLV reverse trancriptase in a 20μl reaction volume. This was done in the following setup:

1μg	RNA
1μl	Random Decamers
1μl	dNTP's (10mM)
ad 15μl	H_2O_{GIBCO}

This wa incubated for 5min at 65°C, afterwards the samples were cooled on ice

Add 4μl of 5x MMLV Reaction Buffer and incubate 2min at 42°C

Add 1μl MMLV Reverse Transcriptase, while the samples were in the cycler, followed by an incubation for 50min at 42°C followed by 15min at 70°C

The resulting cDNA was quantified using qPCR (section 3.2.8) and was stored at -20°C.

3.2.8 Quantitative Real Time PCR (qPCR)

With this procedure mRNA transcripts were quantified either in absolute copy numbers or in relative amounts by using additional normalizing genes. Here the amplification by PCR was done with a SYBR green dye, which binds to double stranded DNA. This complex emits light, and the intensity correlates to the total amount of DNA produced during the reaction. The PCR was done using the Quantifast SYBR green Kit in the following setup:

5μl	Quantifast SYBR green
2μl	H_2O
1μl	sense Primer
1μl	anti-sense Primer
2μl	cDNA (diluted 1:10 in H_2O)

The reaction was either performed in 96 well format on the Realplex Mastercycler epGradient S or in 384 well format at the LightCycler480. In both cases the following PCR program, see Table 3.24 was run.

Table 3.24: qPCR reaction parameters

Cycles	**PCR step**	**Temperature**	**Time**
	Initial melting	95°C	5min
45 cycles	Melting	95°C	8s
	Annealing and extension	60°C	20s
Final cycle	Melting	95°C	15s
	Annealing and extension	60°C	15s
	Melting curve		10-20min
		95°C	15s

All samples and standards were measured in replicates. The standard curve dilutions (1:10, 1:50, 1:100, 1:1000), made with a positive control sample, were measured in addition to the unknown samples, and used later on for quantification. With these a standard curve for each gene was generated, from which the amplification efficiency and was calculated. Using the slope and the intercept the software calculated the amount of cDNA in the unknown samples. Specific amplification was controlled for in the melting-curve analysis and uncertain samples were repeated. The data was exported from the software and analysed using Excel.

Relative Quantification

The amount of cDNA calculated by the realplex software was used for the relative quantification. The concentration calculated for was normalized to the level of GAPDH in the same samples.

Delta Delta CT Quantification

The amplification efficiency of each primer pair was calculated from the slope of the standard curve with the following equation:

$$A = 10^{(\text{-}1/\text{s})}$$

Where s stands for the slope of the respective standard curve.
In the first step the ct value for the control gene, in this case GAPDH (HE), was substracted from the ct value of the experimental gene (TE), this was done for the experimental condition and the control condition. Afterwards these δ ct's were subtracted to get the $\delta\delta$ CT values. Their negative values were taken to the power of the amplification efficiency A calculated above. This led to the following equation:

$$A^{-((\text{TE-HE})-(\text{TC-HC}))}$$

3.2.9 Kapa Quantification of Libraries

The kapa quantification was used for some libraries, which were sequenced on the NextSeq 550. Here the Kapa Library Quant Illumina Kit of the company Kapa Biosystems was used. The qPCR reaction was done at the Realplex system from eppendorf in 96well plates. The standards and the pcr mastermix were taken from the kit and used according to manufacturers instructions. All samples and standards were analysed in triplicates. The libraries were diluted 1:10,000 and 1:100,000 in 10mM Tris/HCl pH 8.0 with added 0.5% Tween20. 4μl of these dilutions were used in the qPCR reaction, which was done according to table 3.25.

Table 3.25: qPCR Program for Kapa Quantification of Libraries

Cycles	**PCR step**	**Temperature**	**Time**
	Initial melting	95°C	5min
35 cycles	Melting	95°C	30s
	Annealing and extension	60°C	45s
Final cycle	Melting	95°C	15s
	Annealing and extension	60°C	15s
	Melting curve		10-20min
		95°C	15s

The data was exported from the realplex software into excel and analysed there. The concentration calculated by the software with standard curve was used to calculate the nM concentration of the libraries with the following equation:

$$\text{c(Sample)} * \frac{\text{Size of DNA Standard in bp(452)}}{\text{Average fragment length of library in bp}}$$

The mean concentration for both library dilutions was used for sequencing.

3.2.10 Agarose Gel electrophoresis

This method was used to determine the size of DNA fragments. The negatively charged DNA moves through the gel, where the speed is determined by the size of the DNA. For a 1% gel 1g of Agarose was solved in 100ml 1x TAE-Buffer, while it was being heated in the microwave. The solution was cooled to 60°C before the ethidium bromide (0.5μg/ml) was added. Now the suspension was poured into a gel chamber, where it was allowed to cool down and solidify. Following which it was transferred into a gel running chamber. The comb and side bars were removed and the gel covered with 1x TAE buffer. The samples were loaded into the slots and a ladder, for the correct size estimation, was run as well. The electrodes were connected to a power station and the electrophoresis was run at 110V for 1 hour, or until the loading dye band reached the lower end of the gel. The size distribution of the samples was checked on the Typhoon.

3.2.11 Assay of Transposase Accessible Chromatin (ATAC-seq)

This method was used for the analysis of open chromatin. The presented protocol was adapted from this paper [146]. Cells were either obtained by elutriation (monocytes) or harvested as described in section 3.2.1 (macrophages).

DNAse Treatment

The cells were treated with DNAse at a concentration of 200U/ml in RPMI medium for 30min at 37°C. Following this step they were washed two times with 1x PBS in order to remove all of the enzyme. The cells were counted again and 50.000 cells per sample used in the next step.

Transposition reaction

The designated cells were centrifuged at 500g, 4° for 5 minutes, the whole supernatant was carefully removed and 50μl ATAC lysis buffer were added to each sample. After an incubation time of three minutes on ice the lysis was stopped using 1ml ATAC-RSB Buffer with added 0.1% Tween20. The tubes were inverted three times, then centrifuged at 500g, 4°C for 10min. The supernatant was carefully removed and discarded. The remaining pellet, consisting only of nuclei, was suspended in 50μl Transposition Mix (see Table 3.26).

Table 3.26: Transposition Master Mix

Reagents	**Volume for 1rxn**
2x TD Buffer	25μl
Transposase	2.5μl
1x PBS	16.5μl
Digitonin (1%)	0.5μl
Tween20 (10%)	0.5μl

The transposition reaction was performed in a heating block at 37°C, 1000rpm shaking for 30min. Afterwards DNA was purified from each reaction using the Monarch PCR Purification Kit from the company NEB, according to the instructions of the manufacturer. All samples were eluted in 21μl EB-Buffer (contained in the kit) and stored at -20°C until the PCR amplification could be performed.

PCR Amplification

In order to be able to build a library compatible with next generation sequencing on an Illumina platform, the samples were amplified using Nextera Index Primers. These contain indices in both sense and anti-sense strands. 10μl of sample was combined with 2.5μl of one i7-index primer and 2.5μl of one i5-index primer, To this mixture 35μl PCR Master Mix (see Table 3.27) was added.

Table 3.27: ATAC-PCR Master Mix

Reagents	**Volume for 1rxn**
Water	10μl
5x HF-Buffer	10μl
5M Betaine	13 μl
dNTP (10mM)	1.5μl
Phusion Polymerase	0.5μl

After mixing thoroughly the following pcr program (see table 3.28) was run.

Table 3.28: PCR Program for the amplification of ATAC samples

Number of Cycles	**Temperature**	**Time**
	72°C	5min
	98°C	30s
13x	98°C	10s
	63°C	30s
	72°C	1min
	72°C	1min
	4°C	forever

Subsequently the samples were purified using AMPure XP Beads. In the first step the DNA is bound to the Beads at a ratio of 1:1.8, therefore 90μL of Beads were added to each sample. This mixture was incubated at room temperature for 5min. The beads are magnetic, therefore the samples were placed on a magnetic rack for 3min and all of the supernatant was discarded. The beads were washed twice using 200μl of 80% ethanol (which was prepared with RNAse/DNAse free water), during these steps the cups were kept on the magnetic rack. Afterwards the beads were air dried on the magnetic rack for a maximum 10 minutes and suspended in 40μL of water in order to elute the DNA from the beads. After 5 minutes of incubation at room temperature the samples were again separated from the beads on the magnetic rack. The supernatant was transferred into a new eppendorf cup. Within the second purification step a size selection for fragments of 250 to 600bp was integrated. For this a sample to bead ratio of 1:0.55 was used, therefore 22μL beads were added to all samples and again incubated for 5 minutes at room temperature. Following this, the samples were placed on the magnetic rack for 3 minutes and the supernatant transferred to a new eppendorf cup. The beads, now bound to fragments larger than 600bp, were discarded. In order to purify the remaining DNA 72μL beads were added to all samples. They were washed as described above, then eluted in 21μL water. The libraries were quantified at the Qubit™using the dsDNA High Sensitivity Kit and their quality was assessed at the TapeStation using the D100 High Sensitivity Kit. Until pooling they were kept at -20°C.

3.2.12 Library Preparation for miRNA-seq

This method was used to profile mature miRNAs in the in vitro differentiated macrophages. These oligonucleotides are important regulators of many cellular processes. The pre-miRNA expression, which could be detected in RNA-seq data, does not immediately lead to functional miRNAs, because of a lot of processing steps in between. Therefore to conclusively determine, wether or not these RNAs play a role in the polarization of macrophages they were analysed as follows.

2μg of total RNA (isolated as described in section 3.2.6) were used for each sample.

Adapter Ligation

In the first step an adapter was ligated to the 3' ends of all RNAs. For this the master mix described in table 3.29 was prepared and incubated at 90°C for 30 seconds.

Table 3.29: 3' Adapter Ligation Master Mix

Reagent	Volume for 1 reaction
chosen RNA	x μl
10x Ligation Buffer	2μl
DMSO (100%)	3μl
H_2O	12μl - x
3'-adenylated Adapter (5pmol)	1μl

To this heated mixture 2μl of RNA-Ligase was added. After a short mix, the sample was incubated over night at 4°C. Following this step, the second adapter was ligated as follows. 18μl of the mix described in table 3.30 was added to all samples.

Table 3.30: 5' Adapter Ligation Master Mix

Reagent	Volume for 1 reaction
10x Ligation Buffer	2μl
ATP (100mM)	0.4μl
DMSO (100%)	3μl

Table 3.30: 5'-Adapter Ligation Master Mix

H_2O	11.6μl
5'-adenylated Adapter (10μM)	1μl

After mixing everything, the samples were again heated for 30 seconds at 90°C, before adding 2μl of T4-RNA Ligase to the mix. All of this was incubated or one hour at 37°C, for the reaction to take place.

cDNA Synthesis

In order to synthesize complementary DNA for all RNAs the kit SuperScript III First Strand cDNA Super Mix from Invitrogen was used. 6μl of the ligation product was combined with 1μl primer (100μM) and 1μl Annealing Buffer. This was mixed and incubated for five minutes at 65°C. The samples were cooled on ice for a minute, then 10μl 2x First Strand Mix and 2μl of SuperScript III were added. The reaction took place at 50°C for 50 minutes. The enzymes were degraded at 85°C for five minutes.

PCR Amplification

To check the correct adapter ligation and how many cycles were needed for optimal library amplification a pilot PCR was run. This PCR was done with only small amount of sample, the rest would be amplified later on, when the correct amount of cycles had been determined. For this the samples were prepared as described in table 3.31.

Table 3.31: Pilot PCR Reagent Mix

Reagents	**Volume for 1 reaction [μl]**
H_2O	6.25
5x Phusion HF Buffer	2
5' PCR Primer	0.2
Tru-seq Index Primer	0.2
dNTP mix (10mM)	0.25
Phusion Polymerase	0.1
cDNA (Sample)	1

On this mixture the PCR program described in table was run.

Table 3.32: PCR Program for miRNA-seq Libraries

Temperature	**Time**	**Step**	**Cycles**
98°C	1s	Initial Denaturing	
98°C	10s	Denature	
60°C	10s	Primer binding	20x

Table 3.32: PCR Program for miRNA-seq Libraries

72°C	30s	Elongation
72°C	10s	Final Elongation
4°C		Storage

Samples were mixed with loading dye and analysed on a 6% PAA gel, which was made with the SequaGel components by the manufacturers instructions. For the size estimation the O'Range Ruler 20bp ladder was run as well. The miRNA band should run at 140bp, from the thickness of this band, the amount of cycles needed per sample was judged, see table 3.33.

Table 3.33: miRNA-seq Sample Barcoding and PCR Amplification

Sample	**Truseq Index Primer**	**Sequence**	**PCR Cycles**
exp95 Monocytes	Index 1	ATCACG	16
exp95 AB Teflon	Index 2	CGATGT	16
exp95 NaCl	Index 3	TTAGGC	17
exp95 TGF-β	Index 4	TGACCA	16
exp95 Adiponectin	Index 5	ACAGTG	16
exp95 Serotonin	Index 6	GCCAAT	16
exp95 Hemin	Index 7	CAGATC	16
exp95 IgG	Index 8	ACTTGA	16
exp95 AB-TcPlastic	Index 9	GATCAG	16
exp95 MCSF	Index 10	TAGCTT	17
exp95 GMCSF	Index 11	GGCTAC	16
exp95 iDC	Index 12	CTTGTA	16
exp95 Collage I	Index 13	AGTCAA	17
exp95 Collagen IV	Index 14	AGTTCC	16
exp95 Fibronectin	Index 15	ATGTCA	17
exp95 Laminin	Index 16	CCGTCC	17
exp95 Rolling Bottles	Index 17	GTAGAG	18
exp98 Monocytes	Index 18	GTCCGC	17
exp98 AB-Teflon	Index 19	GTGAAA	18
exp98 NaCl	Index 20	GTGGCC	18
exp98 TGF-β	Index 21	GTTTCG	18
exp98 Adiponectin	Index 22	CGTACG	17
exp98 Serotonin	Index 23	GAGTGG	18
exp98 Hemin	Index 24	GGTAGC	17
expp98 IgG	Index 25	ACTGAT	18
exp98 AB-TcPlastic	Index 26	ATGAGC	17
exp98 MCSF	Index 27	ATTCCT	18
exp98 GMCSCF	Index 28	CAAAAG	17

Table 3.33: miRNA-seq Sample Barcoding and PCR Amplification

exp98 iDC	Index 29	CAACTA	18
exp98 Collagen I	Index 30	CACCGG	18
exp98 Collagen IV	Index 31	CACGAT	18
exp98 Fibronectin	Index 32	CACTCA	18
exp98 Laminin	Index 33	CAGGCG	18
exp98 Rolling bottles	Index 34	CATGGC	18

For the final amplification of the libraries, 10μl of sample was mixed with the master mix described in table 3.34. The PCR program described in table 3.32 was run with the amount of cycles specified in table 3.33.

Table 3.34: miRNA-seq PCR Mastermix

Reagent	**Volume** [μl]
H_2O	26.25
5x Phusion HF Buffer	10
5' PCR Primer	1
Tru-seq Index Primer (see table 3.33)	1
dNTP mix (10mM)	1.25
Phusion Polymerase	0.5

Purification of Library

After amplification the whole sample was loaded onto another 6& PAA Gel (SequaGel), run with 1x TBE buffer at 300V and stained with ethidium bromide. The miRNA-adapter fragment ran at 140bp size, the band was excised and transferred to prepared eppendorf cups. For this 0.5ml cups were pierced at the bottom and put inside a 1.5ml eppendorf cup. The gel pieces were shredded by centrifugation at 17000g for 2min, until the gel pieces were found in the 1.5ml tube. This pieces were incubated with 300μl Elution Buffer (see table 3.15) over night at 25°C while shaking at 800rpm.

A blue tip (1000μl) was cut at the end to widen the opening, then the suspension of gel pieces and buffer was pipetted onto a SpinX Centrifuge Filter. To separate eluate from the gel the samples were centrifuged at 14000g for 2 minutes. The flow-through was transferred into a DNA-low binding tube (1.5ml) and the DNA was precipitated using 2.5 volume ethanol (100%). For this step the samples were kept over night at -80°C. Following this incubation the samples were centrifuged at 17000g, 4°C for 30 minutes. The supernatant was removed and the samples centrifuged again for 30 seconds. The DNA pellet was air dried and then solved in 15μl water.

The quality of these libraries was checked at the TapeStation using the D1000 high sensitivity kit. The concentration was measured at the Qubit, using the dsDNA high sensitivity kit. All samples were stored at -20° until sequencing.

3.2.13 Chromatin Immunoprecipitation coupled with Next generation sequencing (ChIP-seq)

Chromatin Immunoprecipitation in combination with next generation sequencing was mainly used to profile transcription factor binding or histone modifications in the search for active genomic regions. In the first step proteins had to be cross-linked to DNA in order to conserve the binding throughout the procedure. This was either done only with formaldehyde or the combination of formaldehyde and disuccinimidyl glutarate.

Formaldehyde fixation

The whole procedure was performed under the fume hood.
For 10 imunoprecipitations 20 million cells were harvested and centrifuged for 10 minutes at 300g, 4°C. The Pellet was taken up in 20ml RPMI medium containing 10% FCS, which was warmed to 37°C beforehand. For the fixation formaldehyde was added to a final concentration of 1% (1.33ml) and the suspension incubated at RT for 10 minutes. To stop the reaction 0.125M glycine was added and the sample slowly rotated. Afterwards the cells were centrifuged again at 300g, 4°C for 10 minutes. The supernatant was discarded and the pellet washed twice with 1x PBS with added 10mM NaBut and 1mM PMSF (centrifugation at 300g, 4°C for 8 minutes). Following this step the cells were suspended in 1ml PBS 10mM NaBut, 1mM PMSF, and the suspension transferred into a 2ml eppendorf cup. After another centrifugation the supernatant was discarded and the pellet frozen at -80°C.

Disuccinimidyl Glutarate (DSG) and Formaldehyde Fixation

In order to fixate unstably bound proteins to the chromatin, it was fixated using two reagents. First with Disuccinimide glutarate then with formaldehyde. This was done for the ChIP experiments on Dec-2. The DSG solution had to be freshly prepared (see table 3.35).

Table 3.35: DSG Solution

Reagents	**Volume needed**
DSG	50mg
DMSO	300μl
PBS	76.3ml

20 million cells were solved in 20ml DSG solution and incubated for 30 minutes at room temperature. Afterwards 1.33ml formaldehyde (end concentration 1%) was added and the suspension again incubated for 10 minutes at room temperature. To stop the reaction 1ml glycine solution (2.625M, final concentration 0.125M) was added and the liquid slowly rotated three times. The cells were pelleted at 1000g, 4°C for five minutes, then washed twice using PBS (with 1mM PMSF and 10mM NaBut included). The pellet was stored at -80°C.

Sonification of fixated chromatin

In order to separate nuclei from the cytoplasm, in the first step only the cell plasma membrane was lysed, while the nuclei stay intact. For this the pellet was taken up in L1A (500μL for 20 million cells), freshly prepared as described in table 3.36. Now 500μL of L1B, freshly prepared see table 3.37, was added to start the lysis.

Table 3.36: Lysis Buffers for ChIP

L1A	**Volume for 1ml**
CBM	900μl
Water	100μl

Table 3.37: Lysis Buffers for ChIP

L1B	**Volume for 1ml**
CBM	900μl
NP-40	100μl

Samples were briefly mixed and incubated on ice for 10 minutes, following which the nuclei were pelleted at 700g for 5 minutes. This pellet was suspended in nuclear lysis buffer L2 (see table 3.38, 800μl for 20 million cells), which was kept at RT, in order to now lyse the nuclei.

Table 3.38: L2 with added inhibitors

Reagents	**Volume for 1ml**
L2	960μl
PMSF	10μl
NaBut	10μl
50x Protease Inhibitor	20μl

The chromatin was sheared using the Covaris device with the following settings see table 3.39.

Table 3.39: Covaris Settings

Option	**Settings**
Duty Cycle	20%
Intensity	8
Cycles per Burst	200
Water level	15
Time	40min
Volume	300μl - 3ml

In order to clear the lysate, the solution was centrifuged again at 13000rpm (Biofuge), 4°C for 5min and the supernatant transferred into a new eppendorf cup. 20μl were saved to check the fragmentation on an agarose gel, 5% (4μl) were put aside and later on used as input control. The remaining chromatin was stored at -80°C.

Checking the Fragmentation

In order to analyse the fragmentation pattern produced in the last step, the saved sample was incubated over night at 65°C with 2μg/μl Proteinase K (20μg/μl stock). The next day the sample was shortly centrifuged and 1μl RNase (10μg/μl) was added, followed by an incubation for another 30 minutes at 37°C. The DNA was purified using the Monarch PCR cleanup kit according to manufacturers instructions. The DNA was eluted in 20μl EB Buffer (included in the kit). The concentration was measured at the NanoDrop and 700ng were combined with 6x DNA laoding dye and loaded onto a 2% agarose gel (containing ethidiumbromide). To estimate the size distribution a 50bp ladder was run as well. The gel was analysed on the Typhoon, the sonification was found to be successful if there was a strong smear from around 150 to 400bp.

Immunoprecipitation

In order to remove all un-specifically binding fragments, the lysate was treated with sepharose beads. 50μl of Sepharose CL-4B beads were washed twice with TE-Buffer, for this they were centrifuged at 3000rpm (Biofuge), 4°C for 1min and the supernatant was discarded. Following these steps, the beads were suspended in Dilution Buffer (DB), to match the original volume. Binding sites on the beads were blocked with 25μl BSA (20%) and 4μl Glycogen per 1ml beads. This mixture was rotated at RT for 2h. Subsequently 50μl blocked beads were combined with 80μl lysate (which was equal to 2 million cells) and 120μl DB (with added PMSF, NaBut and protease inhibitors). This mixture was rotated at 4°C for 2h. Subsequently the sample was centrifuged at 13000rpm (Biofuge), 4°C for 5min and 200μl precleared lysate were pipetted into a clean eppendorf cup. To this 2.5μg of the selected antibody was added (see section 3.19) and everything revolved over night at 4°C.

In order to be able to pull down antibody-chromatin complexes, sepharose beads coated with either protein A or G (see table 3.40), were again washed with TE-Buffer and blocked with Glycogen and BSA as described above for the Sepharose CL-4B beads. This blocking step was done over night at 4°C.

Table 3.40: Selection of Sepharose Beads for different types of Antibodies

Species	**Subclass**	**Protein G binding**	**Protein A binding**
Human	IgG$_1$	++++	++++
Mouse	IgG$_1$	++++	+
	IgG$_{2b}$	++++	++++
Rabbit		+++	++++

Subsequently 50μl of blocked Protein G or A coated beads were added to the lysate-antibody mix and rotated for 2-3 hours at 4°C. In the meantime all needed washing buffers and the TE buffer were cooled on ice. After the antibody-chromatin-complex bound to the beads, they were thoroughly washed to remove all unspecific binding. For this they were centrifuged at 500g, 4°C for 5min. Then the supernatant was discarded and the beads suspended in 400μl Wash Buffer 1. The suspension was rotated for five minutes at room temperature, centrifuged at 500g, 4°C for two minutes and the supernatant discarded. Like this the beads were washed again with wash buffer 1, two times with wash buffer 2, two times with wash buffer 3 and three times with TE buffer. During this time the elution buffer was freshly prepared according to section 3.1.5. Once the washing was done, the beads were suspended in 110μl EB buffer and rotated for 20 minutes at room temperature and 1 rpm. Afterwards the beads were centrifuged again, as described above, and 100μl of supernatant was saved in a new eppendorf cup. The Beads were incubated with 100μl fresh EB buffer, this time rotated at room temperature for 10min. After centrifugation, 100μl of supernatant were saved and combined with the previous 100μl. From this step onwards, the input samples were treated the same as the ChIP samples, therefore they were thawed from -80°C and 200μl of EB buffer was added. To all samples 10μl of 5M NaCl was pipetted, following which they were incubated over night at 65°C for reverse crosslinking.

Purification of Chromatin

After this incubation, the remaining RNA and proteins in the samples were degraded as follows. First 7μl of RNase (Qiagen) were added to all samples and they were incubated at 37°C for 1h. Subsequently 5μl Proteinase K were added and everything was incubated another hour at 37°C, in order to remove all remaining proteins. Then the DNA was purified using the Monarch PCR cleanup kit according to manufacturers instructions. The samples were eluted in 50μl EB buffer. 10μl were used to check the enrichment using qRT-PCR and 40μl were used for Library preparation.

Library Preparation

In these steps the DNA fragments selected during the immunoprecipitation are ligated into contructs that enable sequencing on an illumina platform. For this the kit NEBNext® Ultra II DNA Library Prep Kit for Illumina® was used according to instructions from the manufacturer.

3.2.14 RNA-seq

RNA-seq was used to globally quantify gene expression in all macrophage populations.
In order to obtain high quality libraries only RNAs with a RIN value higher than 9 were used. For the preparation either the TruSeq Stranded Total RNA Library Prep Human/-Mouse/Rat (48 Samples) or the ScriptSeq v2 RNA-Seq Library Preparation Kit, was used according to the instructions provided by the manufacturer. The resulting libraries were quantified using the dsDNA High Sensitivity Kit at the Qubit™and at the TapeStation. Some libraries were also quantified using the Kappa Quantification (see section 3.2.9).

3.3 Bioinformatic Analysis

3.3.1 Programs

Table 3.41: List of Programs

Program	Version
Adobe Illustrator CS3	v13
BD FACSDiva	V8.0.1
bedGraphToBigWig	v4
Bedtools	2.27.1
Bowtie2	2.3.4
Cluster3.0	1.54
EvolutionCapt Pulse 6	17.03
FastQC	v0.11.7
HOMER[148]	v4.9
IGV genome Browser	2.4.6
IGVTools	2.3.98
Illumina Experiment Manager	v1.15
JAVA TreeView	1.1.6r4
Libre Office	5.4.3.2 (x64)
Microsoft Office®	2013
Miktex	2.9
miRDeep2	v2.0.0.8
Picard-tools	1.138
R	3.4.3
Rstudio	1.0.143
sambamba	0.6.7
samtools	1.6
skewer	0.2.2
STAR	v.2.5.3a
Texstudio	2.10.8

3.3.2 R packages

Table 3.42: List of R packages

Packages	Version
EdgeR[149, 150]	3.20.1
flowCore[151]	1.44.0
flowQ[152]	1.38.0
flowUtils[153]	1.42.0
flowViz[154]	1.42.0

Table 3.42: List of R packages

GGally[155]	1.3.2
ggcyto[156]	1.6.1
ggplot2[157]	2.2.1
gplots[158]	3.0.1
RColorBrewer[159]	1.1-2
sqldf[160]	0.4-11
Rtsne[161]	0.13
ggpubr[162]	0.1.6
ggrepel[163]	0.7.0

3.3.3 Databases

Table 3.43: List of Databases

Database	Link
DIANA tools	http://diana.imis.athena-innovation.gr/DianaTools
Metascape	http://metascape.org/gp/index.html
GeneCards	www.genecards.org

3.3.4 ATAC-seq Analysis

All three samples of isolated Hofbauer cells (see section 3.2.2) were sequenced in Vienna at the Biomedical Sequencing Facility on a Hiseq3000/4000 in a single read 50bp lane. The data was downloaded from their server in the provided BAM format using the following command. Afterwards the md5 sums were checked, to see if the download was successful. In order to utilize them, they were converted back to the original Fastq format using the program SamToFastq provided by Picard-Tools. To check the quality of the data, FastQC was run on all of them. This program calculates a range of basic statistics, such as the GC content, Quality scores over the base pairs or looks for overrepresented sequences. Only if these quality plots showed no severe problems the reads were processed further. To do all this in one step the bash script getBSFconvertSE.sh (see section 8).

```
#Example Usage of getBSFconvertSE.sh
getBSFconvertSE.sh User Password Flowcell-ID lane Sample_Name Output_Directory
```

Remaining samples were sequenced on our NextSeq in paired end mode. The raw bcl files created by the sequencer were demultiplexed according to the provided indices to sort reads to different experiments, then the data was converted into fastq files. For this a wrapping script was written. Subsequently the Fastq files were copied to the correct rawData folder on the server. FastQC program was again used on all raw files.

```
#transform bcl to fastq
```

```
bcl2fastq_wrap.sh <run number> "-p 12"
#copy all files starting with ATAC to subfolder given in target Directory
NEXTSEQcopy.sh <run number> <target Directory> 'ATAC*'
```

The program Skewer was used to look for adapter contaminations in the reads and trimming them. Subsequently all ATAC samples were mapped to the human genome version hg38 using Bowtie2[164]. The resulting SAM file was converted to the compressed BAM format, at the same time alignments were filtered by a quality cut off of 10 using the program collection Samtools. HOMER package was used to create tag directories for each sample. Additionally a file for visualization of the data in a genome browser was created. For all this the bash script mapATAC.sh (see Section 8) was used:

```
#map ATAC samples
mapATAC.sh -g hg38 -f Read1.fastq.gz -r Read2.fastq.gz -d Output_Directory -n Sample_Name
↪  -t 12 -s NEXTERA
```

During further analysis peaks were called using the findPeaks.pl command from HOMER in two ways. In the first step a peak file with the size parameter set to 150bp were analysed. In the second run the size paramter was set to 250, for both runs the minimum distance was set to 250bp and the required fold change over the local background signal was set to 4. Afterwards only an intersection of both peak files was used as the final peak file, here again a bash script, written by Prof. Dr. Michael Rehli, was used (Section 8). In order to calculate the similarity of replicates, annotatePeaks.pl from HOMER was used to add count data from the replicates to a common peak file, at the same time the values were logarithmic transformed using the -log option, which actually performs log(value+1+rand), where rand is a randomized part of a count to add pseudocounts to low counts. The size parameter was set to 1000, which means that the script counts all tags in a 1kb region around the peaks. This file could be uploaded into R, where a scatterplot was done using ggplot2. The pearson correlation coefficient was calculated to assess the similarity.
Another approach was to define a complete Peak Set over all Samples, which was used for a t-SNE Plot. Using the getDifferentialPeaksReplicates Script contained in HOMER we looked for open chromatin regions specific for either only AB (control) or Collagen treated macrophages. These peaks were used to look for enriched transcription factor motives, histone acetylation or ChIP-seq tags. Furthermore we looked for an ATAC Footprint over selected transcription factor motives using annotatePeaks.pl and R.
The commands for the combined ATAC and ChIP-data are also written in the BasicAnalysis-ATAC-Macrophages.sh script, see section 8. The plots were created in R and assembled in the Adobe Illustrator to create the figures displayed in this thesis, additionally the font size of the plots was increased for readability and further legends were added.

3.3.5 ChIP-seq Analysis

All ChIP-seq Samples were sequenced on a single read 50bp lane of a Hiseq3000/4000 at the Biomedical Sequencing Facility (BSF) in Vienna. The data was downloaded from the server in the provided BAM format. In order to use them, they were converted back to the original Fastq format using the program SamToFastq provided by Picard-Tools, using the same command as described in section 3.3.4. The reads were mapped to the human genome hg38 using Bowtie2[164] Using samtools[165] the output sam file was converted into a bam file, with a quality cut off of 10. Using the HOMER tool suite a bigWig file for visualization within a genome browser was calculated. In order to further process the data with HOMER a tagDirectory for each sample was made. These Tag Directories contain the raw tag counts and positions, one file for each chromosome. This makes it easier for programs in HOMER to use the data. These steps were included in the bash script mapChIP.sh (see Section 8).

```
#Example Usage of mapChIP.sh
mapChIP.sh -g GRCh38 -f
↪  /misc/data/rawData/chromatin/ChIP/MOMACDC/ChIP_CK_13_MAC_SP1_Dec2.fastq.gz -d
↪  MOMACDC/Dec2 -n ChIP_CK_13_MAC_SP1_Dec2 -t 12
```

findPeaks from HOMER was used to define peaks within each sample. The size parameter was set to 200 and the style was chosen depending on the sample. For transcription factor ChIPs the style factor was used, for histone ChIPs the style histone. Furthermore for each sample the input file was used as control in the process of finding peaks, in order to eliminate as many unspecific peaks as possible.

In order to analyse the similarity of replicates, a common tag directory was made using the makeTagDirectory function with the -d parameter, which adds the tag directories specified to the new tag directory. Now again a peak file was called with this tag directory and a combined input tag directory. This peak file was annotated using annotatePeaks.pl (also from HOMER) with the read counts from both samples. Simultaneously the values were log transformed and the region size to be used was set to 1000. This file was loaded into Rstudio, where ggplot2 was used to create scatterplots between the replicates. Their similarity was assessed by calculating the pearson correlation between them. Commands are the same as described for ATAC data analysis (see code sections 3.3.4)

AnnotatePeaks.pl was also used to create histograms of the tag density of other ChIP experiments over specific peaks. For this the hist option was set to 10 and the size to 2000.

The commands for the combined ATAC and ChIP-data are also written in the BasicAnalysis-ATAC-Macrophages.sh script, see section 8. The plots were created in R and assembled in the Adobe Illustrator to create the figures displayed in this thesis, additionally the font size of the plots was increased for readability and further legends were added.

3.3.6 miRNA-seq Analysis

All samples were sequenced at the Biomedical Sequencing Facility in vienna on a Hiseq3000 /4000 using a single read 50bp run. Again the data was provided in a bam format, which was downloaded from their server using the wget command in linux. Subsequently these files had to be transformed into Fastq format, which was done using the SamToFastq command by Picard-Tools.

```
#Example of Sample download:
cd /media/disk3/ngs_raw/ClaudiaK_miRNA
wget --user=ckiesewetter --password=Password https://biomedical-sequencing.at/downloads/
↪  download/BSF_downloads/group_Michael_Rehli/ckiesewetter/
↪  BSF_0351_HKVFJBBXX/BSF_0351_HKVFJBBXX_7_miRNA_30_S27873.bam
#Transform into Fastq
/media/disk1/software/jdk1.8.0_111/bin/java -jar
↪  /media/disk1/software/picard-tools-1.138/picard.jar SamToFastq
↪  INPUT='/media/disk3/ngs_raw/ClaudiaK_miRNA/BSF_0351_HKVFJBBXX_7_miRNA_1_S27844.bam'
↪  FASTQ='/media/disk3/ngs_raw/ClaudiaK_miRNA/Fastq/miRNA_seq_M2_Mono.fastq.gz'
```

FastQC was used for an initial quality assessment. Here the sequencing showed an overall high quality in base calling. Since miRNAs are only 21bp long, all samples showed a high adapter content.

```
fastqc '/media/disk3/ngs_raw/ClaudiaK_miRNA/Fastq/miRNA_seq_M2_AB_TcPlastic.fastq.gz' -o
↪  '/media/disk4/ClaudiaK_miRNA/FastQC'
```

For further analysis the mirDeep2 tool package was used [166].
The mapper.pl script was used to align reads to the human hg38 genome. At the same time this command was used to trim adapter content and filter out too short reads.

```
#Example for mapping of miRNA-seq Data
mapper.pl /media/disk3/ngs_raw/ClaudiaK_miRNA/Fastq/miRNA_seq_M2_AB_Teflon.fastq -e -h -i
↪  -j -k TGGAATTC -l 18 -m -p /media/disk4/ClaudiaK_miRNA/Bowtie_index/
↪  GRCh38_no_alt/GCA_000001405.15_GRCh38_no_alt_analysis_set -s
↪  /media/disk4/ClaudiaK_miRNA/mapped/ miRNA_seq_M2_AB_Teflon_collapsed.fa -t
↪  /media/disk4/ClaudiaK_miRNA/mapped/ miRNA_seq_M2_AB_Teflon.arf -v
```

Option -k was used to trim adapter sequences, -p assigns the genome index used, -s directs output in fasta format and -t is used to output in arf format.
Afterwards the mapped miRNAs were quantified using the quantifier.pl command. It needs known mature (set with -m) and precursor (-p option) sequences of human miRNAs.In the first step known mature miRNAs were mapped to known precursors. In the second step input reads were mapped to the precursors. For quantification the program counts the number of reads falling into an interval 2bp upstream and 5pb downstream of the mature sequence.

```
#Example for quantifying mapped miRNAs
#set correct output directory
cd /media/disk4/ClaudiaK_miRNA/counts
#Quantify miRNAs
quantifier.pl -p /media/disk4/ClaudiaK_miRNA/mirbase/21/ hairpin_ref.fa -m
↪  /media/disk4/ClaudiaK_miRNA/mirbase/21/ mature_ref.fa -r
↪  /media/disk4/ClaudiaK_miRNA/mapped/ miRNA_seq_M2_AB_Teflon_collapsed.fa -t hsa
```

This command outputs a text file, which contains the miRNA identifiers and corresponding read counts. These file were pasted together for all samples into one read count table.

```
cd /media/disk4/ClaudiaK_miRNA/counts
paste miRNAs_expressed_all_samples_1507111624.csv miRNAs_expressed_all_samples_1507112106.csv
↪  miRNAs_expressed_all_samples_1507111648.csv miRNAs_expressed_all_samples_1507112135.csv
↪  miRNAs_expressed_all_samples_1507111672.csv miRNAs_expressed_all_samples_1507112160.csv
↪  miRNAs_expressed_all_samples_1507111697.csv miRNAs_expressed_all_samples_1507112184.csv
↪  miRNAs_expressed_all_samples_1507111728.csv miRNAs_expressed_all_samples_1507112208.csv
↪  miRNAs_expressed_all_samples_1507111755.csv miRNAs_expressed_all_samples_1507112233.csv
↪  miRNAs_expressed_all_samples_1507111787.csv miRNAs_expressed_all_samples_1507112261.csv
↪  miRNAs_expressed_all_samples_1507111823.csv miRNAs_expressed_all_samples_1507112285.csv
↪  miRNAs_expressed_all_samples_1507111851.csv miRNAs_expressed_all_samples_1507112310.csv
↪  miRNAs_expressed_all_samples_1507111880.csv miRNAs_expressed_all_samples_1507112335.csv
↪  miRNAs_expressed_all_samples_1507111907.csv miRNAs_expressed_all_samples_1507112362.csv
↪  miRNAs_expressed_all_samples_1507111940.csv miRNAs_expressed_all_samples_1507112389.csv
↪  miRNAs_expressed_all_samples_1507111967.csv miRNAs_expressed_all_samples_1507112414.csv
↪  miRNAs_expressed_all_samples_1507111998.csv miRNAs_expressed_all_samples_1507112435.csv
↪  miRNAs_expressed_all_samples_1507112023.csv miRNAs_expressed_all_samples_1507112459.csv
↪  miRNAs_expressed_all_samples_1507112052.csv miRNAs_expressed_all_samples_1507112483.csv
↪  miRNAs_expressed_all_samples_1507112081.csv miRNAs_expressed_all_samples_1507112511.csv >
↪  miRNAs_expression_all_samples.txt
```

This file was opened with Excel to change header of the sample columns and reduce the ID to one column. Afterwards it was used in Rstudio to look for differentially expressed miRNAs with EdgeR. This analysis follows the same steps as described for the RNA-seq data (see Section 3.3.7). In this dataset we perceived a strong donor dependent batch effect, which was removed using the removeBatchEffect function, or was included in the generalized linear model for differential miRNA expression analysis.

3.3.7 RNA-seq Analysis

The libraries prepared as described under section 3.2.14 were sequenced in one of three locations/ sequencers as described in the following table 3.44.

Table 3.44: Sequencing of RNA-seq Samples

Sample Name	Library	Facility	Sequencer	Sequencing
HC-6	Scriptseq	BSF Vienna	Hiseq3000/4000	Paired End 75bp
HC-7	Scriptseq	BSF Vienna	Hiseq3000/4000	Paired End 75bp
HC-8	Scriptseq	BSF Vienna	Hiseq3000/4000	Paired End 75bp
M1 Monocytes	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 AB-Teflon	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 NaCl	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 TGF-β	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 Adiponectin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 Serotonin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp

Table 3.44: Sequencing of RNA-seq Samples

M1 Hermin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 IgG	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 AB-TcPlastic	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 MCSF	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 GMCSF	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 iDC	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 Collagen Type I	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 Collagen Type IV	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 Fibronectin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 Laminin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M1 Rolling Bottles	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 Monocytes	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 AB-Teflon	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 NaCl	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 TGF-β	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 Adiponectin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 Serotonin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 Hermin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 IgG	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 AB-TcPlastic	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 MCSF	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 GMCSF	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 iDC	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 Collagen Type I	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 Collagen Type IV	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 Fibronectin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 Laminin	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M3 Rolling Bottles	TruSeq	BSF Vienna	Hiseq3000/4000	Single Read 50bp
M5 Monocytes	Scriptseq	NA	NextSeq550	Single Read 86bp
M5 AB-TcPlastic	Scriptseq	NA	NextSeq550	Single Read 86bp
M5 Collagen Type I	Scriptseq	NA	NextSeq550	Single Read 86bp
M5 Collagen Type IV	Scriptseq	NA	NextSeq550	Single Read 86bp

Table 3.44: Sequencing of RNA-seq Samples

M5 Fibronectin	Scriptseq	NA	NextSeq550	Single Read 86bp
M5 Laminin	Scriptseq	NA	NextSeq550	Single Read 86bp
M6 Monocytes	Scriptseq	NA	NextSeq550	Single Read 86bp
M6 AB-TcPlastic	Scriptseq	NA	NextSeq550	Single Read 86bp
M6 Collagen Type I	Scriptseq	NA	NextSeq550	Single Read 86bp
M6 Collagen Type IV	Scriptseq	NA	NextSeq550	Single Read 86bp
M6 Fibronectin	Scriptseq	NA	NextSeq550	Single Read 86bp
M6 Laminin	Scriptseq	NA	NextSeq550	Single Read 86bp
exp115 Monocytes	Scriptseq	NA	Nextseq550	Paired end 43bp
exp115 7d noT	Scriptseq	NA	Nextseq550	Paired end 43bp
exp115 7d CTRL	Scriptseq	NA	Nextseq550	Paired end 43bp
exp115 7d 684	Scriptseq	NA	Nextseq550	Paired end 43bp
exp115 7d 1793	Scriptseq	NA	Nextseq550	Paired end 43bp
exp116 Monocytes	Scriptseq	NA	Nextseq550	Paired end 43bp
exp116 7d noT	Scriptseq	NA	Nextseq550	Paired end 43bp
exp116 7d CTRL	Scriptseq	NA	Nextseq550	Paired end 43bp
exp116 7d 684	Scriptseq	NA	Nextseq550	Paired end 43bp
exp116 7d 1793	Scriptseq	NA	Nextseq550	Paired end 43bp
exp118 Monocytes	Scriptseq	NA	Nextseq550	Paired end 43bp
exp118 7d noT	Scriptseq	NA	Nextseq550	Paired end 43bp
exp118 7d CTRL	Scriptseq	NA	Nextseq550	Paired end 43bp
exp118 7d 684	Scriptseq	NA	Nextseq550	Paired end 43bp
exp118 7d 1793	Scriptseq	NA	Nextseq550	Paired end 43bp
exp113 Monocytes	Scriptseq	NA	Nextseq550	Paired end 43bp
exp113 7d noT	Scriptseq	NA	Nextseq550	Paired end 43bp
exp113 7d CTRL	Scriptseq	NA	Nextseq550	Paired end 43bp
exp113 7d 684	Scriptseq	NA	Nextseq550	Paired end 43bp
exp113 7d 1793	Scriptseq	NA	Nextseq550	Paired end 43bp
exp129 Monocytes	Scriptseq	NA	Nextseq550	Single Read 86bp
exp129 7d Mock	Scriptseq	NA	Nextseq550	Single Read 86bp
exp129 7d CTRL	Scriptseq	NA	Nextseq550	Single Read 86bp
exp129 7d 1132	Scriptseq	NA	Nextseq550	Single Read 86bp
exp129 7d 2665	Scriptseq	NA	Nextseq550	Single Read 86bp

Table 3.44: Sequencing of RNA-seq Samples

exp130 Monocytes	Scriptseq	NA	Nextseq550	Single Read 86bp
exp130 7d Mock	Scriptseq	NA	Nextseq550	Single Read 86bp
exp130 7d CTRL	Scriptseq	NA	Nextseq550	Single Read 86bp
exp130 7d 1132	Scriptseq	NA	Nextseq550	Single Read 86bp
exp130 7d 2665	Scriptseq	NA	Nextseq550	Single Read 86bp
exp132 Monocytes	Scriptseq	NA	Nextseq550	Single Read 86bp
exp132 7d Mock	Scriptseq	NA	Nextseq550	Single Read 86bp
exp132 7d CTRL	Scriptseq	NA	Nextseq550	Single Read 86bp
exp132 7d 1132	Scriptseq	NA	Nextseq550	Single Read 86bp
exp132 7d 2665	Scriptseq	NA	Nextseq550	Single Read 86bp

After sequencing at the BSF facility in vienna, raw data was downloaded from their server to our server using the commands included in section 3.3.4 for ATAC data. The fastq files were mapped using the STAR aligner. STAR maps reads to the genome, creates a bedGraph file for visualization in a genome browser and counts how many reads fall within annotated genes. The bedGraph file was sorted and transformed into bigWig using the program bedGraphToBigWig. Optionally fastq files were trimmed to a specific length. All of these commands were combined in the bash script mapRNAseq.sh written by Prof. Dr. Michael Rehli and can be found in the appendix.

```
mapRNAseq.sh -g hg38 -l 75 -f /misc/data/rawData/RNA/MOMACDC/RNA_seq_CK_HC_6.R1.fastq.gz
↪  -r /misc/data/rawData/RNA/MOMACDC/RNA_seq_CK_HC_6.R2.fastq.gz -d MOMACDC -n
↪  RNA-seq_CK_HC_6 -t 12 -c 75
```

STAR calculates for each sample a gene count table, which contains the annotated genes from the given annotation file and the amount of reads that fall into each one. The remaining analysis was written in the M1M3M6-AllSamples-diffExpGenes.sh script. These files were pasted together using the paste command. They were combined with an index file, that contained the ensemble ID with corresponding gene symbols, gene length and gene type, this file was created by Prof. Dr. Michael Rehli. Raw gene count data was used for the analysis of differentially expressed genes in Rstudio using R and the R package EdgeR[149, 150]. The basic steps for RNA-seq analysis were done for all different cell types and experiments, these were combined in several bash scripts, which can be found in the appendix (see Section 8). In the first step the raw gene count values were imported into R and stored as a DGEList object, including information about replicates. A first filtering step is included, where genes with a count lower than 1 count per milion (cpm) in at least two samples are removed. This is necessary, to remove genes carrying no relevant information, since they are too lowly expressed. Afterwards the values were normalized for RNA composition using the trimmed mean of M-values between all pairs of samples, to calculate a scaling factor necessary to minimize log fold changes for the majority of genes and samples. Together with the library size, it gives rise to the effective library size, which was used in further analysis.

In most analysis a strong batch effect from the library preparation was seen. Therefore the function removeBatchEffect within R from the limma package was used on the logarithmic transformed values. It is useful for removing batches introduced by technical variation. We performed dimensionality reduction analysis using the t-SNE, t-Distributed Stochastic Neighbor Embedding, Plot. It performs non-linear dimensionality reduction to display multidimensional data in two dimensions, which is used to identify patterns inside the data [161, 167].To look for differentially expressed genes we used a generalized linear model [168] in combination with a quasi-likelihood F-test [168, 169] implemented in the glmQLFTest and glmQLFit functions of EdgeR. For this a design matrix was done, which included information about replicates and batch effects. Afterwards the common and gene-wise dispersions were calculated and the generalized linear model fitted to the data. Using these specific comparisons between groups of samples can be made using the quasi-likelihood F test.

Another way to look for differentially expressed genes is implemented in the anova-like analysis, also implemented in EdgeR. The comparisons between groups were specified in the same way as before, but in the fitting of the generalized linear model, not one specific comparison is chosen, but all are made in the same analysis. Edge R then adjusts the p-Values and false discovery rates to multiple testing. Afterwards the heatmap was generated in the same way as in the direct sample to sample comparison.

In order to create heatmaps of these genes, the file containing fold changes, p-values and false discovery rates (FDR) has to be combined with the corresponding expression data. Here the logarithmic transformed, batch removed values were used. Genes were selected to be differentially expressed between two conditions if the absolute logarithmic fold change was higher than two, the false discovery rate lower than 0.05 and the gene expressed at least at 1 count per million. The genes found to be up-regulated or down-regulated in the comparisons were used to look for gene ontology terms using the online database Metascape. Interesting genes were selected for Dot plots to look at the expression levels between different conditions. For this the logarithmic transformed, batch effect removed RPKM values were used. The figures were assembled using the Adobe Illustrator, in addition the font size was adjusted in most plots. Furthermore several of the legends were added in the Illustrator.

Chapter 4

Results

4.1 *In Vitro* Differentiation of Macrophages

Macrophages are highly plastic cells, which change their phenotype based on cues from their environment. Each stimulus induces a different polarization program, thus activation of these cells has been described in a spectrum model of activation [58]. The response of *in vitro* generated macrophages to various immune derived stimuli has been published recently [59]. Hence, we wanted to observe the reaction of macrophages to further stimuli, not directly related to immune responses. Therefore, we differentiated human monocytes, isolated from the peripheral blood of healthy, voluntary donors, under sixteen different conditions, which are schematically depicted in Figure 4.1. These were divided into three groups based on culture substrates, source of serum and the addition of soluble stimuli.

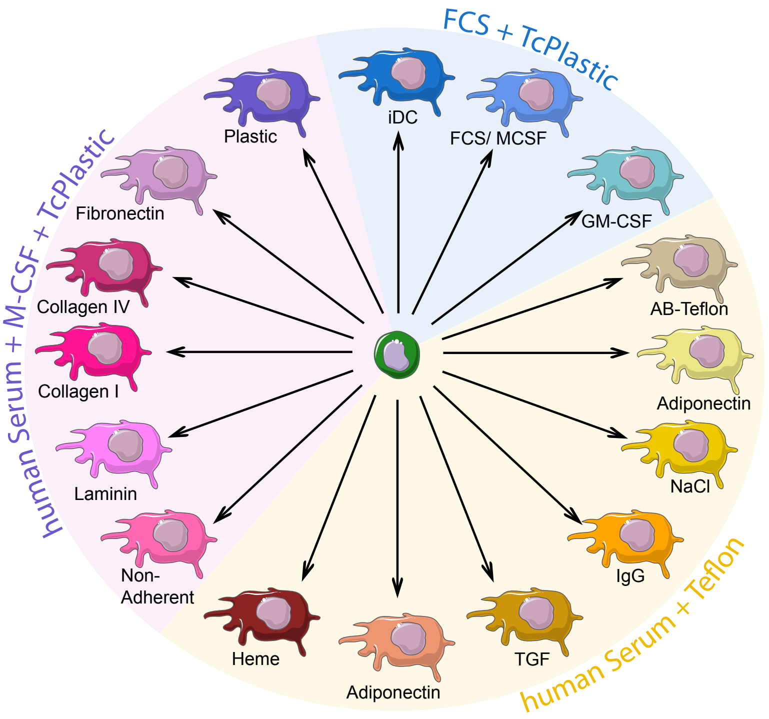

Figure 4.1: Schematic View of *in vitro* Differentiation Conditions
Schematic overview of *in vitro* differentiation conditions. These were divided into three groups, marked by different colours. Depicted in yellow is the group of soluble mediators, in green the group of standard differentiation protocols and in blue the group of macrophages cultured on diversely coated plastic dishes. All cells were cultured for seven days at 37°C, 5% CO_2 and 95% H_2O;

4.2 Surface Marker Profiling

All macrophages were profiled for classical surface marker expression using a multi-colour flow cytometry panel. We analysed CD11b, CD14, CD16, CD64, CD71, CD115, CD163 and HLADR. We observed expected changes, such as low CD14 and high HLADR levels in immature dendritic cells (iDC) and GM-CSF treated macrophages. In addition, we could show that culturing monocytes on collagen slightly increased CD16, CD64, CD115, CD163 and HLADR surface levels (Figure 4.2). Furthermore, TGF-β treatment resulted in a significant reduction of CD14 surface expression, whereas it increased the levels of HLADR Figure 4.2.

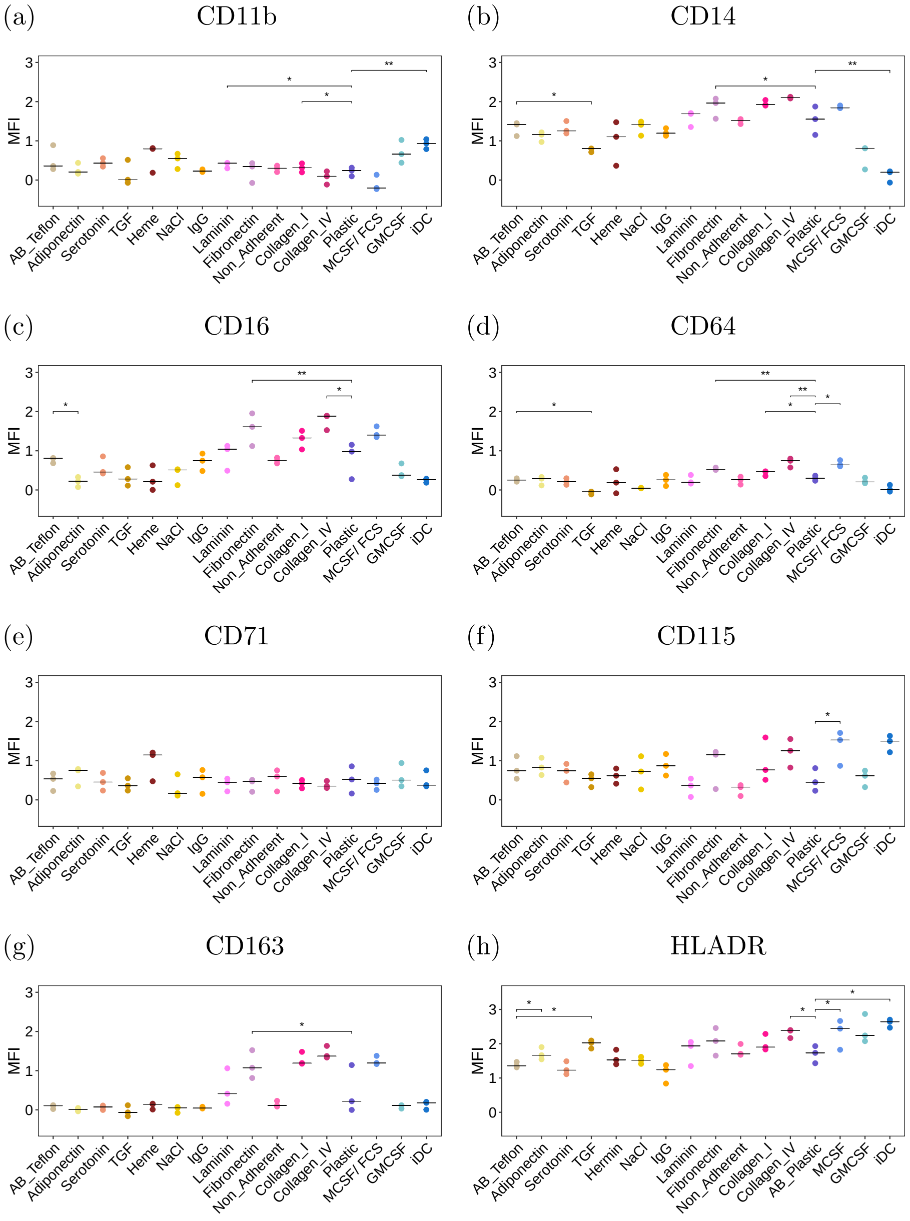

Figure 4.2: Surface Marker profiling of all *in vitro* generated Macrophages

All Dot plots display the median fluorescence intensity corrected with the respective fluorescence minus one (FMO) controls. The values were measured for each macrophage population after gating for the main population in the forward-sideward scatter, negative of 7-AAD staining and singlets. Significant differences between conditions were calculated using a paired t-test (n = 3). Only significant comparisons are shown in the dotplots.

4.3 Similarity Assessment between all Samples

In addition to surface marker profiling, we analysed global gene expression profiles using RNAseq and the open chromatin structure with ATAC. We began with investigating the transcriptome of all macrophages, for this we used the Bioconductor EdgeR package implemented in R, which was programmed for count data such as RNA-seq. We analysed the overall similarity between different macrophages with a dimensionality reduction analysis. Here, the t-SNE algorithm implemented in R was used [161, 167] and the resulting plot is displayed in Figure 4.3a. We observed a high similarity between all replicates, since they cluster close together. In addition, monocytes were clearly separated from all macrophages, whereas the latter formed several separate clusters. These largely resembled the sample classifications defined in Figure 4.1. Especially, all cells cultured in teflon bags formed a dense cluster, indicating that these stimuli did not induce strong gene expression changes. In comparison, immature dendritic cells and GM-CSF treated macrophages clustered separately. Interestingly, cells differentiated on collagen type I and IV and M-CSF/ FCS treated macrophages formed a separate cluster relative to the remaining samples, suggesting that these conditions induced a distinct macrophage polarization. To further confirm these results, we performed a Pearson correlation calculation between all samples and plotted the resulting coefficients in a heatmap (Figure 4.3b) and observed again that biological replicates showed a very high correlation ($R^2 > 0.93$). Monocytes and dendritic cells formed separate clusters, while collagen- and FCS/ MCSF treated macrophages clustered together.

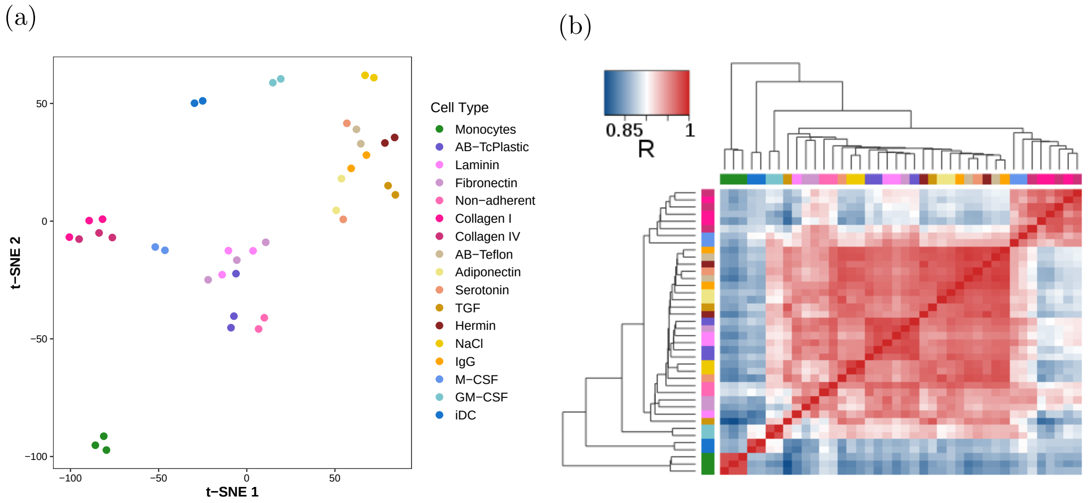

Figure 4.3: Similarity Assessment between all Samples

(a) t-SNE Plot of all RNA-seq Samples, perplexity was set to 3.5 and theta to 0. The different Samples are colour coded. (b): Pearson Correlation Coefficient Heatmap between all Samples, colour coding as in (a).

4.4 Cytokine treatment is instructive during macrophage differentiation

There are several routines in the EdgeR package for differential gene expression analysis. We chose the generalized linear model, suitable for multifactorial experiments [168], in combination with quasi-likelihood F-tests [169] for the present analysis (the complete Script can be found in Chapter 8).

An ANOVA-like test was performed to obtain genes differentially expressed between GM-CSF, immature dendritic cells (GM-CSF + IL-4) and M-CSF treated macrophages, resulting in four clusters of differentially expressed genes (Figure 4.4a). In order to assign a function to these genes, we performed a gene ontology analysis using the online tool Metascape (http://metascape.org, [170]). This program uses the given gene list and looks for overrepresented functional categories, the so called Gene Ontologies (GO). This analysis provides an insight into which cellular functions or components might be changed by a cer-

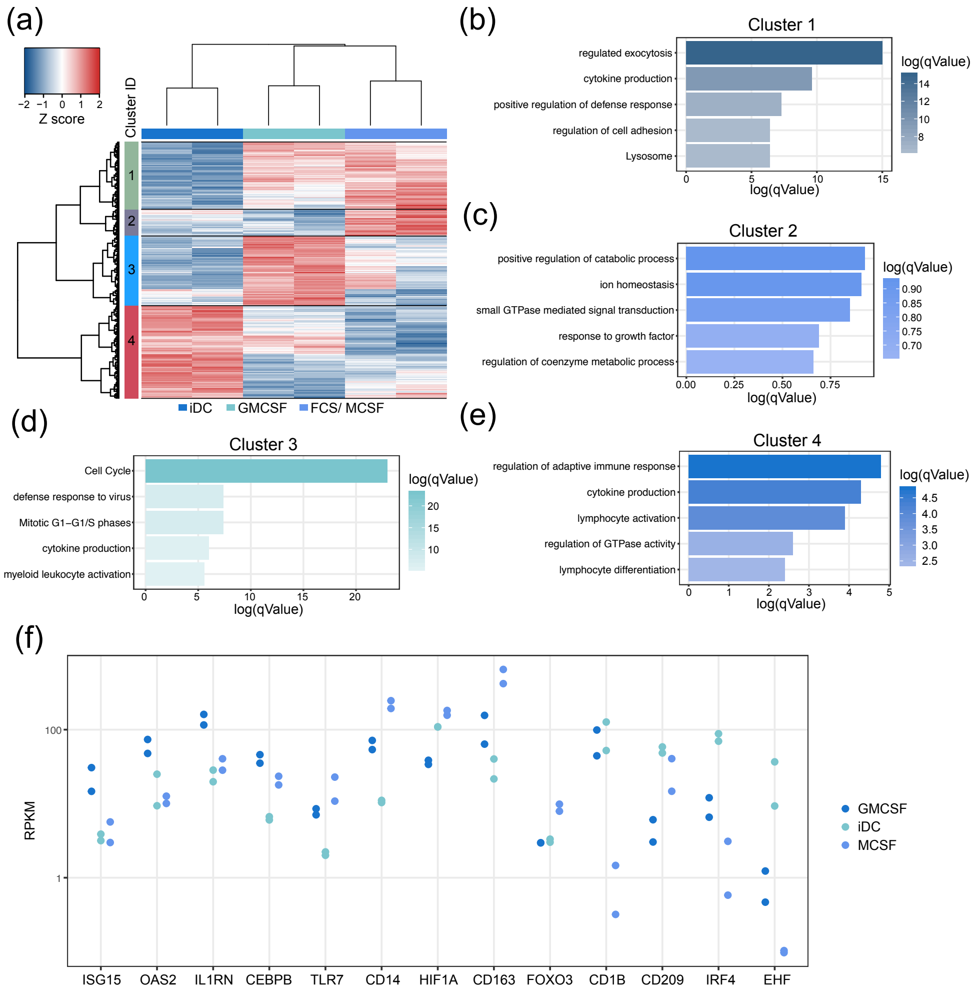

Figure 4.4: Comparison between iDC, GM-CSF and M-CSF treated cells
(a): Heatmap of anova-like analysis between iDC, GM-CSF and M-CSF/ FCS. The genes were filtered for cpm > 1 and FDR < 0.05; (b - e): Gene ontology analysis of gene clusters defined in a; Cluster 1 (b), Cluster 2 (c), Custer 3 (d) and Cluster 4 (e); (f): Dotplot of selected genes from (a), in RPKM values (n = 2).

tain treatment. Genes within the first cluster were highly expressed by both macrophages and enriched for lysosome and cell adhesion (Figure 4.4b). In contrast, genes specifically expressed in M-CSF treated cells (Cluster 2) were associated with catabolic and metabolic processes (Figure 4.4c). We confirmed the slight pro-inflammatory phenotype of GM-CSF treated macrophages [50], as genes highly expressed in these cells (Cluster 3) were enriched for gene ontology terms such as cytokine production and myeloid leukocyte activation (Figure 4.4d). Dendritic cells highly expressed genes associated with regulation of adaptive immune response and cytokine production (Cluster 4, Figure 4.4e).
As the difference between these *in vitro* generated cell types have been extensively analysed in previous studies [50, 171, 55, 172, 51, 173], we focused our attention on macrophages differentiated on collagen.

4.5 Analysis of the collagen-induced phenotype

Since the interaction of monocytes and macrophages with the extracellular matrix could be important *in vivo* for the phenotype of tissue resident macrophages and the collagen induced polarization was consistent over different protein types and coating procedures, we decided to focus the following analysis on this condition. Already during differentiation we observed differences between collagen-treated cells and control samples. The former did not adhere as strongly to the plate as control samples on tissue culture plates did. Macrophages differentiated on collagen type IV were even more non-adherent. Furthermore the cells contained many vacuoles in their cytoplasm, which we did not observe in the control samples (AB-TcPlastic) (see Figure 4.5). At first we looked at global gene expression changes using RNA-seq.

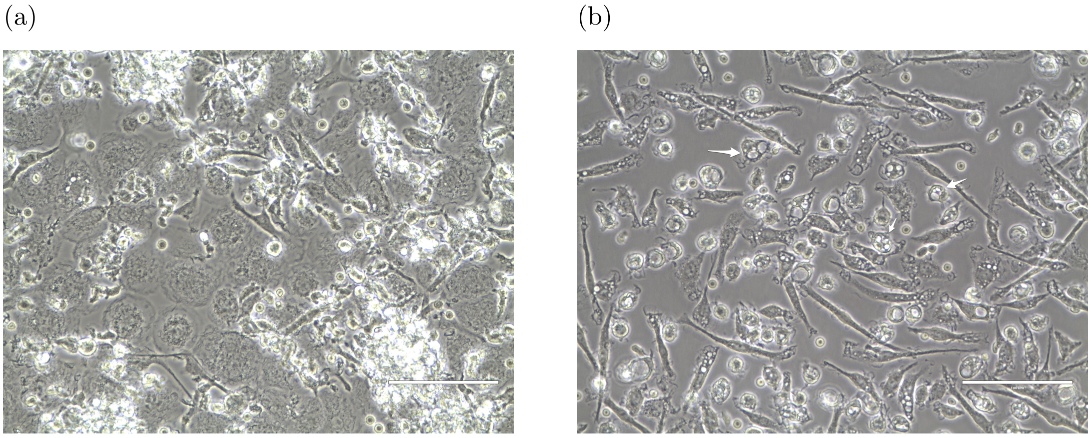

Figure 4.5: Morphology of Collagen treated Macrophages and Control cells
Microscope pictures were taken at the Evos Digital Microscope with an amplification of 40x; (a): Control macrophages (AB-TcPlastic with addiditonal M-CSF) (b) Collagen type I treated macrophages. Example cells with high amount of vacuoles are indicated by white arrows.

4.5.1 Gene Expression Analysis

In order to obtain differences in gene expression, we performed pairwise comparisons between collagen type I, type IV, laminin, fibronectin and plastic-macrophages. The genes were filtered for a logarithmic fold change (logFC) greater than 2, counts per million (cpm) greater than 1 and a false discovery rate (FDR) smaller than 0.05 (Figure 4.6a). Fibronectin treated cells show some expression changes compared to the remaining samples, but this effect was not consistent over several donors, whereas laminin-macrophages showed no differentially expressed genes compared to plastic-macrophages. In contrast, we observed 523 up- and 791 significantly down-regulated genes by seeding the monocytes on a collagen matrix. To assign a function to this list of genes, we performed a gene ontology analysis using the online tool Metascape (http://metascape.org, [170]), with default settings.

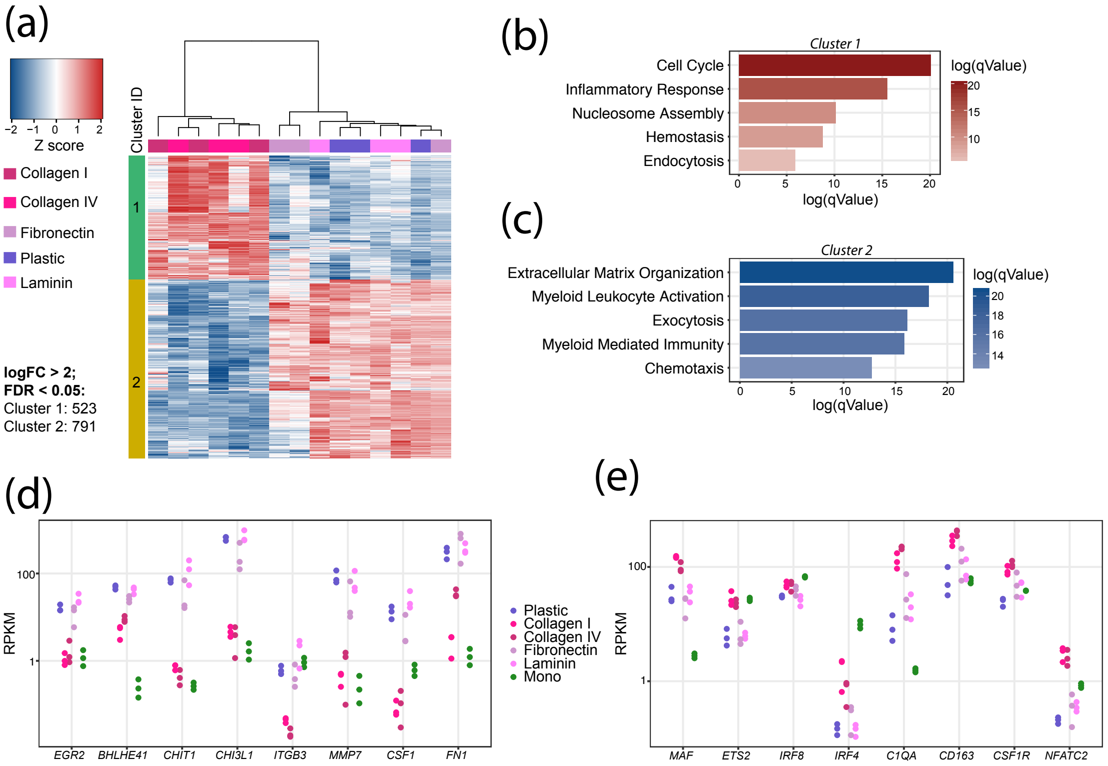

Figure 4.6: Differentially Expressed Genes between Collagen and Control samples
(a): Heatmap of differentially expressed genes after pairwise comparisons between all included samples, where the genes were filtered for a logFC > 2, cpm > 1 and FDR < 0.05; (b - c): Gene Ontology Analysis of genes in cluster 1 (b) or cluster 2 (c), performed by the online tool metascape. Five highly enriched terms were chosen and plotted in R; (d-e): Dotplot of selected genes either down-regulated (d) or up-regulated (e) by collagen treatment;

Shown in Figure 4.6b is a bar plot of significantly enriched gene ontology terms found for collagen induced genes (Cluster 1 in Figure 4.6a). Information concerning the genes described below was retrieved from the GeneCards online database (www.genecards.org, April 26, 2019), if not stated differently. One of the highest enriched terms was endocytosis, it contains several scavenger receptors (STAB1, MARCO and CD163), HFE, encoding Hemochromatosis, and the folate receptor β, encoded by FOLR2, which was thought to be specifically expressed in placental tissue, but has been found in other tissues and on activated macrophages as well [174]. Another cluster consisted of categories enriched for cell cylce proteins, including mainly histone protein one variants, for example *HIST1H3C*, but also centromere proteins (*CENPA*) and several kinesin factors (*KIF11*). The inflammatory response category contained all three genes (*C1QA, C1QB, C1QC*) required to build the first complement complex, c1q, which also forms one GO cluster. In addition, the gene ontology cluster contained *CD28, TLR7, CCL13* and *CCL24*. CD28 is usually expressed on T-cells and binds to CD80 or CD86 on antigen presenting cells, as a co-stimulatory molecule to induce complete T-cell receptor signal transduction. TLR7 is a member of the toll-like receptor family, which are important pathogen recognition receptors in innate immunity. TLR7 recognizes single stranded RNA in endosomes, and is important for the defence against viruses. In addition, we observed increased expression of another pathogen recognition receptor, *CD209* (alias DC-SIGN). It is typically induced by interleukin 4 treatment of macrophages and expressed on Hofbauer cells [175, 46]. Similarly, collagen-macrophages expressed high levels of the insulin like growth factor 1 (IGF1), which is also induced by interleukin-4 treatment, furthermore it is a potent enhancer of tissue regeneration and known to influence macrophage activation in an autocrine loop [176, 177].

The described activation of gene expression could be mediated via the induction of a transcription factor by collagen treatment. Therefore, we searched for highly expressed transcription factors. *MAF, ETS2, NFATC2* and *IRF4* were induced with a logFC greater than 2 by collagen-macrophages. Dot plots of selected genes are displayed in Figure 4.6d, e.

In addition, to the up-regulated genes, many were down-regulated by the treatment. These were found in cluster 2 of Figure 4.6a, for which we also performed a gene-ontology analysis, displayed in Figure 4.6c. The enriched Gene Ontology terms included many terms related to the extracellular matrix. These comprised several integrin genes (*ITGAE, ITGB3, ITGAX, ITGA3, ITGB8*), but also metalloproteinases (*MMP7, MMP8, MMP12, MMP19*), which are enzymes necessary for degrading extracellular matrix proteins. Besides that, genes coding for different types of collagen (*COL6A1, COL6A2, COL8A2, COL9A2, COL22A1*), laminin (*LAMB3, LAMA1*), fibronectin (*FN1*) and *TNC*, which encodes another extracellular matrix component, were found to be down-regulated. In addition, several growth factors were not expressed after culturing monocytes on a collagen matrix. These included *PDGFA*, a platelet derived growth factor, the essential growth factor for macrophages M-CSF, encoded by the *CSF1* gene and *VEGFA*, a vascular endothelial growth factor, which are all down-regulated in collagen-treated macrophages.

Regulatory exocytosis, another enriched Gene Ontology term, included two genes normally induced during macrophage differentiation, *CHIT1* and *CHI3L1* [178, 179, 180]. They encode chitinases, which are capable of cleaving proteoglycans inside microbial cell walls, thereby killing them [178].

These gene changes in gene expression could be explained by the loss of specific transcriptional activators, therefore we specifically looked for down-regulated transcription factors. We observed a significantly lower expression of *EGR2* (alias Krox20) and *BHLHE41* (alias Dec2) in collagen-macrophages compared to plastic-, laminin- and fibronectin-macrophages.

In summary, we observed that collagen induced genes were associated with cell cycle, endocytosis, complement cascade, chemotaxis and inflammation, whereas it inhibited genes concerned with extracellular matrix organization, cell migration, adhesion and vasculature development. These expression changes could be mediated by up-regulation of the transcription factors MAF, ETS2, NFATC2 and IRF4, or by the inhibition of the transcription factors EGR2 and Dec-2.

4.5.2 Profiling of Regulatory Regions

To shed light on the regulatory mechanisms behind the described expression profile we used ATAC-seq to analyse the chromatin landscape of the *in vitro* generated macrophages. This data encompasses all accessible regions, including cis-regulatory elements such as enhancers and promoters.

After mapping the samples to the human genome, we identified regions where the mapped reads pile up and form so-called peaks, indicating that this region was open at the time of analysis. A bash script containing all used commands is provided in Section 8. In order to determine the similarity of biological replicates, we plotted them against each other in several scatter plots (see Figure 4.7a-f). As shown in Figure 4.7a-f they show a Pearson correlation coefficient of 0.89 or higher, indicating that the samples are highly similar. This result was confirmed by the dimensionality reduction analysis, performed using the t-SNE algorithm. This plot, displayed in Figure 4.8a, further confirms that replicates are closely related, since they cluster close together in both dimensions. Furthermore, we observed

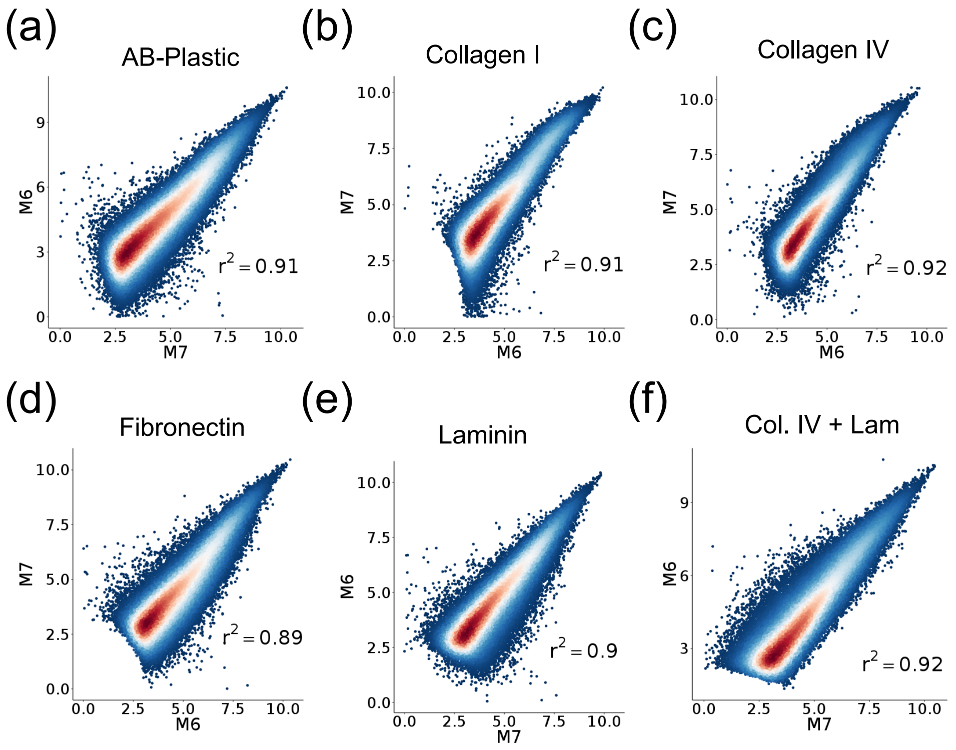

Figure 4.7: Scatter plots of ATAC Replicates

(a-f) Scatter plots between replicates of each condition. The tag densities in 1000bp around common ATAC Peaks were counted and the logarithmic transformed values plotted against each other in R using ggplot2. Pearson correlation coefficient was calculated for all samples using R and annotated to the plot. Dot colour indicates the density, blue equals low and red equals high dot density.

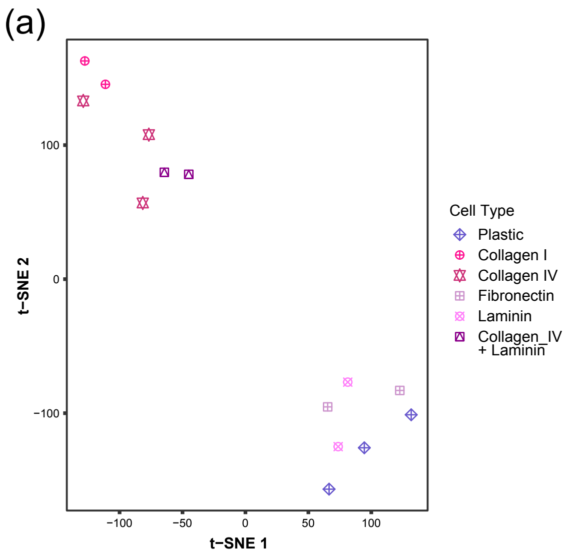

Figure 4.8: Similarity Assessment of ATAC data
(a): t-SNE Plot of ATAC Data; The tag counts were annotated to a combined peak file for all samples (n = 2).

that macrophages treated with a combination of collagen type IV and laminin clustered close to the other collagen treated samples, indicating that in our setting, collagen has a dominant effect compared to laminin treatment.

Condition specific ATAC peaks were selected using the getDifferentialPeaksReplicates.pl script contained in the HOMER suite, with default settings. From the comparison of both collagen type I and type IV against control cells we obtained 2579 peaks, whereas the reversed search returned 30296 peaks.

An example of a region containing collagen-specific ATAC peaks is displayed in Figure 4.9a. It shows an IGV Browser view of the IGF1 locus, a gene which is induced by collagen treatment and contains three collagen specific open chromatin signals. We used the program findMotifsGenome.pl to look for enriched transcription factor motifs in a 200bp region around the identified ATAC Peaks. This script works in two ways, first of all it searches for known transcription factor motifs included in the HOMER motif database, which are enriched in the given regions. Secondly, it searches for enriched DNA sequences of a given length without applying any prior knowledge and looks for matching transcription factors, this method is called a *de novo* motif enrichment analysis.

Collagen specific open chromatin revealed a distinct *de novo* motif composition, displayed in Figure 4.9b. The top 10 included an ETS, ETS/IRF composite, RUNX, NFAT and MAF motif. PU1 is capable of forming a hetero-dimer with IRF8 or IRF4, when binding the composite DNA motif known as the Ets/interferon consensus elements (EICE) [68]. We observed a strong footprint over collagen specific ATAC peaks for the composite motif, which shows that PU-1 and an IRF factor bind to it. Also plastic-macrophage specific ATAC peaks contained a strong footprint over this motif. The drop of the ATAC signal over the right side of the ETS:IRF motif was slightly smaller in plastic-macrophages compared to collagen-macrophages (Figure 4.9). These plots indicate that the motif is bound in both cell types, but at different chromatin regions. The ETS motif can be bound by

a variety of ETS factors, but on RNA level we observed an increased expression only for ETS2. Likewise, we observed an enrichment for the MAF and NFAT motifs in combination with increased mRNA expression levels for MAF and NFATC2 in collagen-macrophages (see Figure 4.6e). Furthermore, the MAF, ETS, composite ETS / IRF, NFAT and RUNT motifs showed a strong ATAC footprint in collagen-macrophage specific ATAC peaks (see Figure 4.9c - e and 4.10b,c).
To further confirm the importance of the selected ATAC peaks, we performed ChIP-seq for histone 3 lysine 27 acetylation (H2K27ac). The level of acetylation over collagen specific ATAC peaks was higher in collagen-macrophages compared to control cells differentiated on plastic (Figure 4.10d). In order to analyse if MAF is the driver behind the collagen induced phenotype, we performed ChIP-seq also for c-MAF. We observed a small overlap between collagen-specific ATAC peaks and c-MAF binding (Figure 4.10e). But plastic-specific ATAC peaks showed a similar overlap with c-MAF (see Figure 4.12f), indicating that c-MAF is active in both cell types, but at different binding sites.
In summary, we were not able to identify a specific transcription factor, which could be responsible for the observed increased gene expression levels. Probably, a combination of several factors, including cMAF, PU1/IRF and NFATC2, as well as their post-transcriptional activation could be important, but this hypothesis should be further tested.

In comparison to collagen-macrophage specific chromatin, we observed many regions specifically open in plastic cultured macrophages (30.296 ATAC peaks). Most of the genes down-regulated in collagen-macrophages were in close proximity to one or several of plastic-macrophage specific ATAC peaks (see Figure 4.12e), as shown in the CSF1 locus (Figure 4.11a). Again we used HOMER to search for enriched *de novo* transcription factor motifs within these peaks (Figure 4.11b). We observed the typical macrophage motif composition already reported for macrophage specific enhancer regions [181], including ETS, CEBPA, EGR, bZIP and bHLH motifs. Macrophages differentiated on a collagen matrix did not express EGR2 or Dec2, whose binding sequences match two of the *de novo* enriched motifs. We observed a transcription factor footprint over the ETS, CEBPA, bZIP, EGR and bHLH motif in the plastic-macrophage ATAC data (see Figure 4.11c,d and 4.12b - d). In addition, the histone acetylation (H3K27ac) level over AB-plastic specific ATAC peaks was clearly higher in plastic-treated cells compared to collagen-macrophages (Figure 4.11e). Since EGR2 and Dec2 were down-regulated on RNA-level, we performed ChIP-seq for Dec2 and used published ChIP data for EGR2 [181], in order to analyse their binding across plastic-macrophage specific open chromatin regions. As shown in Figure 4.11f) both factors show a small enrichment over plastic-macrophage specific accessible chromatin.

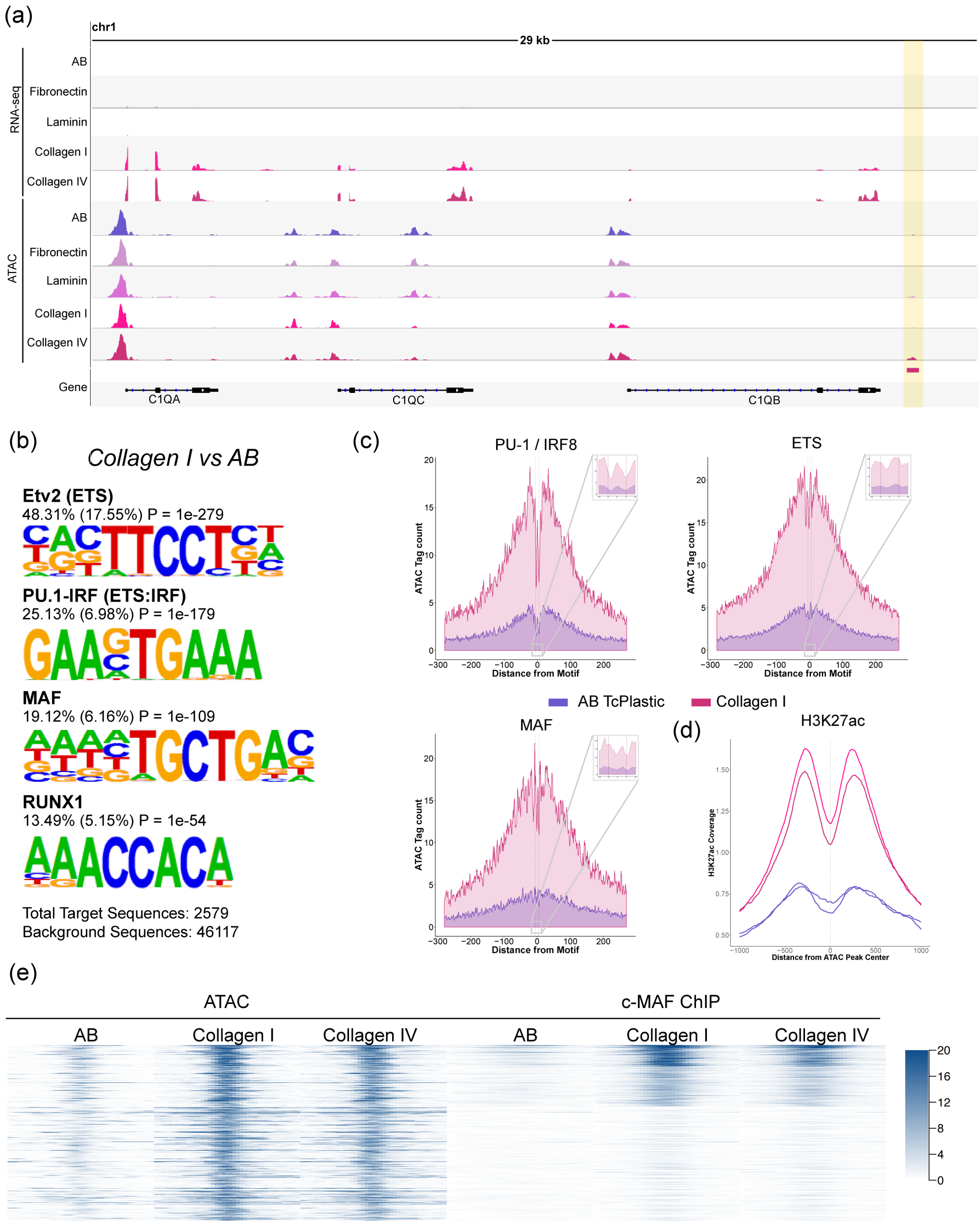

Figure 4.9: Macrophages cultured on collagen induce a specific open chromatin structure
(a): IGV Browser Shot of the IGF1 locus, where yellow bars indicate collagen specific ATAC Peaks; (b): De novo enriched transcription factor motifs, analysis was performed by HOMER. Shown in the figure are the best matching transcription factors to the respective motifs, the percentage of target and background regions containing the motif and the p-Value; (c): Histogram of ATAC data from AB and Collagen treated macrophages over Collagen specific ATAC Peaks (left) and AB specific ATAC Peaks (right) centred on the PU-1:IRF composite motif. Black lines indicate the motif position, whereas the red dotted line highlights the lowest point of the ATAC signal around the motif; (d - e): Histogram of ATAC data from Control and Collagen treated macrophages over Collagen specific Peaks, which were centred on the MAF motif (d), or on the NFAT motif (e); (f): Histogram of H3K27ac in either collagen treated or control macrophages over Collagen specific ATAC peaks. Grey area indicates the confidence interval; (g): Region Heatmap of the ATAC tag density (left) or c-MAF ChIP tag density over collagen specific ATAC Peaks, which were clustered by kmeans (k=4).

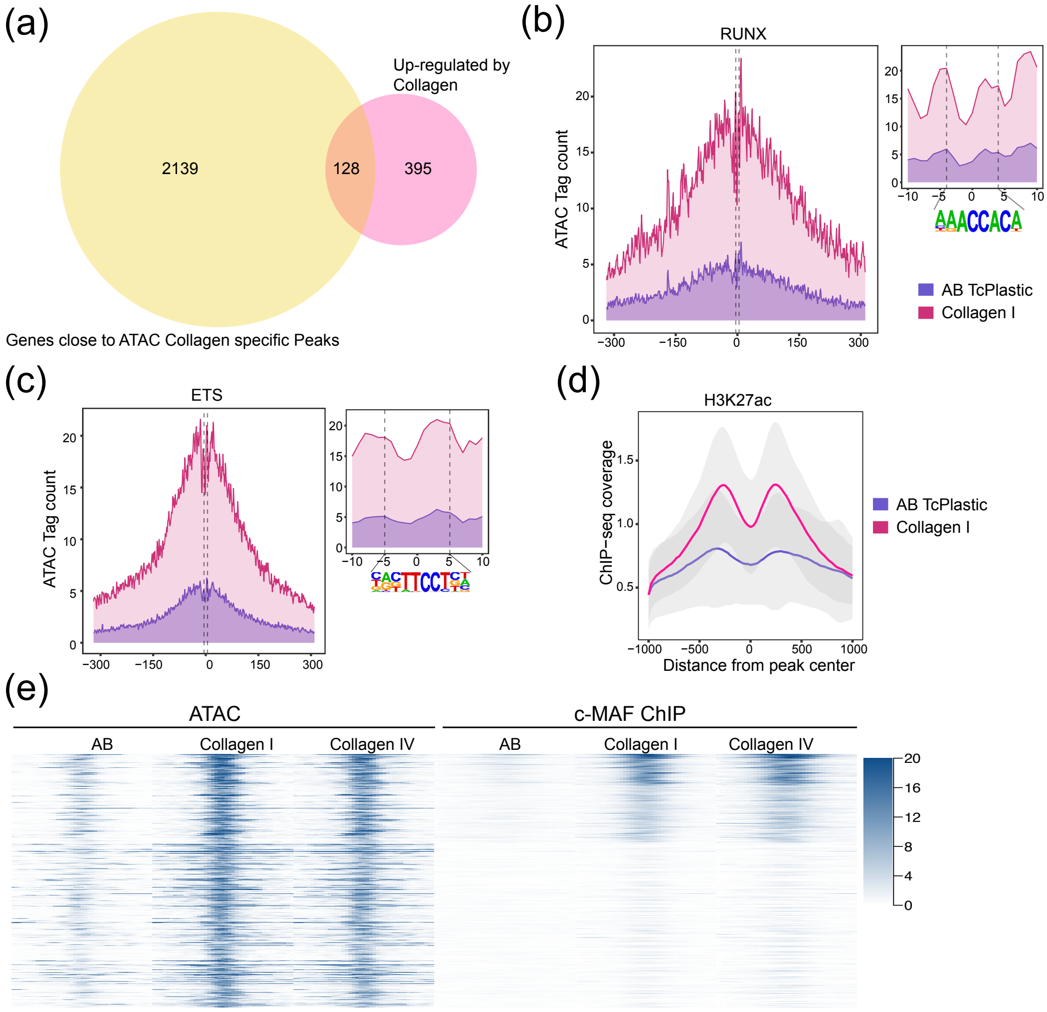

Figure 4.10: Several transcription factors could be responsible for the collagen phenotype
(a): Venn Diagram of genes up-regulated by collagen induced macrophages and genes situated closely to ATAC Peaks selectively open in macrophages cultured on collagen; (b-c): Histogram of ATAC Data from either collagen or control macrophages over collagen specific ATAC Peaks, which were centred, on the RUNX motif (b) or on the ETS motif; (d): H3K27ac tag count over collagen specific ATAC peaks in either collagen induced or control macrophages; Measured by ChIP-seq; (e): Region heatmap of c-MAF ChIP-seq tag counts and the ATAC read counts in either collagen induced or control macrophages over collagen specific ATAC Peaks. which were clustered using kmeans clustering (k=4).

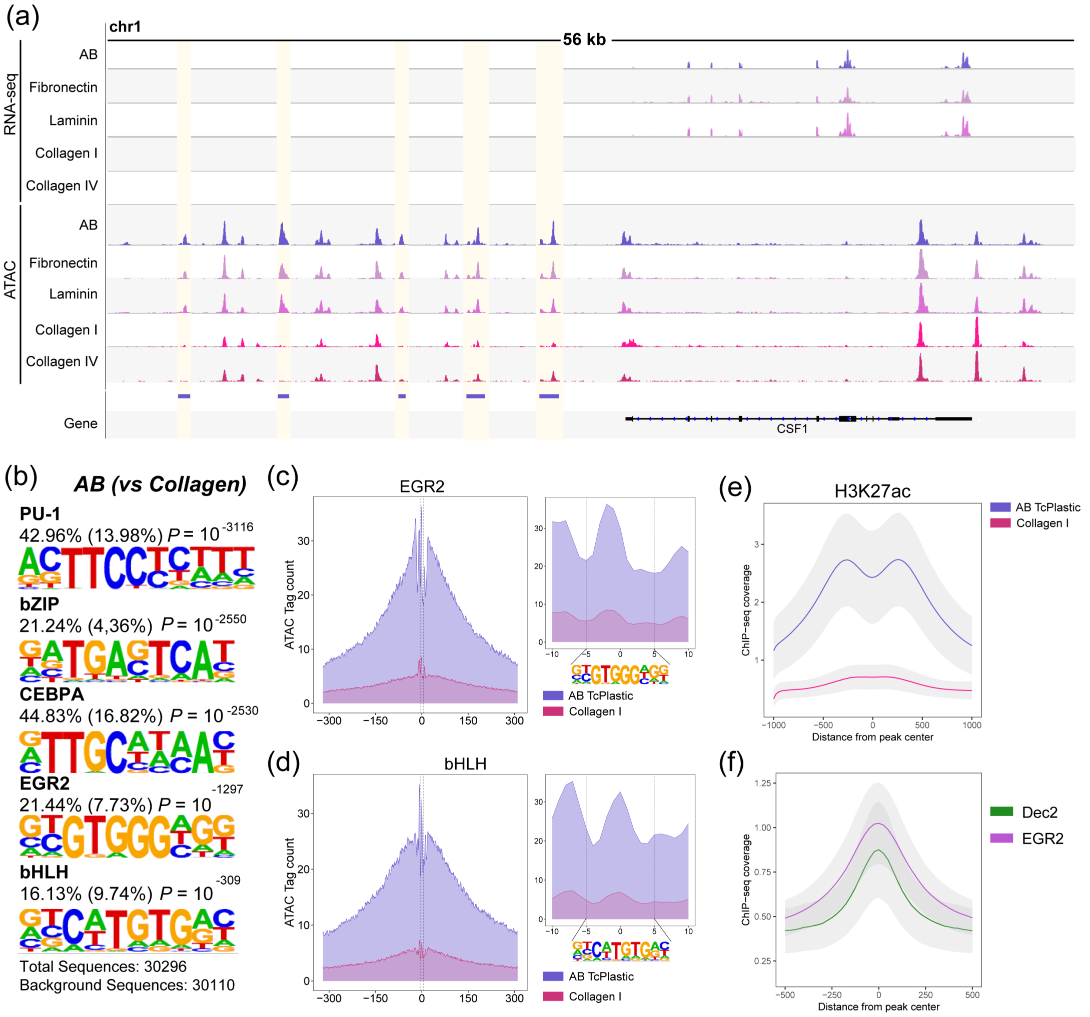

Figure 4.11: Collagen treatment leads to a strong loss of open chromatin

(a): IGV Browser Shot of the CSF1 locus; Yellow bars indicate AB (Control) specific ATAC signals; (b): Top 5 enriched de novo transcription factor motifs, analysis was performed using HOMER. Shown in the figure are the best matching transcription factors to the respective motifs, the percentage of target and background regions containing the motif and the p-Value; (c - d): Histogram of ATAC data from control and collagen macrophages over AB specific ATAC peaks, which were centred either on the EGR2 motif (c) or the bHLH motif (d); (e): Histogram of H3K27ac over AB specific ATAC peaks in either collagen or control samples. Grey area indicates the confidence interval; (f): Histogram of either EGR2 or Dec-2 ChIP tag densities over AB specific ATAC Peaks. Grey area indicates the confidence interval.

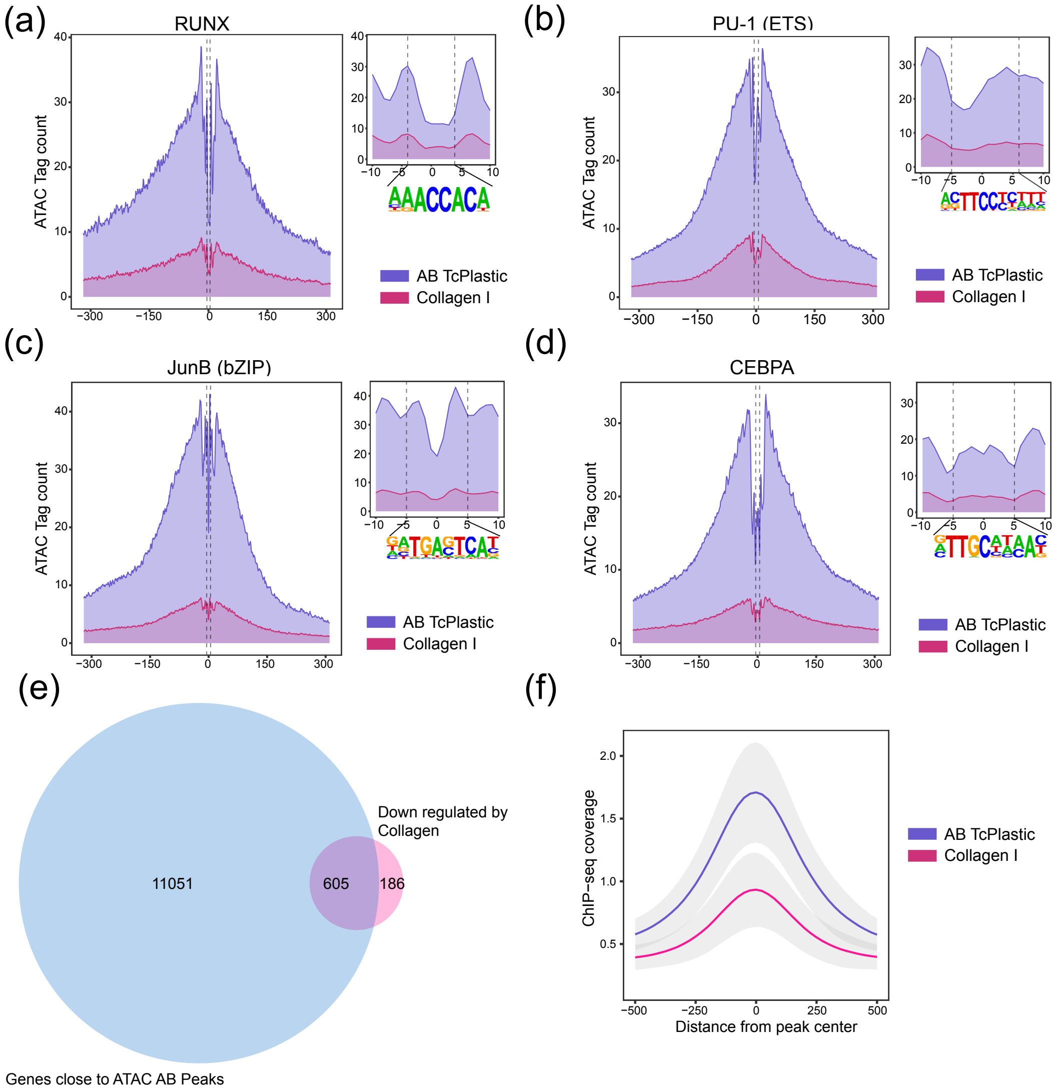

Figure 4.12: Open chromatin analysis reveals further candidate transcription factors
(a - d): Histogram of ATAC Data from either collagen or control macrophages over AB specific ATAC Peaks, which were centred, on the RUNX motif (a), on the PU-1 motif (b), the bZIP motif (c) or the CEBPA motif (d); (e): Venn Diagram, showing the overlap of Genes close to AB specific ATAC Peaks and Genes downregulated by collagen treatment;(f): c-MAF ChIP-seq tag coverage in either AB-Plastic or collagen induced macrophages over AB-Plastic specific ATAC peaks.

4.5.3 miRNAs induced by Collagen Treatment

In addition to regulation by transcription factors, post-transcriptional regulation by miRNAs plays an important role for cellular phenotypes [133]. These are small 21bp long RNA oligonucleotides, that possess regulatory functions. Based on sequence complementarity, they are capable of binding to target mRNAs, which leads to their translation inhibition or degradation, rendering them important regulators of cellular phenotypes. Therefore, we profiled the levels of mature miRNAs in our *in vitro* generated macrophages. Since standard total RNA sequencing poorly represents the availability of mature miRNAs in a cell, we used a small RNA sequencing technology established in the group of Prof. Dr. Gunter Meister at the University of Regensburg. After sequencing and mapping (see Section 3.3.6 for detailed code), the data was analysed using the EdgeR package. At first we performed a dimensionality reduction analysis using the t-SNE algorithm to display all samples in two dimensions.

The resulting plot, shown in Figure 4.13a, displayed a similar distribution of the samples as already described for the RNA-seq data. Thus, we used again the combination of the generalized linear model with the quasi-likelihood F-test to find miRNAs regulated by collagen treatment. We observed several miRNAs that were enriched in collagen type I and IV treated macrophages in comparison to plastic, fibronectin and laminin induced cells (see Figure 4.13b). To assess if they could be of interest in elucidating the regulation of collagen induced phenotype, we looked in the database TarBase v7.0 (included in Diana tools, miRPath v3.0 [182]), for reported and predicted targets of these miRNAs. One of them, miR148a-3p, was reported to interact with 41 genes down-regulated by collagen treatment, including the transcription factor Dec-2.

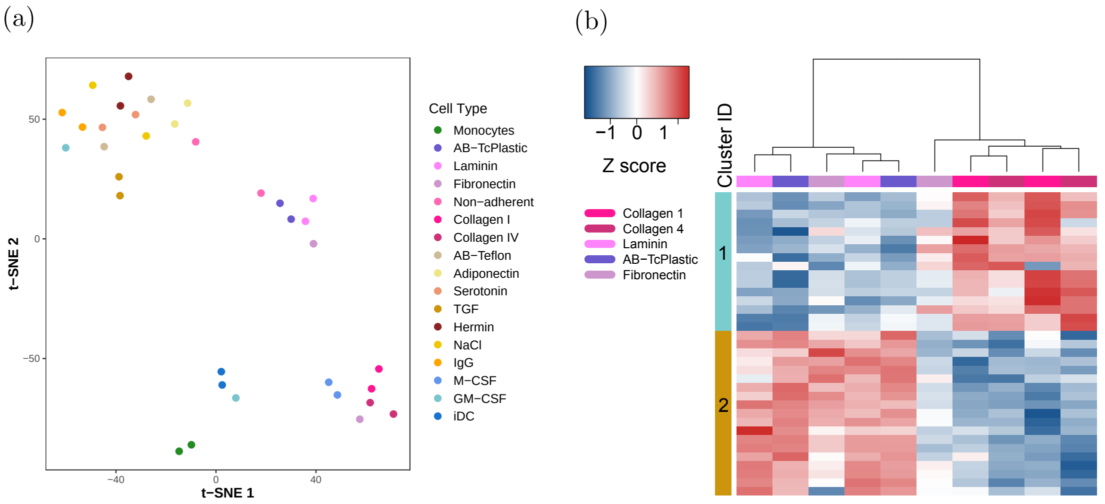

Figure 4.13: Collagen induces and represses a few miRNAs

(a): t-SNE Plot of all miRNA-seq samples, replicates are colour coded. In the t-SNE calculation perplexity was set to 2.5, theta to 0.13 and a maximum of 2000 iterations were done; (b): Heatmap of differentially occurring miRNAs between collagen treated macrophages and control samples; Up-regulated miRNAs are coloured in red, down-regulated in blue. Samples are indicated by column colours. miRNAs were selected based on a FDR < 0.05, cpm > 1 and logFC > 1.

4.5.4 miRNA148a-3p Knock Down

miRNA mediated inhibition of Dec-2 in collagen-macrophages could provide an explanation for the differential gene expression observed within these cells. In order to test the effect of miR 148a-3p, we performed knock down studies using a hairpin Inhibitor. It was electroporated into freshly isolated monocytes, which were set to differentiate with human serum and recombinant M-CSF, either on tissue culture treated dishes or on collagen type I coated plates. As negative controls we used an unspecific hairpin inhibitor and mock transfected cells. The level of Dec-2 expression was confirmed using qPCR and Western Blot analysis, both plots are shown in Figure 4.14. We could not detect any effect of the inhibitor on Dec-2 RNA expression or protein level. In macrophages adhered on plastic the expression level remained high, whereas on collagen it stayed low, independent of the haripin inhibitor we used.

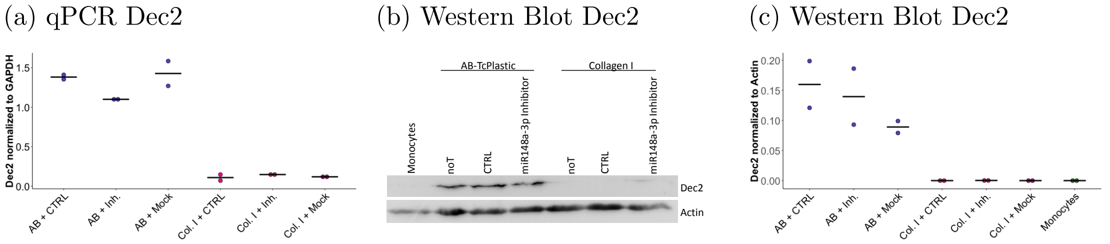
(a) qPCR Dec2
(b) Western Blot Dec2
(c) Western Blot Dec2

Figure 4.14: Expression Level of Dec-2 after miRNA148a-3p Knock Down
(a): Dec2 gene expression measured by qPCR in AB-TcPlastic and collagen treated macrophages, with or without miR148-3p inhibitor; (b): Representative Western Blot for miR148-3p Inhibition; (c): Quantification of Western Blots (n=2).

4.5.5 Dec2 Knock Down

To gain a better insight into the role of Dec-2 in regulating gene expression, we decided to perform knock down experiments using two siRNAs. These were produced by the company Axolabs and contained a terminal modification, which blocks binding of the oligonucleotides to pathogen recognition receptors, thereby suppressing a pro-inflammatory activation of the cells. As negative controls we used mock electroporated monocytes and a siRNA against the luciferase gene, which is not present in the human genome (CTRL). All three siRNAs were electroporated into freshly isolated monocytes. Since these are not proliferating, the siRNA level should remain high over the complete differentiation process of seven days, which was performed on tissue culture treated dishes. The expression level of Dec-2 was checked after 18, 66 hours and 7 days, by qPCR and Western Blot, displayed in Figure 4.15.

The knock down was highly efficient at the RNA level (see Figure 4.15a) and stable during the seven day differentiation protocol. All analysed donors expressed Dec-2 on protein level after seven days, whereas both knock down conditions showed no protein signal. Concluding, that the knock down was successful, we chose three donors to perform RNA sequencing in order to check the global gene expression changes after Dec-2 Knock down.

To gain insight into the relationship between all samples, we performed a dimensionality

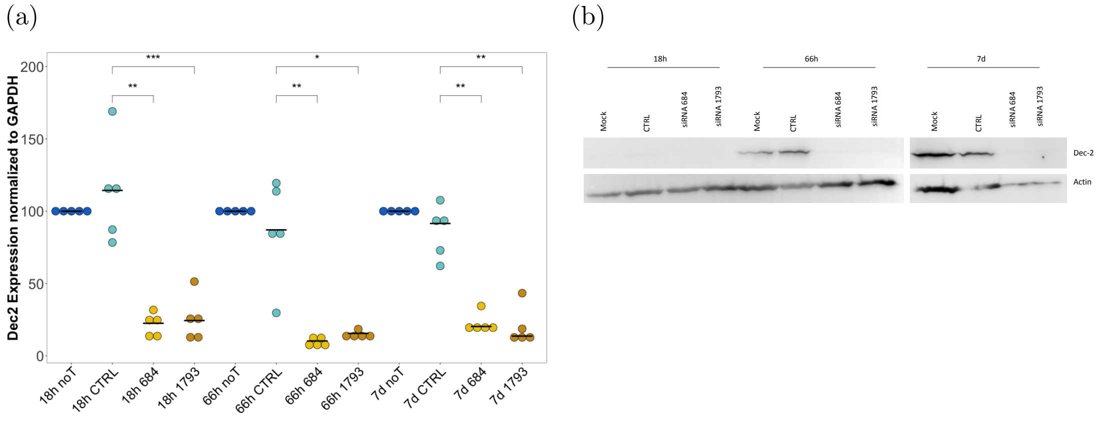

Figure 4.15: Dec2 Expression Level after Knock Down

(a): Dec-2 expression level, measured by qPCR. The expression values were normalized to GAPDH and the mock (noT) control was set to 100%. Significance was calculated using a paired t-test (n=5). (b): Representative Western Blot of Dec2 protein level after knock down.

reduction analysis using the t-SNE algorithm. In Figure 4.16a the distribution of all samples in two dimension is displayed. Monocytes clearly separated from the macrophages, which did not separate based on the Dec-2 knock down. In fact, the differences between Knock Down conditions seems to be greater than the differences between the knock down samples and control samples. Since both siRNAs reduced Dec-2 expression to the same level (Figure 4.15a) one of the siRNA seems to have unspecific side-effects. As a consequence, we focused our analysis on genes regulated by both siRNAs. We used the EdgeR package of R to perform a differential gene expression analysis using the generalized linear model in combination with a likelihood ratio test. This analysis resulted in few regulated genes, which are displayed in a Heatmap (see Figure 4.16b). These overlapped to a small extent with genes regulated by collagen, examples are shown in Figure 4.16c, including the gene encoding lysozyme an important mediator of innate immunity. Nevertheless, the knock down did not lead to the same drastic gene expression changes as collagen treatment does leading to the conclusion that Dec2 does not have a strong effect on macrophage polarization.

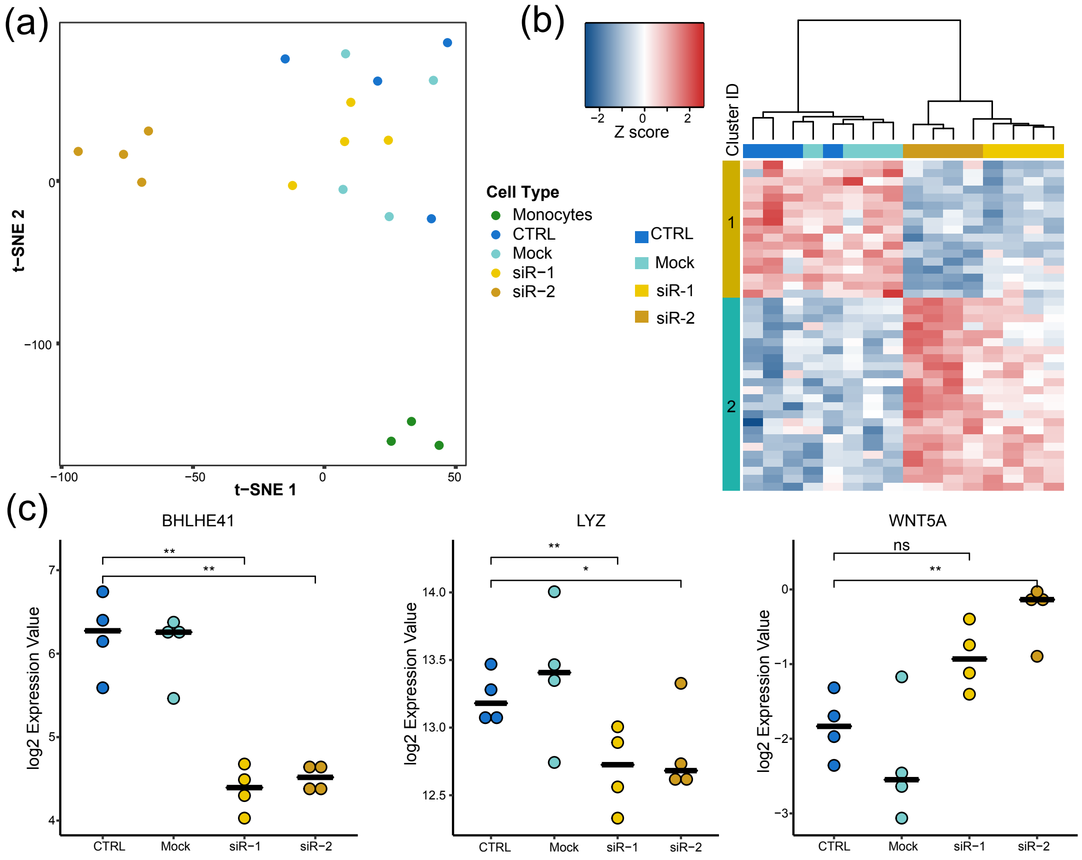

Figure 4.16: Analysis of the Dec-2 Knock Down

(a): t-SNE Plot of RNA-seq Data derived from Dec-2 knock down cells; The t-SNE algorithm was performed in R, using a perplexity of 3.5 and theta of 0; (b): Heatmap of differentially expressed genes between siRNA treated versus control treated macrophages, filtered for a logFC $>$ 1, cpm $>$ 1 and a FDR $<$ 0.05. Expression level is displayed using z-scores, red represent up-, blue down-regulated genes; Samples are displayed on columns and are colour coded; (c) Dot plots of selected Genes; Shown are logarithmic transformed expression levels measured by RNA-seq. Significance was calculated using a paired t-test (n=3);

4.5.6 EGR2 Knock Down

Another transcription factor down-regulated by collagen treatment is EGR2. To see if the reduced expression of this factor could be responsible for further genes regulated by collagen, we performed again knock down experiments using siRNAs. The efficacy of these was already checked by Corinna Kirschner in our lab. The experimental procedure used was the same as described for the Dec-2 knock down.

The cells receiving siRNAs against EGR2 showed a different morphology than control macrophages. They adhered poorly on the plastic dishes and developed a high number of vacuoles in their cytoplasm, as shown in Figure 4.17 (b). In fact they looked very similar to macrophages cultured on collagen treated dishes, as indicated by the microscope photographs of Figure 4.5, portending the conclusion that this knock down could capture more of the collagen phenotype than the knock down of Dec-2.

To assess the effect of EGR2 knock down on macrophages we performed RNA sequencing for three donors at day seven of differentiation. The analysis was done as described for Dec-2 knock down and collagen treatment (Code is given in Section 8). One of the first steps was encompassed a dimensionality reduction analysis, for which we used again the t-SNE algorithm. The resulting plot is displayed in Figure 4.18. We observed that in comparison to the Dec-2 knock down samples the EGR2 knock down cells formed a distinct cluster apart from the controls, whereas both macrophage cell types were clearly separated from the monocytes.

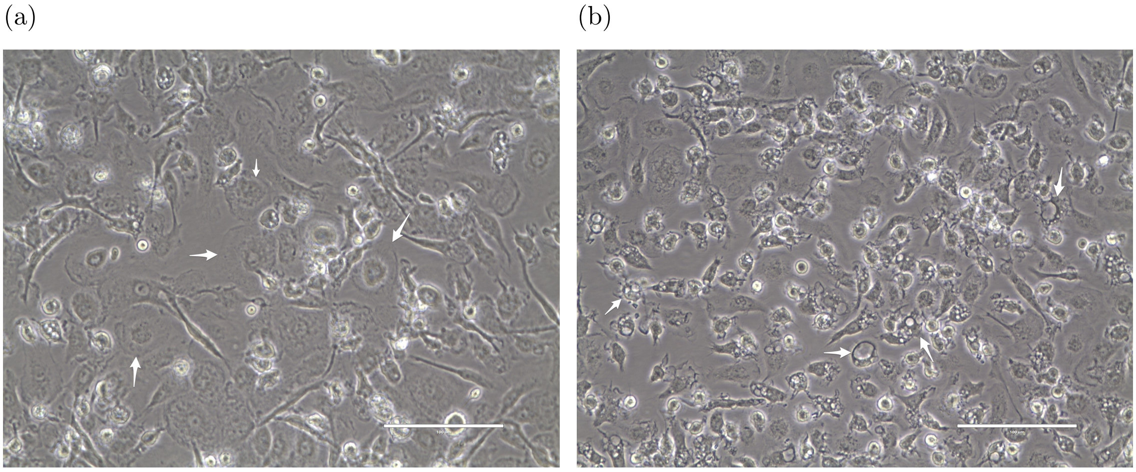

Figure 4.17: Morphology of Macrophages after EGR2 Knock Down
Microscope pictures of cells mock electroporated (a) or with siRNA 2665 against EGR2 (b) at day 7, taken at the Evos digital microscope with 40x amplification. Sample cells for either control or the knock down phenotype are indicated with white arrows.

In the next step, we searched for differentially expressed genes using EdgeR, for this analysis we used the generalized linear model in combination with quasi-likelihood F-tests. In order to find genes down-regulated by both siRNAs, thereby excluding siRNA specific effects, we combined all knock down samples into one group and compared them to both mock and CTRL transfected cells. This analysis was done in the batch-wise mode of EdgeR to reduce variability between replicates. Genes passing the thresholds were plotted in the heatmap, see Figure 4.19a. We observed 421 down- and 34 up-regulated genes,

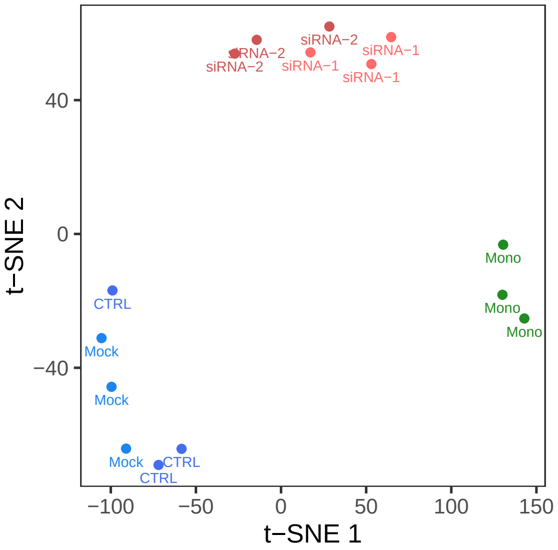

Figure 4.18: tSNE-plot of EGR2 knock down and control samples

indicating that EGR2 acts as a transcriptional activator in macrophages.

In order to determine whether the loss of EGR2 was responsible for down-regulation of gene expression by collagen treatment, we plotted the differentially expressed genes between collagen treated macrophages and control cells in a volcano plot (see Figure 4.19b). Here, dark blue coloured dots indicate genes down-regulated by both EGR2 knock down and collagen treatment, leading to a list of 311 genes. The overlap is also shown in the Venn diagram in Figure 4.19d. Furthermore, EGR2 knock down samples showed a significantly lower expression of genes down-regulated in collagen-macrophages than control or mock transfected macrophages (Figure 4.19f).

A gene ontology analysis of genes down-regulated by EGR2 knock down, using the online tool Metascape (http://metascape.org, [170]), showed that they are enriched for nearly the same categories (see Figure 4.19c, as we described for genes repressed by collagen treatment. The term extracellular matrix organization contained genes encoding collagen (*COL6A1, COL6A2, COL8A2, COL9A2, COL22A1*), laminin (*LAMB3*), fibronectin (*FN1*), integrins (*ITGA3, ITGAE, ITGB3*) and metalloproteinases (*MMP7, MMP8, MMP19*). Another enriched functional category was cell migration and chemotaxis including chemokine ligand 22 (*CCL22*), known to be regulated by EGR2 [183], *CXCR5, FLT1, SDC2, SDC4* as well as *CSF1*. Furthermore, several genes associated with immune response and myeloid leukocyte activation were affected by EGR2 knock down, for example *CHIT1, CHI3L1, BPI, IL1B* and *IL1RN*. A few examples of down-regulated genes are displayed in Figure 4.19e.

In addition to these findings, we observed that EGR2 knock down increased only a small amount of genes. This gene list overlapped with collagen induced genes to a small degree (see Figure 4.20a). Interestingly, the transcription factor IRF4 was significantly up-regulated and IRF8 showed at least a trended increase in gene expression (see Figure 4.20c), indicating that EGR2 represses their expression in macrophages. Both factors are capable of forming a dimer with PU-1 and bind a composite ETS:IRF motif, which we found to be enriched in collagen induced open chromatin (Figure 4.9b) and in EGR2 knock down induced open chromatin (Figure 4.24g). However most genes up-regulated by Collagen, such as C1QA, CD163 or CSF1R, were not induced by the knock down (Figure 4.20c), suggesting that the de-repression of IRF4/ IRF8 is not the main mechanistic circuit for these genes. In summary, these results led to the conclusion that during standard macrophage differentiation, EGR2 is induced by cell adherence to the plastic surface. When the cells were cultured on collagen fibres the interaction with this protein inhibits the up-regulation of EGR2, thereby failing to induce genes involved with extracellular matrix organization, cell-matrix adhesion, chemotaxis and immune response.

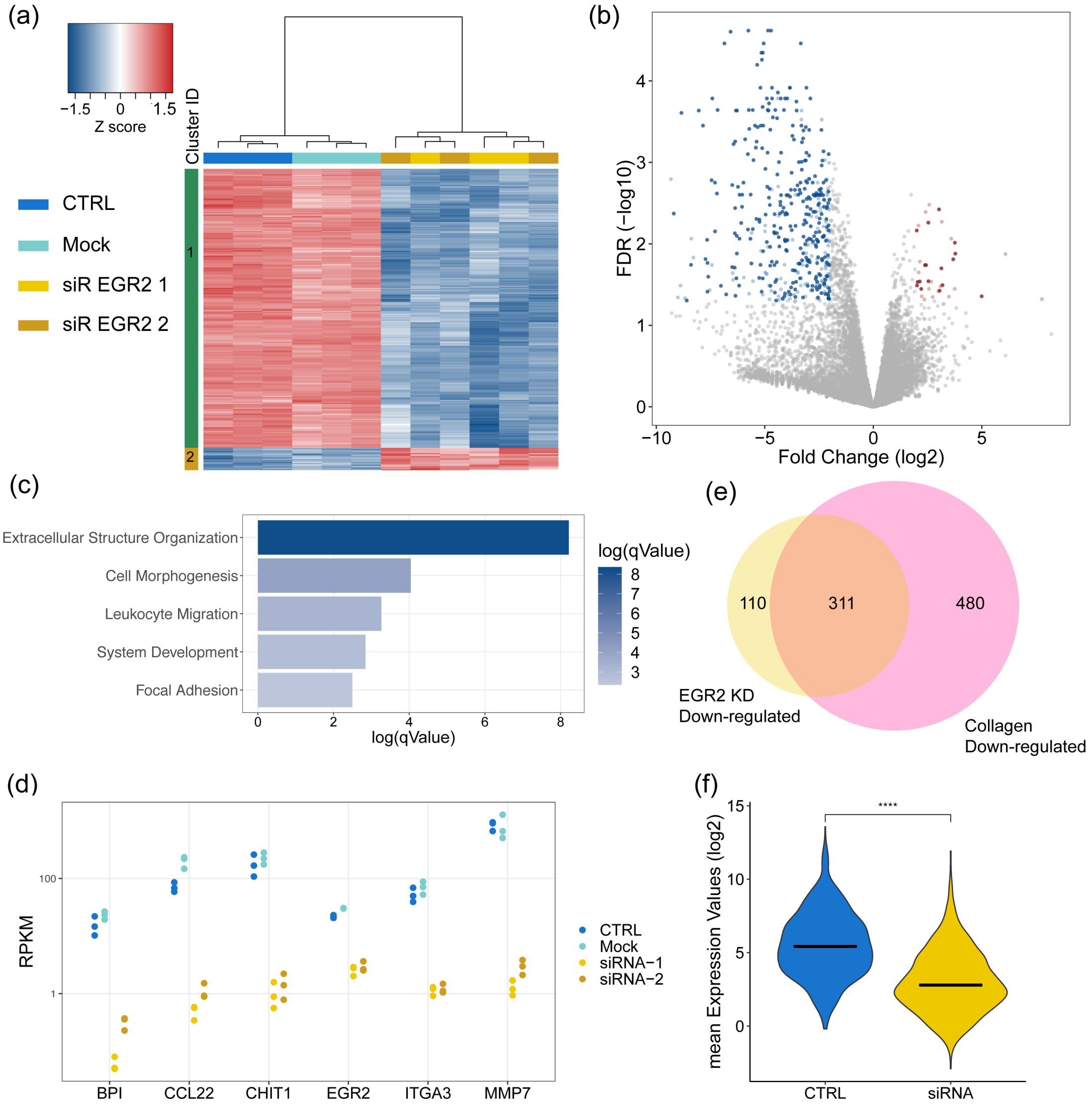

Figure 4.19: RNA-seq Analysis for EGR2 knock down Samples

(a): Heatmap of differentially expressed Genes between controls (including CTRL and Mock) and both EGR2 knock down samples, selected for a logFC > 2, cpm > 1 and FDR < 0.05; (b): Volcano Plot of the differentially expressed Genes, where dark blue dots represent genes down-regulated by both collagen treatment and EGR2 knock down, dark red dots represent up-regulated genes found in both analysis; (c): Gene Ontology Analysis, performed by the online Tool Metascape, of genes down-regulated by both EGR2 Knock Down and Collagen treatment. Five highly enriched terms were displayed in the bar plot in R; (d): Venn Diagram showing the overlap between genes down-regulated by both EGR2 knock down and collagen treatment; (e): Dot plot of selected down-regulated genes, where the expression level is displayed in RPKM values on a logarithmic scale; (f): Violin plot showing the mean expression level of control (both CTRL and Mock) or EGR2 knock down samples in genes down-regulated by collagen treatment. The significance was calculated using a paired t-test (n=6).

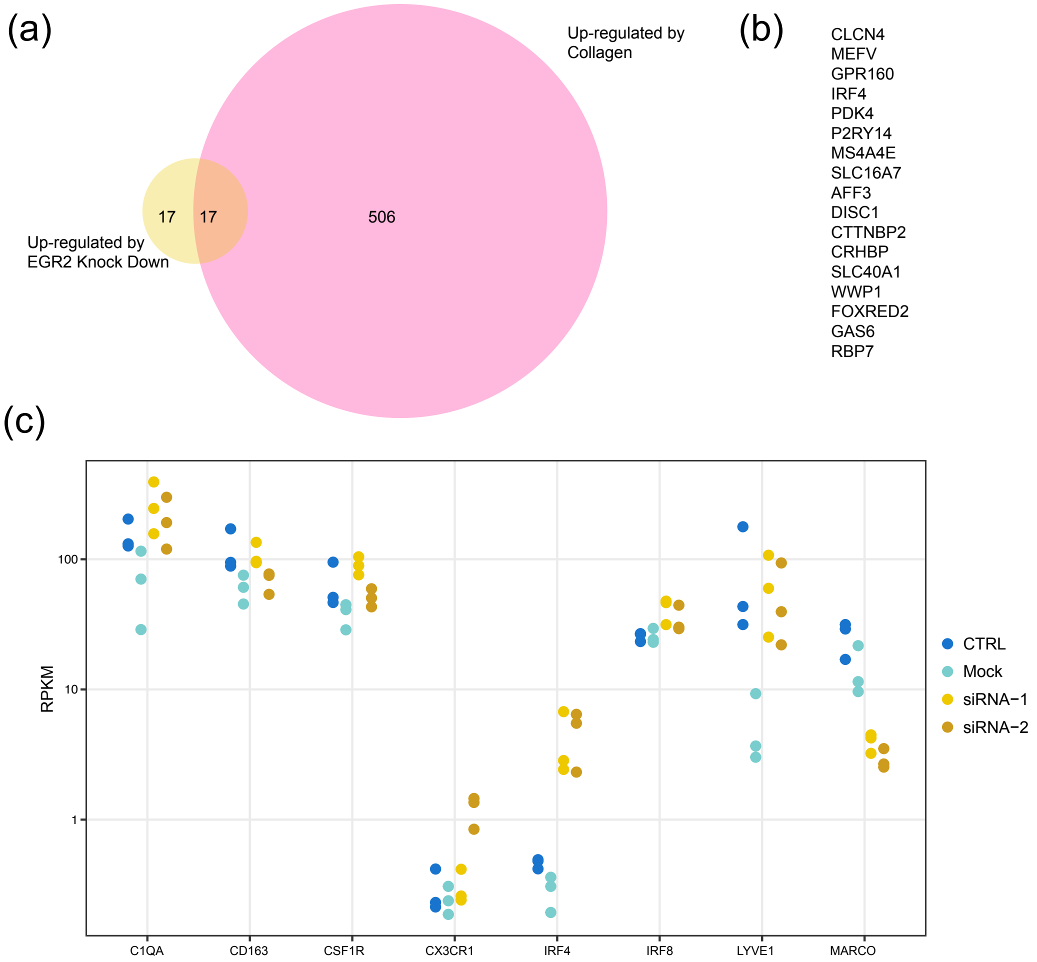

Figure 4.20: EGR2 Knock down induces only a small amount of genes

(a): Venn Diagram of Genes up-regulated by collagen treatment or EGR2 Knock Down; (b): Genes from the intersection of (a); (c): Dotplot of selected genes in either control or EGR2 Knock Down macrophages, shown are RPKM values for 3 replicates.

EGR2 knock down leads to a similar open chromatin structure as by culturing monocytes on collagen

To look more deeply into the phenotype induced by EGR2 knock down, we also profiled the open chromatin structure of the knock down samples using ATAC-seq. Again biological replicates showed a high Pearson correlation coefficient (Figure 4.21a-d). One of the first analysis steps taken was to look for regions specifically open in either control or knock down cells. We observed that the knock down inhibited the opening of many regions, considering the fact that we obtained 28533 peaks specific for controls, at the same time we found only 703 peaks specific for the control cells. Since the EGR2 knock down recapitulated the gene expression loss observed with collagen treatment, we analysed the similarity between open chromatin structures from EGR2 knock down, controls, plastic- and collagen-macrophages. For this we plotted the intensity of the ATAC signal in a 6000bp window centred on plastic-macrophage specific ATAC peaks (see Figure 4.22a). This analysis clearly shows that the loss of EGR2 leads to a similar chromatin structure as observed in collagen-macrophages. In contrast all other samples which express EGR2 showed an accessible chromatin structure at these regions.

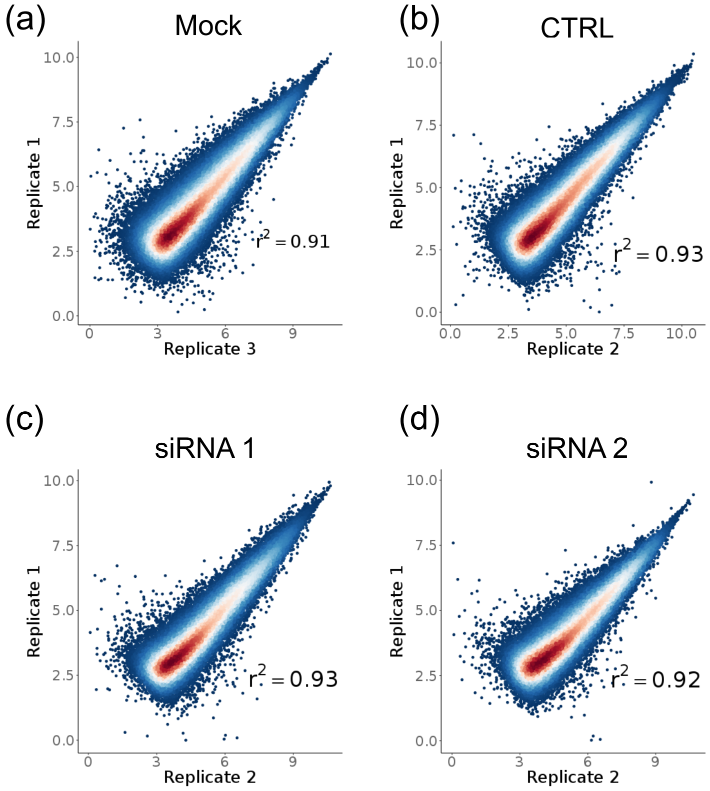

Figure 4.21: Similarity of ATAC replicates after EGR2 Knock Down
(a-d) Scatter plots between replicates of each condition. The tag densities in 1000bp around common ATAC Peaks were counted (using annotatePeaks.pl from HOMER) and the logarithmic transformed values plotted against each other in R using ggplot2. Pearson correlation coefficient was calculated for all samples using R and annotated to the plot. Dot colour indicates the density, blue equals low, red equals high dot density.

To exclude the possibility that the ATAC data from collagen-macrophages and EGR2

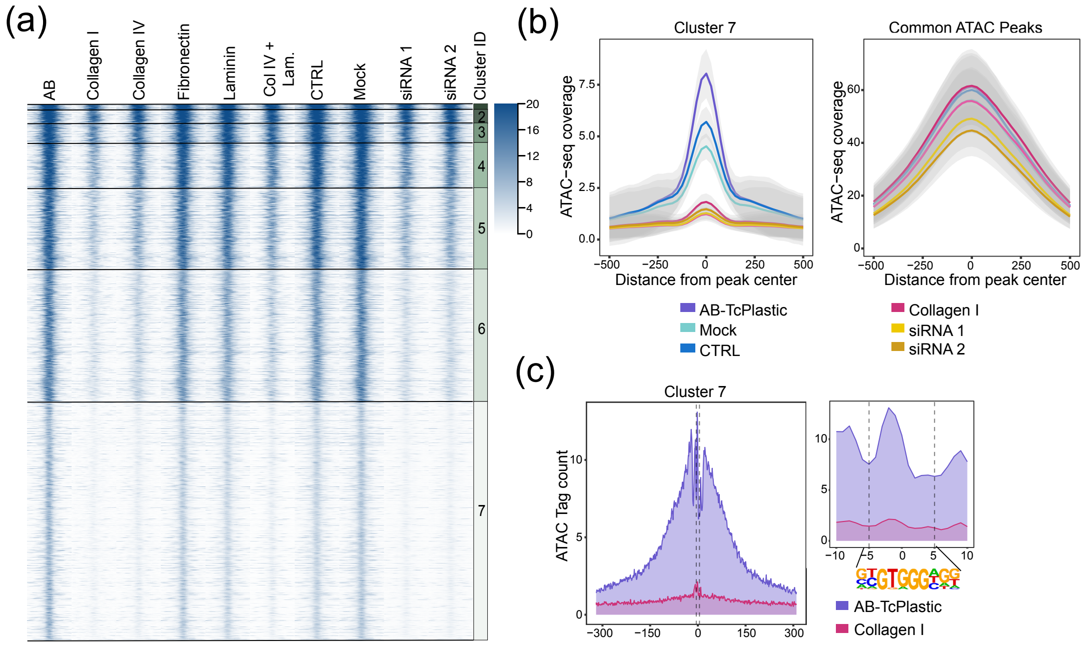

Figure 4.22: Overlap of the open chromatin signature between collagen treated macrophages and EGR2 knock down cells

(a): Region Heatmap showing ATAC signal over AB specific ATAC Peaks, which were clustered using kmeans clustering (k = 7); (b): Histogram of ATAC data over Cluster 7 of AB specific Peaks on the left side and over top 5000 common ATAC Peaks on the right side; (c) Histogram of ATAC data from AB and collagen treated macrophages over Cluster 7 Peaks centred on the EGR2 motif.

knock down samples generally has a lower tag density than plastic-, mock- and control-samples, we plotted the ATAC signal of all samples over a common peak set and over plastic-macrophage specific open regions (Figure 4.22b). Thereby we could show that plastic-macrophages show an increased open chromatin signature at specific regions not globally.

In the next step we looked at the enrichment of motifs from plastic-macrophage specific ATAC peaks in the different clusters. To achieve this, we used the top 8 motifs as known motifs in the findMotifsGenome.pl script. Afterwards, the enrichment of each motif was normalized to the mean background over all clusters and the result plotted in a balloon plot (see Figure 4.23a). For most motifs the enrichment decreases from cluster 1 to cluster 7, while the EGR2 motif shows an increased enrichment in cluster 7. Furthermore, we observed a distinct EGR2 ATAC footprint over regions contained in cluster 7, see Figure 4.22c. In summary these results indicate that EGR2 could bind to and induce opening of specific chromatin regions.

To further confirm this theory we also separately analysed the condition specific ATAC peaks, where we observed that the EGR2 knock down specific peaks overlapped to a small extent with collagen specific open chromatin, whereas both CTRL and AB specific peaks highly overlapped (Figure 4.23b). Therefore, it was not surprising that we observed again the standard transcription factor motive composition as published for macrophages in the control specific ATAC peaks [181]. ETS, bZIP, CEBPA, EGR and bHLH motifs were enriched in control-specific ATAC peaks. In addition, all motifs showed a transcription factor footprint in control-macrophage ATAC data (Figure 4.24b-f). In contrast, EGR2

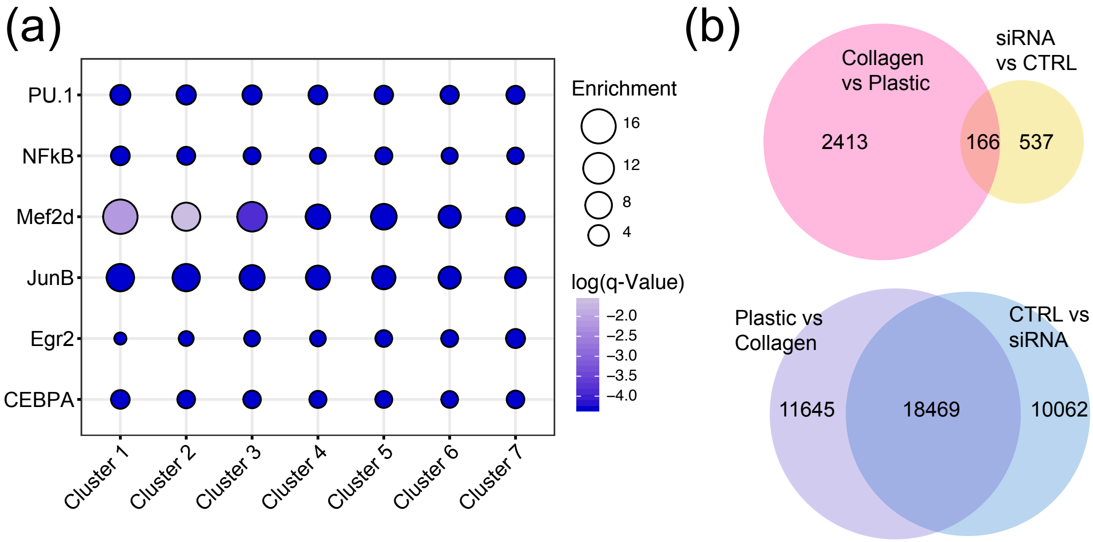

Figure 4.23: EGR2 Motif enrichment in AB specific ATAC peaks

(a): Balloon plot of motifs, which were highly enriched in AB specific ATAC Peaks (see Figure 4), in clusters 1 - 7. Balloon size represents the enrichment of the motif in the respective cluster over the mean background. The colour scale indicates the logarithmic q-Value of motif enrichment; (b): Venn diagrams showing the overlap between different sets of ATAC peaks. The overlap was calculated with HOMERs mergePeaks with default parameters and the -venn option.

knockdown specific open chromatin was enriched for an ETS, ETS:IRF composite and RUNT motif (Figure 4.24g). These motifs also showed an ATAC footprint as shown in Figure 4.24h - j.

In summary we showed that by the loss of EGR2, either by experimental knock down or by culturing monocytes on a collagen matrix, specific chromatin regions remained closed. In addition, the loss of EGR2 led to a loss of gene expression for many macrophage specific genes, including *CSF1*, *LYZ* and *CHIT1*. In contrast, the loss of EGR2 could not explain the induction of genes by collagen-macrophages. We observed an increased expression level and a respective motif enrichment for several transcription factors, however these analyses did not clearly identify a specific target factor.

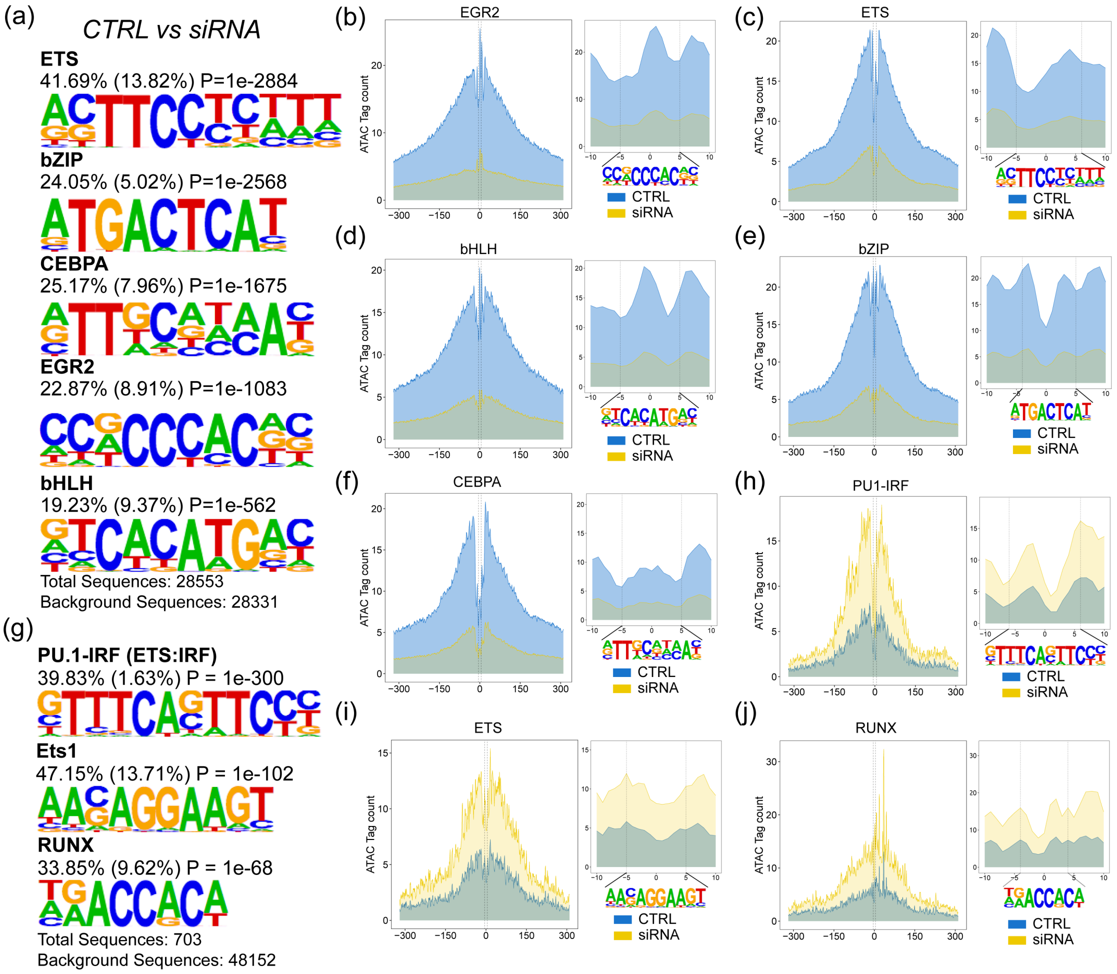

Figure 4.24: Open chromatin analysis of EGR2 knock down samples

(a): Enriched de novo transcription factor motifs in control samples compared to EGR2 knock down samples; The analysis was performed using HOMER, displayed is the best matching transcription factor, the percentage of target and background regions containing the motif and the p-Value; (b-f): Histogram of ATAC Data from control and EGR2 knock down samples over Control specific ATAC peaks, which were centred on the EGR2 motif (b), the ETS motif (c), the bHLH motif (d), the bZIP motif (e) and the CEBPA motif (f); (g); Enriched de novo transcription factor motifs observed in EGR2 Knock Down specific ATAC Pekas; The analysis was performed using HOMER, displayed is the best matching transcription factor, the percentage of target and background regions containing the motif and the p-Value; (h-j): Histogram of ATAC Data from control and EGR2 knock down samples over Knock Down specific ATAC peaks, which were centred on the PU-1:IRF composite motif (h), the ETS motif (i) and the RUNX motif (j).

4.6 Hofbauer Cell Phenotype

Since collagen is ubiquitously found in all extracellular matrices, we postulated that it's effect could be important for the phenotype of resident tissue macrophages. Accordingly we isolated human tissue macrophages from term placenta, the Hofbauer cells (procedure described in Section 3.2.2), in cooperation with Dr. Gregor Kisielewizc from the St. Josef clinic in Regensburg. In addition we obtained RNA-seq data for isolated bone marrow macrophages (BMM), provided by Dr. Heiko Bruns from the university clinic of Erlangen. Both datasets were compared to the *in vitro* generated macrophages treated with control conditions (AB-TcPlastic), laminin and both types of collagen as well as to freshly isolated monocytes.

4.6.1 Gene Expression Analysis

The analysis was done in R using RStudio, where we started with a dimensionality reduction analysis, in order to show all samples in relation to each other. The resulting t-SNE plot is shown in Figure 4.25 (a). Monocytes, BMM, Hofbauer cells and collagen-macrophages formed separate clusters, whereas plastic- and laminin-macrophages clustered together (Figure 4.25 (a)). The differences between plastic- and collagen-macrophages were clearly visible in this t-SNE plot as well. Interestingly, when looking at the Pearson correlation coefficient Heatmap, displayed in Figure 4.25 (b), we noticed that BMM where more closely related to monocytes, whereas collagen treated cells were more similar to the other *in vitro* generated macrophages. However, Hofbauer cells showed only a relatively small correlation to the remaining samples. Since the differences between collagen

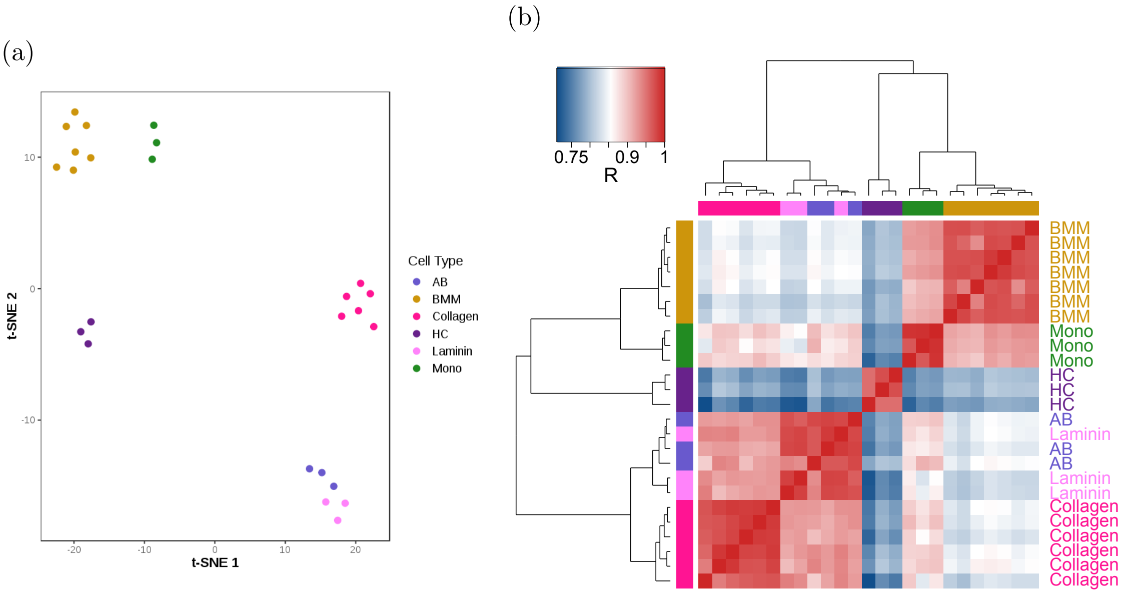

Figure 4.25: Similarity Assessment of Hofbauer cells, Bone Marrow Macrophages, Collagen-, Laminin-, Plastic-Macrophages, and Monocytes
(a): t-SNE Plot of Hofbauer cells (HC), Monocytes, BMM, laminin, AB-TcPlastic and collagen treated macrophages, performed with theta = 0.125, perplexity = 5 and maximum iterations of 2000; (b): Pearson Correlation Coefficient Heatmap, samples are indicated and colour coded on columns and rows.

and laminin have been discussed above and we proposed that interaction with collagen could be important for tissue resident macrophages, we decided to look for differentially or commonly expressed genes between Hofbauer cells, collagen treated macrophages, monocytes and bone marrow macrophages. Therefore, we performed a one-way analysis of variance (ANOVA) in EdgeR, the resulting heatmap is shown in Figure 4.26. Several clusters of differentially expressed genes were identified. The second cluster contained genes highly expressed in the Hofbauer cell population. The third cluster included genes expressed both in Hofbauer cells and collagen treated macrophages. Genes included in the fourth cluster were expressed by collagen treated cells, whereas the first cluster contained genes mainly expressed by bone marrow macrophages. In order to gain information about these genes, we performed again a gene ontology analysis using Metascape. Informations about the genes described below were taken from the online database GeneCards (www.genecards.org, April 26, 2019), if not stated differently.

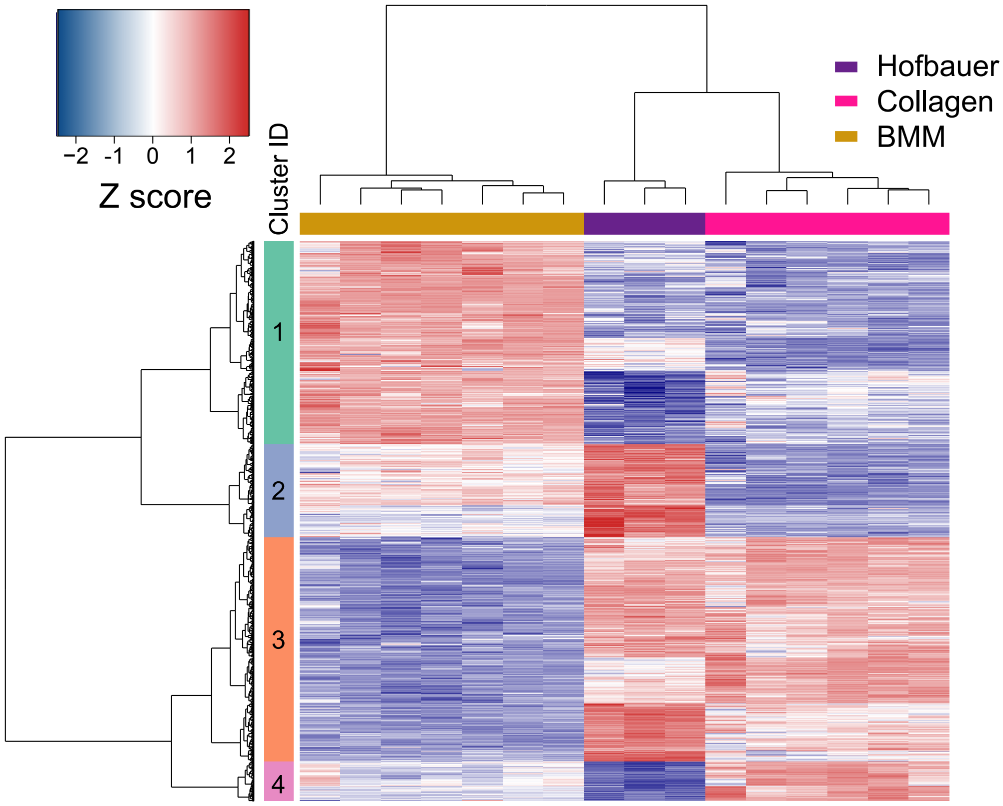

Figure 4.26: Heatmap of Differentially Expressed Genes between Hofbauer Cells, Collagen treated Macrophages, Monocytes and Bone Marrow Macrophages
Heatmap of genes found for the ANOVA analysis between Hofbauer Cells, Monocytes, Bone marrow macrophages and collagen treated macrophages, filtered based on FDR < 0.01 and cpm > 1. Expression values are displayed as z-scores, in red up-, in blue down-regulated genes, as shown in the colour key. Samples in columns are colour coded, as well as cluster identity in row colouring.

In the first step we performed an analysis on genes specifically expressed in Hofbauer cells, resulting in a network of enriched functional terms shown in Figure 4.27a. One of the most enriched categories is the blood vessel morphogenesis, encompassing genes for collagen (*COL4A1, COL4A2*) and metalloproteinases (*MMP2, MMP19*). This functional category contained the fibroblast growth factor receptor FGFR1, two gap junction proteins (*GJA5, GJC1*) and the LDL receptor related protein (*LRP2*) as well. Furthermore, several members of the WNT signalling pathway were found (*WLS, WNT2B, WNT5A*) to be induced in Hofbauer cells. WNT5a encodes the Wingless-Type MMTV Integration Site Family, Member 5A, responsible for regulating developmental pathways during embryogenesis. *MAP3K8*, part of the MAP signalling pathway, encodes an oncogene, which localizes to the cytoplasm, where it is capable of activating both JNK and MAPK pathways. *PPARG* encoding a transcription factor of the Peroxisome Proliferator Activated Receptor Gamma family was up-regulated as well. This factor has been reported to be activated by interleukin 4 signalling [184]. *SIRT1* encodes one of the most conserved protein deacetylase, which has been implicated in metabolic regulation inside various tissues [185]. Important for the correct morphogenesis of vessels is the regulation of angiogenesis factors. For this, we found two factors firstly angiopoietin like 4 (*ANGPTL4*) and secondly angiomotin (*AMOT*), the latter has been reported to inhibit the effect of angiostatin on endothelial cell migration. Closely related to vessel development is the term of embryonic and tissue morphogenesis, that encompasses *EFNA1*, encoding a member of the ephrin family of g-protein receptors involved with migration, repulsion and adhesion during development. In addition, this functional category included several proteins of the extracellular matrix, for example two glycoproteins (*EFEMP1, FBN1*), two laminin proteins (*LAMA3, LAMB1*) and another matrix component (*FRAS1*). Several members of the ADAM (a disintegrin and metalloprotease domain) protein family (*ADAM12, ADAM9, ADAMTS4, ADAMTS9*) were up-regulated in Hofbauer cells, which have been reported to be involved in cell-cell and cell-matrix adhesion. Furthermore, Hofbauer cells expressed three members of the heat shock protein family (*HSPB1, HSPD1, HSPH1*), the epidermal growth factor receptor (EGFR), arginase 2 (ARG2) and the homeobox transcription factor *MEIS2*. Another transcription factor up-regulated by these cells is *NR4A1* (alias Nur77), which belongs to the steroid-thyroid hormone-retinoid receptor family. Additionally, expression of the essential growth factor for macrophages, M-CSF, was up-regulated by Hofbauer cells.

In the next step we performed a gene ontology analysis for genes specifically down-regulated in Hofbauer cells (Cluster 4 in Figure 4.26), displayed in Figure 4.28. Here we found many functional terms associated with myeloid activation and inflammation. For example, two cathepsin family members (*CTSH, CTSS*), which encode lysosomal cysteine proteinases important for overall protein degradation. Another down-regulated enzyme is the human lysozyme (*LYZ*), whose natural substrate is the bacterial cell wall peptidoglycan, therefore possesses antibacterial activity. Viral defence is also an important part of innate immunity here we found to ribonucleases (RNASE2, RNASE3) to be down-regulated in Hofbauer cells. In addition, two c-type lectins (*CLEC4E, CLEC12A*), which are proteins involved in cell-cell, cell-matrix adhesion were down-regulated. Furthermore, two transcription factors were less expressed by Hofbauer cells, STAT6 and RUNX2. The latter has been implicated in osteoclast differentiation.

In the next step we looked more closely at genes up-regulated in both collagen treated

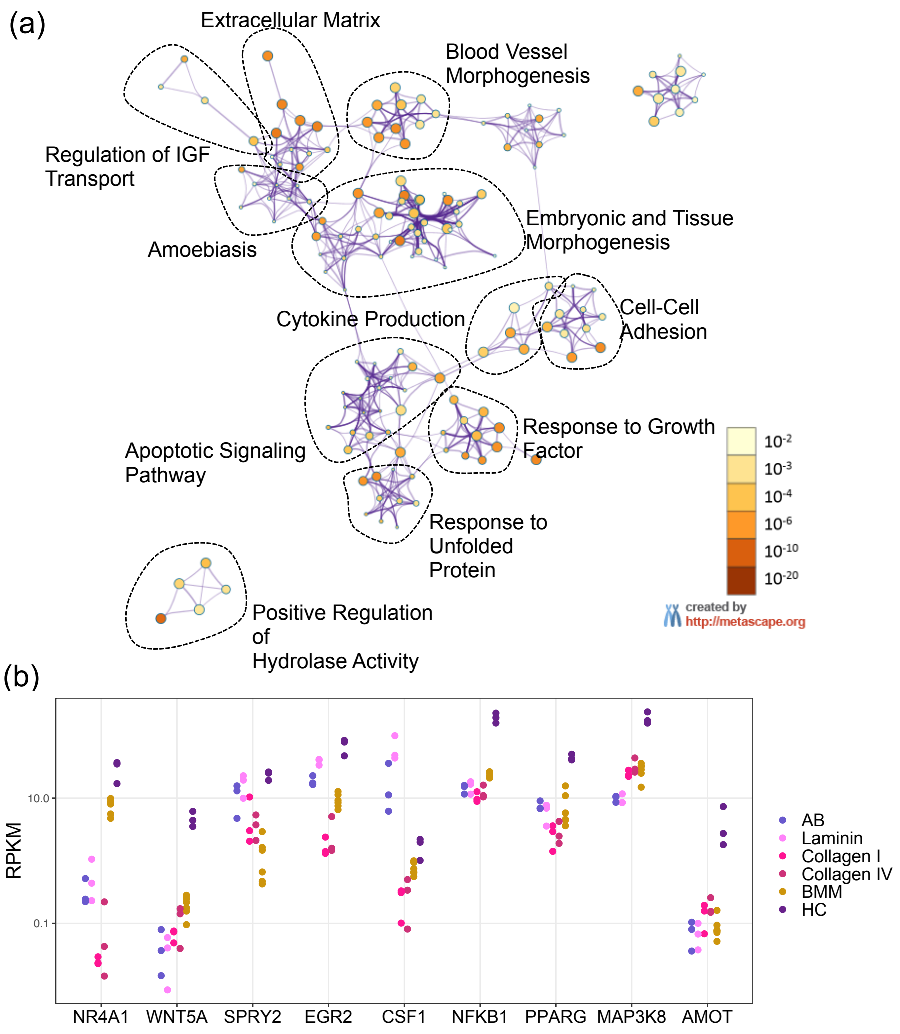

Figure 4.27: Gene Ontology Analysis of Genes expressed in Hofbauer Cells
(a): Gene Ontology Analysis of Genes expressed in Hofbauer Cells by the online tool Metascape. A network of enriched functional terms is shown, where the node colour corresponds to the respective p-value. Clusters of closely related terms were circled and labelled. (b): Dotplots of RNA-seq normalized log2 expression values for selected Genes.

macrophages and Hofbauer cells (Gene Cluster 2 in Figure 4.26), the gene ontology analysis is shown in Figure 4.29. These genes were associated with phagocytosis and endocytosis, including several scavenger receptors (*MRC1, STAB1, CD163, SCARB2, MSR1*). Hofbauer cells and collagen-macrophages expressed several growth factors (*PDGFB, VEGFB, IGF1*), members of the complement cascade (*C1QA, C1QC, C2*) and surface receptors (*TLR3, CD209, CSF1R*).

In summary, on the one hand Hofbauer cells specifically express genes involved with angiogenesis, vessel morphogenesis and response to growth factors. Additionally, they specifically up-regulate expression of the transcription factors EGR2, MEIS2, PPAR-γ and NR4A1. Furthermore, they share several genes with collagen treated macrophages, including scavenger receptors, complement proteins, and chemotaxis. The transcription factor MAF was also expressed by both cell types. On the other hand, the cells show down-regulated expression for genes involved with inflammation, antibacterial and antivi-

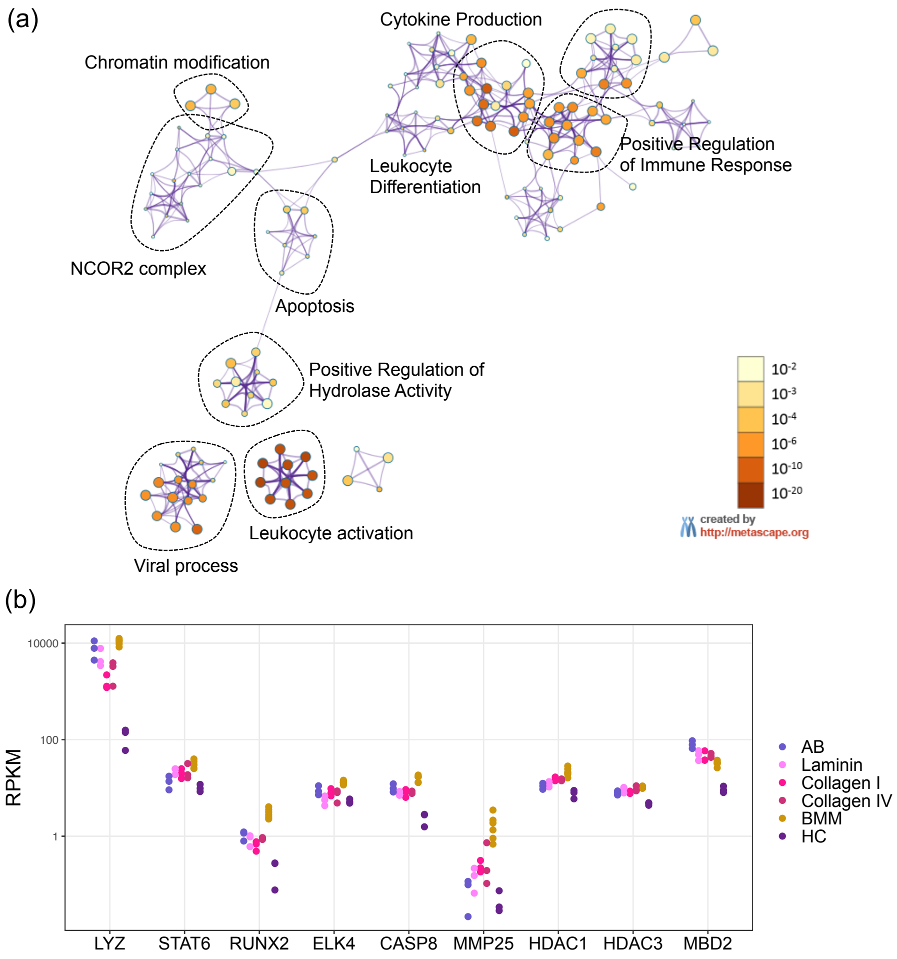

Figure 4.28: Gene Ontology Analysis of Genes down-regulated in Hofbauer Cells
(a): Gene Ontology Analysis by Metascape for genes in Cluster 3 of Figure 4.26. Nodes are coloured corresponding to the respective p-value, functionally related terms were circled and labelled; (b): Dotplots of RNA-seq normalized log2 expression values for indicated genes.

ral responses, as well as histone deacetylases. In addition, several transcription factors were slightly expressed including Elk4, TP53, STAT6, and RUNX2.

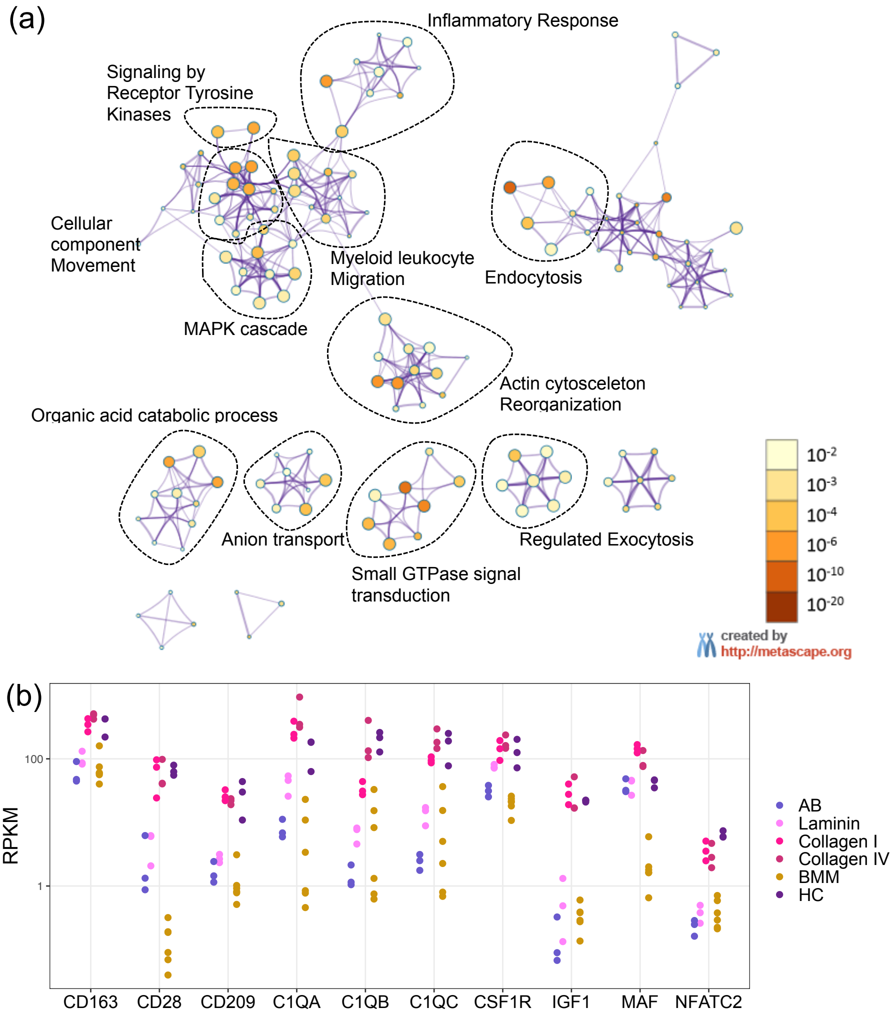

Figure 4.29: Gene Ontology Analysis of Genes expressed both in collagen treated macrophages and Hofbauer cells

(a): Gene Ontology Analysis using the Online Tool Metascape. A network of enriched functional terms is shown, where node size represents the amount of categories included, and colour corresponds to the p-Value. Clusters of closely related terms are circled and labelled. (b-h): Dotplots of RNA-seq log2 Expression Values for selected Genes. Significance was calculated with an unpaired Wilcox-test.

4.6.2 Open chromatin Analysis

In order to look for transcription factors, that could be involved in creating the unique phenotype of Hofbauer cells, we looked into the open chromatin of the cells using ATAC-seq. Biological replicates showed a strong Pearson correlation (Figure 4.30). After calling peaks on the ATAC data (45442), we annotated them to the human genome, and looked at their genomic distribution (see Figure 4.31 (a)). We compared the open chromatin of Hofbauer cells with that derived from all *in vitro* generated macrophages, in order to obtain peaks specific for the foetal cells. This resulted in 2384 Peaks, which we annotated again to the human genome and analysed their distribution over genomic features (see Figure 4.31 (b)). As shown in Figure 4.31 (a) most ATAC peaks were situated in introns and intergenic regions, but around 20% were found to be mapped to promoters and transcription start sites. In comparison, looking at the distribution of Hofbauer specific peaks (see Figure 4.31 (b)), only approximately 2% were found at transcription start sites, although the percentage of peaks in intergenic and intronic regions increased. Indicating that open chromatin changes occurred mostly at cis-regulatory elements such as enhancers not directly at gene level. As a result, we used these regions to look for enriched transcription factor motives, with the command findMotifsGenome.pl from HOMER. The resulting *de novo* motifs found are listed in Figure 4.31 (c).

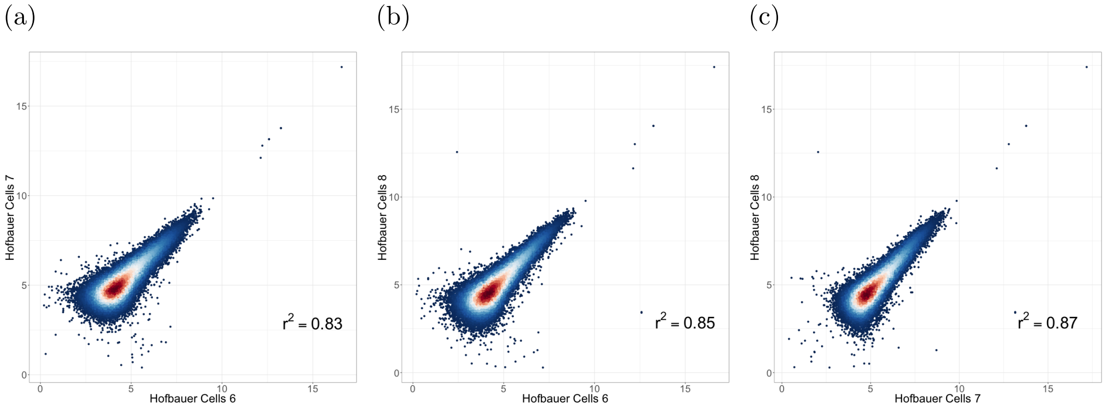

Figure 4.30: Scatterplots of Hofbauer cell Replicates

Scatter plots for the comparison of Hofbauer cell replicates with each other. The tag densities in 1000bp around common ATAC Peaks were counted (using annotatePeaks.pl from HOMER) and the logarithmic transformed values plotted against each other in R using ggplot2. Pearson correlation coefficient was calculated by R and annotated to each plot. Dot colour indicates the density, blue equals low, red equals high dot density. (a): Hofbauer cell preparation 6 vs 7; (b): Hofbauer cell preparation 6 vs 8; (c): Hofbauer cell preparation 7 vs 8.

Again the most highly enriched motifs were described already for macrophage enhancers, such as PU-1, bZIP and an e-Box/ bHLH motif [181]. However, we noticed that the MAF signature, which is enriched in collagen treated macrophages, was represented here too and a nuclear receptor recognition sequence, for which Nur77 (alias NR4A1) was the best match. The latter was also found to be specifically expressed by Hofbauer Cells on RNA level, compared to either collagen treated macrophages, bone marrow macrophages or monocytes (see Figure 4.27). Whereas MAF and NFATC2 were expressed both by collagen treated macrophages and Hofbauer cells, see Figure 4.29, which could indicate, that it induces expression of genes in both cell types.

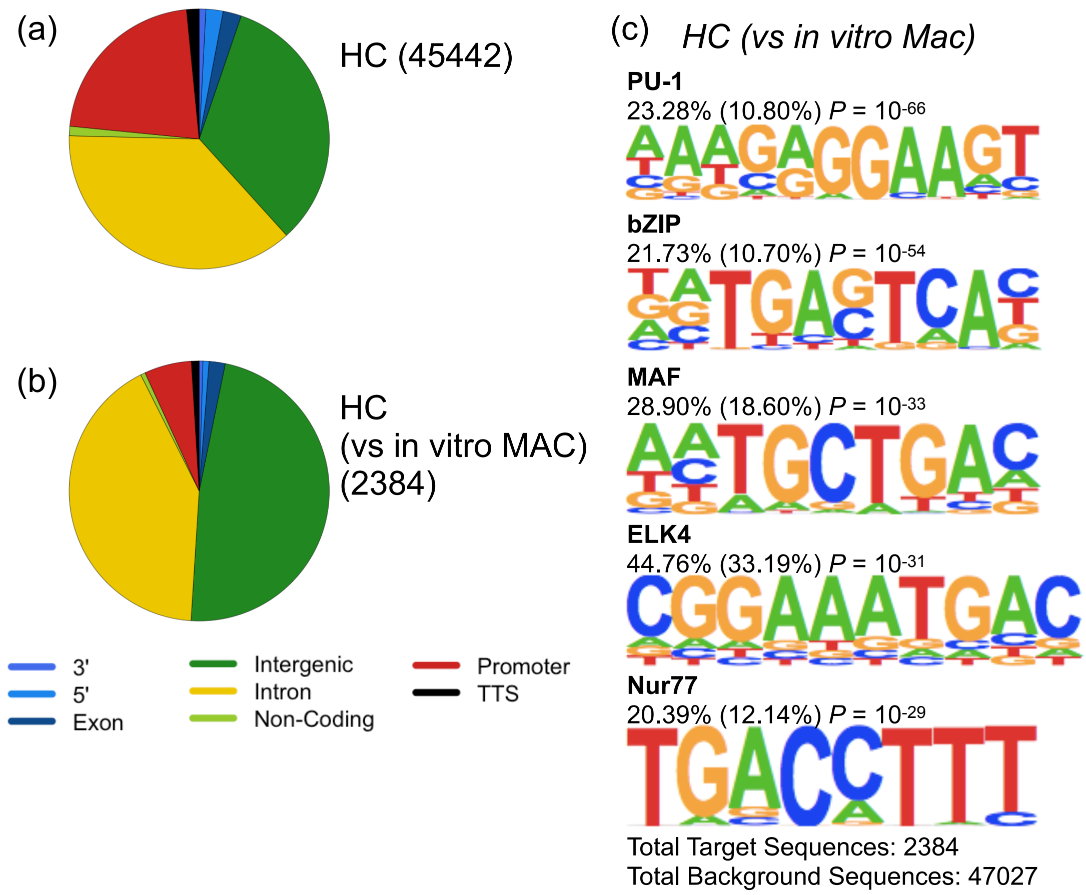

Figure 4.31: Open chromatin Analysis for Hofbauer Cells

Bar plots of the percentage of ATAC peaks found within specific annotated regions in the genome, we summarized them into intergenic regions, introns, promoters and transcription start sites (TSS), transcription termination sites (TTS), Exons, 3'- End and 5'- End; (a): All Peaks found for Hofbauer Cells (45442 Peaks); (b): Hofbauer Cell specific Peaks found by comparing them to all in vitro generated macrophages (2384 Peaks); (c): Enriched Transcription factor motives for Hofbauer cell specific Peaks.

In summary we showed that Hofbauer cells have a unique transcriptome signature, nevertheless they also showed some overlap with *in vitro* generated collagen-macrophages. These data confirm that the described effect of collagen on macrophage differentiation partly influences human tissue resident macrophages. However, it does not complete describe their phenotype, which is not surprising since tissue macrophages encounter many more signals in the tissue in addition to interacting with collagen.

Chapter 5

Discussion & Perspectives

5.1 *In vitro* Differentiation of Monocytes to Macrophages

Recent analyses of murine tissue resident macrophages have revealed that each one of them possesses their own transcriptome, phenotype and function. This diversity is controlled by tissue specific stimuli from their local environment [26]. Additionally, in an *in vitro* system of macrophage polarization with divers immune related stimuli, Xue and colleagues were able to show that monocyte derived macrophages reacted with an unique transcriptome to each challenge [59]. So far, macrophage polarization has mainly been analysed in murine tissue macrophages or with immune-related stimuli. Therefore, we asked ourselves what impact other factors, such as extracellular matrix proteins or a high salt concentration, could have on macrophage polarization. In this thesis, we differentiated human monocytes under various conditions into macrophages and applied high throughput techniques to globally analyse the resulting transcriptome (RNA-seq) and open chromatin landscape (ATAC-seq).

Gene expression analysis revealed that differentiation of monocytes on a collagen matrix or with M-CSF, GM-CSF or GM-CSF & IL-4 induced distinct phenotypes. In contrast, monocyte differentiation with a high salt concentration, TGF-β, heme, serotonin or adiponectin resulted in similar macrophages. Primarily, all of these stimuli were chosen, because they are known to influence macrophage activation, for example, heme, the co-factor of haemoglobin, has been reported to induce a specific gene expression program in red pulp macrophages [186, 27]. Additionally, it reduced inflammation in mice and inhibits monocyte chemoattraction [187, 188, 189, 190, 191]. Another example shows that serotonin treatment of M-CSF or GM-CSF differentiated macrophages led to decreased expression of M1- and increased expression of M2-related genes [192]. In summary, reported effects of the stimuli listed above were measured after relatively short incubation times, ranging from 4 to 48 hours, hence they described only the initial responses of the cells [187, 188, 189, 190, 191, 192, 193, 194]. In our case, we added all stimuli once to freshly isolated monocytes without refeeding and measured the effect after seven days to obtain information on their long term effects during macrophage differentiation. At this time point we did not observe significant gene expression changes, suggesting that either the concentrations used were too low or that the effects were gone after an initial response. In contrast, differentiation of monocytes with either M-CSF or GM-CSF induced distinct macrophage phenotypes. The differences between these two states have been studied extensively, indicating that maturation with M-CSF or GM-CSF induces a moderate anti- or pro-inflammatory state, respectively [50, 171, 55, 172, 51, 195]. The combination of GM-CSF with interleukin-4 is known to induce monocyte to dendritic cell differentiation, who, in comparison to macrophages, express many genes involved in antigen-presentation and cytokine production [173]. Using an ANOVA-like analysis on our transcriptome data, we obtained clusters of differentially expressed genes between these three conditions. These analyses confirmed reported gene expression changes, including a high level of CD163 in M-CSF treated macrophages or IRF4 in immature dendritic cells [50, 173]. However we were not able to identify new patterns with these conditions, hence we focused our analysis on the remaining conditions.

In addition to these cytokine treatments, the differentiation on extracellular matrix proteins induced diverse macrophage phenotypes. Especially, collagen-treated macrophages showed a strong polarization, whereas laminin- and fibronectin-treated cells were similar to plastic-macrophages. It has been shown previously, that short term interactions with laminin and fibronectin increased the phagocytic capability of macrophages [196]. We did not observe this effect in laminin- and fibronectin-macrophages after seven days of culture, maybe due to the expression of matrix metalloproteinases (MMP), which are responsible for degrading extracellular matrix proteins [197]. Suggesting that macrophages on laminin and fibronectin could have degraded the protein coating and interacted with the plastic below, thereby receiving the same stimuli as control-macrophages. Consequently, we focused our analysis on the effect of collagen, since it was strong and highly reproducible over different protein types, coating protocols and donors.

5.2 Collagen induced Macrophages

Collagen is one of the most abundant proteins in eukaryotes. It is mainly located in the extracellular matrix (ECM), where it provides physical scaffolding for surrounding cells [89]. Monocytes and macrophages can interact with these fibres inside tissues, hence the influence of collagen on macrophage polarization could be important *in vivo*. Collagen type IV is specifically found in basement membranes, a special form of extracellular matrix, which is located directly underneath epithelial cells, here it forms a three dimensional structure together with laminin and nidogen [90]. Therefore, we analysed the effect of collagen type IV in combination with laminin during macrophage differentiation. The resulting cells highly resembled macrophages differentiated on collagen alone, indicating that the effect of collagen was dominant over laminin. Interestingly, collagen type I and type IV induced highly similar macrophage polarization, which was characterized by strong gene expression changes. In contrast to fibronectin-, laminin- and plastic-macrophages, collagen-macrophages down-regulated genes associated with the extracellular matrix, exocytosis, immune response and cell-adhesion.

During the differentiation process, collagen-macrophages did not adhere as strongly to the substrate as plastic-macrophages and were dependent on supplementation with M-CSF in order to survive. The reduced adherence could be explained by the decreased expression levels of several integrins in collagen-macrophages. These proteins form heterodimers on the cell surface and mediate cellular interactions with the extracellular matrix, for example, the integrin $\alpha 1 \beta 1$ provides interactions with collagen [91]. In addition to integrins, collagen-macrophages lowly expressed several matrix-metalloproteinases, including *MMP8*, which can degrade collagen fibres [197]. Therefore we postulate that the collagen matrix is not degraded during the differentiation process, suggesting that the cells cannot interact with the underlying plastic. Adherence of monocytes to plastic has been reported to mediate strong gene expression changes, for example it induces *CSF1* and *EGR2* expression [198, 199, 85]. Collagen did not induce these genes, therefore we hypothesized that the adhesion stimulus on monocytes by collagen was not strong enough to induce these genes.

In addition to down-regulated genes, collagen-treatment induced several scavenger receptors, the insulin like growth factor 1 (*IGF1*) and complement complex C1q. It was previously shown, that differentiating monocytes on a collagen matrix increases their capability to phagocytose bacteria by activating complement receptors [97, 200]. In addition, induction of phagocytosis by scavenger receptor ligand binding is important during host defence, as these receptors recognize patterns broadly expressed by pathogens [201]. Collagen-macrophages also expressed the hemoglobin/ haptoglobin scavenger receptor CD163, which has been used as a marker for tissue macrophages [202], indicating that collagen treatment may recapture some features of tissue macrophages.

In vivo, differentiation of monocytes, in close contact with collagen fibres, could occur in the intestine, as peripheral blood monocytes constantly replenish the resident macrophage population [34]. Therefore, we hypothesized that collagen-macrophages could resemble colonic macrophages. The transcriptome of murine intestinal macrophages has been analysed and we noticed several overlaps, for example, both populations highly expressed complement C1q and phagocytosis associated genes (MRC1, GAS6) [203]. Additionally, an important growth factor for intestinal homoeostasis, the insulin like growth factor 1 (IGF-1), was induced in collagen-macrophages. IGF-1 is well known as an autocrine, endocrine and paracrine growth factor regulating proliferation and survival of epithelial cells in the gastrointestinal tract [204, 205, 206, 207]. Furthermore, several tissue macrophages have been shown to express IGF-1 [208, 209, 176]. These reports suggest that the expression of this growth factor in tissue macrophages could be induced *in vivo* by the interaction with collagen.

5.2.1 Collagen induces a unique open chromatin landscape

In order to understand the regulatory mechanisms behind the described gene expression changes, we analysed the open chromatin landscape of the cells. Comparing collagen-macrophages with plastic-macrophages, we observed specifically open chromatin regions, which possessed an unique transcription factor motif composition, consisting of ETS, ETS:IRF composite, MAF, RUNX and NFAT motifs. On RNA level several transcription factors were induced by collagen, including IRF4, MAF and NFATC2, which recognize some of the enriched motifs, therefore, we tried to analyse their binding patterns either by ATAC footprints or by ChIP-seq analysis. The ETS:IRF composite motif can be bound by PU1 in combination with IRF4 or IRF8 in order to regulate gene expression in macrophages [68, 71]. The reported control of the C1q genes by the PU1:IRF dimer provides a hint that these factors could be important to regulate further genes in collagen-macrophages [210]. In addition, the ATAC footprint over this motif was slightly stronger in collagen-macrophage specific regions compared to plastic-macrophage open chromatin, but the effect was too small to be responsible for all observed gene expression changes. Therefore, further transcription factors are probably involved in this process, for example NFATC2, which was highly expressed in collagen-macrophages and has been implicated in the development of osteoblasts [72, 73]. We observed that the NFAT motif showed a strong footprint in both collagen- and plastic-macrophages, indicating that also this transcription factor is not specific for collagen-macrophages and therefore was also not solely responsible for the collagen-induced gene expression. Since both NFATC2 and PU1:IRF probably were not be responsible for the induced gene expression changes, we analysed c-MAF binding with the ChIP-seq technique in both collagen- and plastic-macrophages. However, we observed that the open chromatin of both cell types was enriched for c-MAF binding, indicating that this factor was not specific for collagen-macrophages as well. Therefore, a combination of several transcription factors, their rearrangement on the chromatin and their post-transcriptional activation might be necessary to mediate collagen-induced gene expression.

It is also possible, that down-regulated transcription factors were responsible for the observed gene expression pattern. Dec2 and EGR2 were not expressed by collagen-macrophages and their binding sequences were enriched in the *de novo* motif analysis of plastic-macrophage specific open chromatin, suggesting that they could regulate macrophage specific gene expression. Dec2 is a basic helix-loop-helix transcription factor, which can bind to the e-Box motif observed in macrophage enhancers [181]. It has previously been shown to be involved in the circadian clock as a transcriptional repressor [211, 212, 213, 214]. However, we wanted to analyse the effect of reduced Dec-2 expression in macrophages without any confounding influences. Therefore, we performed Dec2 knock down experiments in monocytes, which were then differentiated on plastic dishes. We observed only a few differentially expressed genes, leading us to the conclusion that Dec2 is not the main factor responsible for the loss of gene expression. Hence, in the next step we analysed the effect of EGR2 during macrophage differentiation.

5.2.2 The role of EGR2 during plastic-macrophage differentiation

EGR2, along with three other members of the early growth response family (EGR1, EGR3, EGR4), is expressed in several cell types, including myeloid cells and T cells [215, 216, 217]. EGR1 was early on identified as an important regulator during macrophage differentiation [218], whereas EGR2 was first characterized in the central nervous system as a regulator of peripheral nerve myelination [219, 220]. Consequently, mutations in the EGR2 gene result in different demyelinating neuropathies, such as Charcot-Marie-Tooth disease [221]. In addition, these mutations are associated with a bad prognosis in chronic lymphocytic leukemia, further illustrating the importance of this transcription factor [222, 223]. The transactivation activity of EGR2 can be modulated by corepressors such as NGFI-A-binding protein 2 (NAB2) [224], both proteins are important during the differentiation of progenitor cells to macrophages by inhibiting expression of the neutrophil program and activating macrophage specific genes [225]. During *in vitro* differentiation of monocyte to macrophages, EGR2 induction occurs soon after adherence on a plastic surface [85, 226], but it can also be induced by growth factors and interleukin-4 in macrophages [227, 228, 229]. We did not observe this expression pattern in collagen-macrophages in con-

trast to plastic-macrophages. Suggesting that the reduced adhesion stimulus in collagen-macrophages was responsible for the loss of EGR2 expression in these cells.
To further analyse the importance of EGR2 in macrophages, we performed siRNA mediated knock down experiments. Already during the differentiation process, we observed a similar morphology in EGR2 knock down cells and collagen-macrophages, hence it was not surprising that the transcriptome analysis revealed an overlap between genes down-regulated by EGR2 knock down and collagen-treatment. We observed a reduced expression of collagens, integrins, matrix-metalloproteinases and chemokine ligands in both cell types. For example, CCL22 was reported previously to be under the control of EGR2 in macrophages [183]. Additionally, we could show that EGR2 knock down inhibits expression of NAB2 and CHI3L1. In contrast, only a few genes were up-regulated by this knock down, indicating that EGR2 primarily acts as a transcriptional activator and induces many macrophage specific genes. The accessible chromatin landscape of EGR2 knock down macrophages showed reduced accessibility of chromatin in the same regions, as observed for collagen-macrophages. These regions were highly enriched for the EGR2 motif, which showed a strong footprint in control macrophages. This data indicates, that EGR2 expression during monocyte to macrophage differentiation is required for the successful expression of macrophage specific genes. In the central nervous system, EGR2 has been shown increase recruitment of the SWI-SNF complex by SOX10 to induce chromatin remodelling [230]. A similar mechanism could be involved in the activation of macrophage specific gene expression.

5.3 Hofbauer Cells

In vitro generated macrophages cannot recapture the complete phenotype of human tissue macrophages, as these cells encounter many different signals in their environment. Therefore, we wanted to analyse the transcriptome and open chromatin landscape of human, primary tissue macrophages. To achieve this goal we cooperated with the St. Josef hospital in Regensburg to obtain placentas from voluntary caesarian dissections. Additionally, we obtained RNA-seq data of human bone marrow macrophages, kindly provided by Dr. Bruns at the university clinic in Erlangen. Placental tissue contains foetal macrophages, so called Hofbauer cells, which are located inside the chorionic villi of the placenta [36]. They appear early in gestation and are important for the defence against pathogens, but also for angiogenesis and development of the tissue structure [41, 43, 231, 42]. Especially, high levels of M-CSF, IGF1 and VEGFA were shown to be important by inducing trophoblast growth and differentiation [232, 233]. Therefore, it was not surprising that we observed a strong expression of several growth factors, including Sprouty2, IGF-1, M-CSF and VEGFA in Hofbauer cells. Furthermore, it has been shown previously, that WNT signalling is important for human oestrogen-dependent endometrial cell proliferation, decidualisation, trophoblast attachment and invasion (Reviewed in [234]). Several components of this pathway were highly expressed in Hofbauer cells, including *WNT5A*, a secreted glycoprotein. Hence, the production of these factors by placental macrophages could be important for correct tissue development and a successful pregnancy.

Additionally, Hofbauer cells have been shown to possess an anti-inflammatory phenotype, which is important to maintain immune tolerance during pregnancy [44, 46]. The gene ontology analysis of genes down-regulated by Hofbauer cells included several terms such as cytokine production and viral processes. For example, the gene encoding lysozyme, an important antimicrobial enzyme, was not expressed in the cells [235], hence we were able to confirm the described Hofbauer cell phenotype and function.

In order to understand the regulatory mechanism behind the described gene expression changes, we analysed the open chromatin landscape of the cells. Comparing Hofbauer cells to all *in vitro* generated macrophages, we obtained 3775 specific open chromatin regions. These were enriched for an ETS, bZIP, NR4A1 and MAF motif. The nuclear receptor NR4A1 (alias Nur77) is involved in several biological processes, including metabolism, stress and immunity [236]. It is essential for the development of murine Ly6C(low) monocytes [237], whereas in macrophages it is induced during the phagocytosis of apoptotic cells by inhibiting NFκB signalling and IL-12 production to prevent induction of an inflammatory response [238]. Additionally, Nur77 was highly expressed in the analysed foetal macrophages, therefore it could be one of the responsible factors mediating the phenotype of Hofbauer cells. In addition to NR4A1, we observed an enriched MAF motif in the accessible chromatin regions of Hofbauer cells. Interestingly, the same motif was enriched in collagen-macrophage specific open chromatin. Besides that, both collagen-macrophages and Hofbauer cells highly expressed MAF, therefore this transcription factor could be responsible to induce commonly expressed genes. Both cell types expressed several scavenger receptors (*STAB1, MARCO, CD163, CD209*), complement factors (*C1QA, C1QC*), the growth factor IGF1 and the cell surface receptors (*CD28, CSF1R*). In summary these analyses indicate that collagen-macrophages partly capture the phenotype of tissue macrophages. The interaction with collagen inside the tissue could be important for the phenotype of resident macrophages.

CD163 has been intensively used as a marker for Hofbauer cells, which is why we sorted for CD163 positive cells to obtain a pure macrophage population [239]. The isolation procedure lasted up to seven hours and consisted of tissue dissection, several digestion steps, density centrifugation and cell sorting. Since macrophages are very responsive to changes in their environment, the isolation procedure could have had an impact on the resulting cellular phenotype, as has been described for rat alveolar macrophages [240]. Hence, we cannot exclude the possibility, that some of the described Hofbauer cell phenotype could be due to tissue processing.

In summary we could show that Hofbauer cells express essential factors for the correct tissue morphogenesis inside the placenta. Additionally, many genes involved in inflammation were down-regulated in Hofbauer cells, confirming their described anti-inflammatory phenotype [44, 42]. A transcription factor candidate for regulating these genes could be NR4A1, since it is highly expressed by the cells and we observed its binding motif in the open chromatin regions specific for Hofbauer cells. In addition, we could show that several collagen-induced genes were expressed by Hofbauer cells as well, including complement C1q and scavenger receptors. Consequently we postulate that interaction of the cells with collagen fibres inside the tissue increases their capacity to recognize and eliminate pathogens.

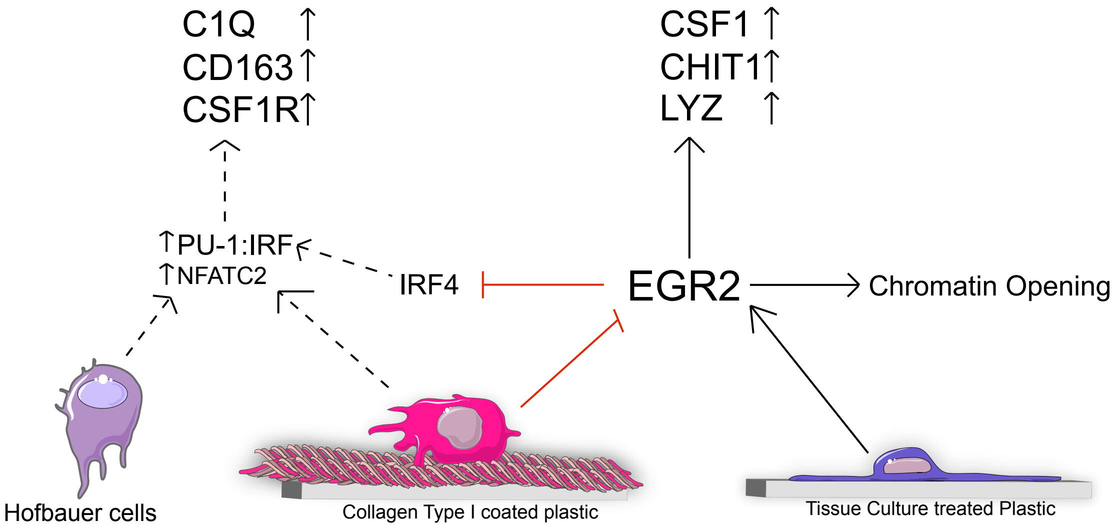

Figure 5.1: Schematic view of the differences between collagen- and plastic-macrophages

Conclusively, the results obtained in this thesis expand our current knowledge of macrophage polarization by one stimulus, as we could show that culturing monocytes on a collagen matrix effectively changed the transcriptome and open chromatin structure of the cells. One of the main differences between collagen- and plastic-macrophages was the expression level of EGR2, our current hypothesis is depicted in Figure 5.1. Using knock down experiments, we could confirm that EGR2 expression during monocyte to macrophage differentiation is needed for chromatin opening and activation of macrophage specific genes. In addition, collagen-macrophages highly expressed genes associated with inflammatory response and endocytosis. Interestingly, some of these genes, especially the scavenger receptor CD163 and complement C1q complex, were highly expressed by Hofbauer cells as well, suggesting that the interaction with collagen fibres might have an impact on the phenotype of tissue resident macrophages *in vivo*.

5.4 Perspectives

Macrophage activation has been revised from the dichotomy of two opposing states to a spectrum model of activation [58]. In this thesis we could expand the current knowledge on macrophage polarization capabilities by one more stimulus. We showed that especially the adhesion *in vitro* to collagen coated plastic dishes, during the differentiation from monocytes to macrophages, had a great impact on their phenotype. We showed many changes on gene expression level, which were mirrored by the open chromatin analysis. Unfortunately we could not identify a specific transcription factor responsible for mediating collagen-induced gene expression changes. Therefore further experiments analysing the binding patterns for ETS2, NFATC2 or PU1/IRF should be done to clarify their role.

Additionally, the post-transcriptional activation of transcription factors is an important regulatory mechanism, hence the phosphorylation of IRF and NFAT could be further investigated in collagen-macrophages.

Furthermore, we could show that EGR2 expression is important during macrophage differentiation to activate gene expression and increase chromatin accessibility at specific regions. But the precise mechanism how EGR2 mediates these changes is not clear yet. Therefore it would be interesting to globally map interaction partners of this transcription factor to see if it actively recruits chromatin remodelers.

Another interesting aspect for future study would be to mix more extracellular proteins into the collagen coating, in order to create an environment more similar to the *in vivo* situation, since macrophages don't encounter collagen on it's own, but in the context of proteoglycans, fibronectin and laminin inside the extracellular matrix. It could be that macrophages react differently to a mixture of extracellular proteins, than to each one alone.

Furthermore, functional studies on collagen-macrophages, for example phagocytosis assays, could validate the phenotype described in this thesis. In addition, it would be interesting to see if collagen-macrophages react differently to known activation stimuli, for example LPS or interleukin-4.

In the second part of this thesis we demonstrated, that Hofbauer cells from term placenta possess a unique transcriptome, which could be regulated by increased MAF and NR4A1 expression. Therefore ChIP-seq experiments for these transcription factors could clarify their role in these cells. Since we analysed macrophages from term placenta, the tissue has already fulfilled its function and gone through all developmental stages. It would be interesting to compare Hofbauer cells from first trimester placental tissue with foetal macrophages from term placentas, in order to see if the cells change during tissue development.

In addition, it would be of great interest to analyse more tissue resident macrophage populations, particularly intestinal macrophages. These are the only resident population, which is constantly replenished by peripheral blood monocytes. Hence they could be more closely related to the collagen-macrophages presented in this thesis.

Generally, most information about tissue macrophages was obtained in the murine system, but this data cannot simply be transferred to the human situation. Therefore more human tissue populations need to be profiled in order to gain an accurate picture of human macrophage phenotypes.

Chapter 6

6.1 Summary

Macrophages are an essential part of innate immunity, here they are responsible for recognizing pathogens and their elimination, as well as activating cells of the adaptive immune system, by antigen presentation. Furthermore, in every tissue there is a resident macrophage population, which performs fundamental homoeostatic functions in addition to host defence. This is why they have been implicated in the progression of many diseases, including cancer, rheumatoid arthritis, pre-eclampsia and asthma. In addition, these highly plastic cells react with phenotypic switches to minute changes in their environment. Thus a better understanding of the polarization capabilities of human macrophages could further our understanding of their de-regulation in diseases and provide better therapeutic approaches.

In this thesis, we differentiated human monocytes in a variety of conditions to macrophages. In order to gain a global picture of the resulting phenotypes, we analysed their transcriptome using RNA-seq and their accessible chromatin landscape with ATAC-seq. From all analysed conditions, collagen-treatment had the strongest effect on macrophage differentiation, which was characterized by many gene expression changes. The cells highly expressed genes associated with the complement cascade, endocytosis and chemotaxis, whereas they down-regulated genes involved in the inflammatory response and extracellular matrix organization, including collagens, integrins and matrix-metalloproteinases. In order to shed light on the regulatory mechanisms behind the described gene expression patterns, we analysed the accessible chromatin landscape of collagen-macrophages in comparison to plastic-macrophages. A *de novo* motif analysis revealed an enrichment for ETS, ETS:IRF, NFAT and Maf motifs in collagen-macrophages, additionally, the cells highly expressed *ETS2, MAF, IRF4* and *NFATC2*, indicating an involvement of these factors in mediating gene expression changes. Unfortunately, using ATAC footprints over the enriched motifs and ChIP-seq analysis for c-MAF, we could not identify, which transcription factor could be responsible for mediating collagen-induced gene expression changes. Possibly, a combination of several factors, their rearrangement on the chromatin and their post-transcriptional activation is involved, but this hypothesis needs to be tested further. Not only the up-, but also the down-regulation of transcription factors could provide an explanation for altered gene expression patterns. Since Dec2 and EGR2 were not expressed in collagen-macrophages, we further analysed their binding patterns. Their motifs were enriched in plastic-macrophage specific open chromatin and ChIP-seq data showed an enrichment of EGR2 and Dec2 binding in plastic-macrophage open chromatin regions, further pointing to an involvement of these factors during macrophage differentiation. To clarify their role, we performed siRNA mediated knock down experiments for both factors in monocytes, which were then differentiated on plastic. The removal of Dec2 had only a small impact on macrophage polarization, whereas the knock down of EGR2 led to a loss of gene expression in many genes also repressed by collagen treatment. The open chromatin analyses revealed that collagen-treatment and EGR2 knock down led to reduced chromatin accessibility in the same genomic regions, which were furthermore enriched for the EGR2 motif. This data indicates, that EGR2 expression during monocyte

to macrophage differentiation may be required for chromatin remodelling and successful expression of macrophage specific genes.

In the second part of this work, the transcriptome and accessible chromatin landscape of *ex vivo* isolated, CD163 sorted Hofbauer cells were analysed. We could show that the cells expressed many growth factors, whereas they down-regulated genes associated with cytokine production and viral processes, thereby, we confirmed the reported anti-inflammatory, tissue-remodelling phenotype of the cells. Furthermore, both Hofbauer cells and collagen-macrophages expressed genes involved in the complement cascade (*C1QA, C1QC*) and CD163, suggesting that the interaction of tissue macrophages with collagen influences the gene expression patterns of Hofbauer cells *in vivo*. In order to analyse the regulatory mechanisms behind the gene expression patterns, we analysed the accessible chromatin landscape of the cells. Hofbauer cell specific open chromatin regions, in comparison to *in vitro* generated macrophages, were enriched for MAF and nuclear receptor NR4A1 motifs. Additionally, Hofbauer cells highly expressed NR4A1, suggesting that this factor could be involved in regulating their gene expression. Interestingly, collagen-treatment also induced MAF and collagen-specific open chromatin regions were enriched for the same MAF motif, indicating that this transcription factor could be regulating commonly expressed genes.

6.2 Zusammenfassung

Makrophagen sind ein essentieller Teil der angeborenen Immunabwehr. Hierbei sind sie verantwortlich für die Erkennung und Phagozytose von Bakterien. Zusätzlich führen Gewebemakrophagen viele homöostatische Funktionen aus, die vom jeweiligen Gewebe abhängen. Dadurch sind sie häufig involviert in der Entstehung und Progression von Krankheiten, wie zum Beispiel in Tumoren, Arthritis, Präeklampsie und Asthma. Außerdem können Makrophagen durch die Vielzahl an exprimierten Rezeptoren auf ihrer Oberfläche, kleinste Veränderungen in ihrer Umgebung wahrnehmen und darauf mit einem veränderten Phänotyp reagieren. Eine Untersuchung der möglichen Veränderungen der Zellen könnte helfen, ihren Beitrag zu verschiedenen Krankheiten besser zu verstehen und neue therapeutische Möglichkeiten zu eröffnen.

Dementsprechend wurden in dieser Arbeit Monozyten aus dem peripħären Blut von gesunden Spendern in verschiedenen Konditionen zu Makrophagen differenziert. Um die resultierenden Makrophagen eingehend zu studieren wurde ihr Transkriptom mittels RNA Sequenzierung und die offene Chromatinstruktur mit ATAC-seq analysiert. Von allen analysierten Konditionen, hatte Kollagen den größten Einfluss auf die Zellen. Es induzierte Gene, die mit der Komplement Kaskade, Chemotaxis und der Endozytose assoziiert sind, wohingegen es inflammatorische Gene und Gene der Extrazellulären Matrix inhibierte. Um mehr über die Regulatoren, die hinter diesen Unterschieden in der Genexpression stehen, zu erfahren, haben wir das offene Chromatin der Zellen analysiert. Kollagen-spezifische zugängliche Regionen waren angereichert für ETS, ETS:IRF, NFAT und MAF Motive. Da Kollagen-Makrophagen eine signifikant erhöhte Expression von MAF, IRF4, ETS2 und NFATC2 aufwiesen, wurden diese als mögliche Kandidaten weiter analysiert. Mit Hilfe von ChIP-seq Daten für c-MAF und so genannten Footprint Analysen der restlichen Motive, konnten wir leider keinen der Faktoren als Verantwortlichen für die Genexpressionsunterschiede identifizieren. Allerdings könnten nicht nur induzierte, sondern auch inhibierte Faktoren für die Regulation der Genexpression verantwortlich sein. Zwei Transkriptionsfaktoren, Dec2 und EGR2, kämen dafür in Frage, da sie in Kollagen-Makrophagen nicht exprimiert wurden. Da die Bindungssequenzen der Faktoren im Standardmakrophagen spezifisch offenen Chromatin angereichert waren, analysierten wir die Bindungseigenschaften der beiden Faktoren mittels ChIP-seq. Dabei konnten wir zeigen, das sie diese Regionen im Chromatin binden und somit in der Regulation von Genen während der Differenzierung eine Rolle spielen könnten. Um die Auswirkung der reduzierten Expression dieser zuletzt genannten Transkriptionsfaktoren weiter zu untersuchen, wurden Knock-Down Experimente für beide in humanen Monozyten durchgeführt. Reduktion des Dec-2 Levels zeigte lediglich einen geringeren Effekt auf den Phänotyp von Makrophagen, wohingegen der Verlust von EGR2 einen stärkeren Einfluss zeigte. EGR2 knock down Makrophagen wiesen eine ähnliche Morphologie wie Kollagen-Makrophagen auf. Des weiteren zeigten sie ebenfalls reduzierte Expression von Makrophagen spezifischen Genen, zum Beispiel CSF1 und CHI3L1. Daraus lässt sich schließen, dass EGR2 während der Differenzierung notwendig ist um Makrophagen spezifische Gene zu aktivieren. In der Analyse des aktiven Chromatins nach EGR2 knock down fanden wir ebenfalls eine hohe Ähnlichkeit zwischen den mit Kollagen behandelten Makrophagen und den EGR2 Knock Down Zellen. Im Vergleich mit Plastik-Makrophagen zeigten beide eine verringerte Zugänglichkeit in speziellen Chromatin Regionen. Dies deutet darauf hin, dass die Induktion von EGR2, durch die Adhäsion der Zellen auf Plastik, notwendig ist für eine Öffnung des Chromatins und der Expression vieler Makrophagen spezifischer Gene. Im zweiten Teil dieser Arbeit wurde das Transkriptom und die Chromatinlandschaft von CD163 positiv sortierten Hofbauer Zellen analysiert. Mittels RNA Sequenzierung konnten wir zeigen, dass sie eine erhöhte Expression von Wachstumsfaktoren, zum Beispiel VEGFA oder M-CSF, aufweisen. Im Gegensatz dazu inhibieren Hofbauer Zellen Gene, die mit Immunabwehr und der viralen Abwehr zusammenhängen. Interessanterweise exprimierten Kollagen-Makrophagen und Hofbauer Zellen zum Teil ähnliche Gene, diese umfassten unter anderem den Komplement C1q Komplex und CD163, was darauf hinweist das die Interaktion mit Kollagen auch in residenten Gewebemakrophagen eine Rolle spielt. Um Kandidaten für die Regulation der Genexpression zu entdecken, analysierten wir die Motiv Signatur des offenen Chromatins in Hofbauer Zellen, wobei wir eine Anreicherung von ETS, bZIP, MAF und NR4A1 Motiven entdeckten. Da die Transkriptionsfaktoren MAF und NR4A1 in Hofbauer Zellen stark exprimiert waren, könnten sie verantwortlich sein für die beschriebenen Expressionsmuster. Zusammenfassend konnten wir feststellen, dass Hofbauer Zellen essentiell für die korrekte Entwicklung der Plazenta sind. Dabei weisen sie ein anti-inflammatorischen Phänotyp auf, der für die Toleranz der Mutter zum Fötus eine wichtige Rolle spielt. Darüberhinaus exprimieren sie aber auch viele Rezeptoren, welche verantwortlich für die Erkennung von Pathogenen sind. Dies deutet darauf hin, dass diese Makrophagen auch in der Lage sind eine inflammatorische Reaktion einzuleiten. Des weiteren haben wir zwei Transkriptionsfaktoren, MAF und NR4A1, gefunden, die möglicherweise für die Regulation der Genexpression verantwortlich sein könnten.

Chapter 7

Bibliography

Bibliography

[1] Jordi Bella. Collagen structure: new tricks from a very old dog. *Biochemical Journal*, 473(8):1001–1025, 2016.

[2] Michael Leiss, Karsten Beckmann, Amparo Girós, Mercedes Costell, and Reinhard Fässler. The role of integrin binding sites in fibronectin matrix assembly in vivo. *Current Opinion in Cell Biology*, 20(5):502 – 507, 2008. Cell-to-cell contact and extracellular matrix.

[3] Carla Sawan, Thomas Vaissière, Rabih Murr, and Zdenko Herceg. Epigenetic drivers and genetic passengers on the road to cancer. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 642(1):1 – 13, 2008.

[4] Slava Epelman, Kory J. Lavine, and Gwendalyn J Randolph. Origin and functions of tissue macrophages. *Immunity*, 41:21–35, 2014.

[5] Abul K. Abbas, Andrew H. Lichtman, and Pillai Shiv. *Cellular and Molecular Immunology*. Elsevier Saunders, 8 edition, 2015.

[6] Bernard Burke and Claire E. Lewis. *The Macrophage*. Oxford University Press, 2 edition, 2002.

[7] R van Furth, Z A Cohn, J G Hirsch, J H Humphrey, W G Spector, and H L Langevoort. The mononuclear phagocyte system: a new classification of macrophages, monocytes, and their precursor cells. *Bull World Health Organ*, 46:845–852, 1972.

[8] A Volkman, NC Chang, PH Strausbauch, and PS Morahan. Differential effects of chronic monocyte depletion on macrophage populations. *Laboratory investigation; a journal of technical methods and pathology*, 49(3):291—298, 1983.

[9] Florent Ginhoux, Melanie Greter, Marylene Leboeuf, Sayan Nandi, Peter See, Solen Gokhan, Mark F. Mehler, Simon J. Conway, Lai Guan Ng, E. Richard Stanley, Igor M. Samokhvalov, and Miriam Merad. Fate mapping analysis reveals that adult microglia derive from primitive macrophages. *Science*, 330(6005):841–845, 2010.

[10] Christian Schulz, Elisa Gomez Perdiguero, Laurent Chorro, Heather Szabo-Rogers, Nicolas Cagnard, Katrin Kierdorf, Marco Prinz, Bishan Wu, Sten Eirik W. Jacobsen, Jeffrey W. Pollard, Jon Frampton, Karen J. Liu, and Frederic Geissmann. A

lineage of myeloid cells independent of myb and hematopoietic stem cells. *Science*, 336(6077):86–90, 2012.

[11] Simon Yona, Ki-Wook Kim, Yochai Wolf, Alexander Mildner, Diana Varol, Michal Breker, Dalit Strauss-Ayali, Sergey Viukov, Martin Guilliams, Alexander Misharin, David A. Hume, Harris Perlman, Bernard Malissen, Elazar Zelzer, and Steffen Jung. Fate mapping reveals origins and dynamics of monocytes and tissue macrophages under homeostasis. *Immunity*, 38:79–91, 2013.

[12] Daigo Hashimoto, Andrew Chow, Clara Noizat, Pearline Teo, Mary Beth Beasley, Marylene Leboeuf, Christian D. Becker, Peter See, Jeremy Price, Daniel Lucas, Melanie Greter, Arthur Mortha, Scott W. Boyer, E. Camilla Forsberg, Masato Tanaka, Nico van Rooijen, Adolfo García-Sastre, E. Richard Stanley, Florent Ginhoux, Paul S. Frenette, and Miriam Merad. Tissue-resident macrophages self-maintain locally throughout adult life with minimal contribution from circulating monocytes. *Immunity*, 38:792–804, 2013.

[13] Slava Epelman, Kory J. Lavine, Anna E. Beaudin, Dorothy K. Sojka, Javier A. Carrero, Boris Calderon, Thaddeus Brija, Emmanuel L. Gautier, Stoyan Ivanov, Ansuman T. Satpathy, Joel D. Schilling, Reto Schwendener, Ismail Sergin, Babak Razani, E. Camilla Forsberg, Wayne M. Yokoyama, Emil R. Unanue, Marco Colonna, Gwendalyn J. Randolph, and Douglas L. Mann. Embryonic and adult-derived resident cardiac macrophages are maintained through distinct mechanisms at steady state and during inflammation. *Immunity*, 40:91–104, 2014.

[14] Martin Guilliams, Ismé De Kleer, Sandrine Henri, Sijranke Post, Leen Vanhoutte, Sofie De Prijck, Kim Deswarte, Bernard Malissen, Hamida Hammad, and Bart N. Lambrecht. Alveolar macrophages develop from fetal monocytes that differentiate into long-lived cells in the first week of life via gm-csf. *Journal of Experimental Medicine*, 210(10):1977–1992, 2013.

[15] Claudia Jakubzick, Emmanuel L. Gautier, Sophie L. Gibbings, Dorothy K. Sojka, Andreas Schlitzer, Theodore E. Johnson, Stoyan Ivanov, Qiaonan Duan, Shashi Bala, Tracy Condon, Nico van Rooijen, John R. Grainger, Yasmine Belkaid, Avi Ma ayan, David W.H. Riches, Wayne M. Yokoyama, Florent Ginhoux, Peter M. Henson, and Gwendalyn J. Randolph. Minimal differentiation of classical monocytes as they survey steady-state tissues and transport antigen to lymph nodes. *Immunity*, 39:599–610, 2013.

[16] Katrin Kierdorf, Daniel Erny, Tobias Goldmann, Victor Sander, Christian Schulz, Elisa Gomez Perdiguero, Peter Wieghofer, Annette Heinrich, Pia Riemke, Christoph Hölscher, Dominik N Müller, Bruno Luckow, Thomas Brocker, Katharina Debowski, Günter Fritz, Ghislain Opdenakker, Andreas Diefenbach, Knut Biber, Mathias Heikenwalder, Frederic Geissmann, Frank Rosenbauer, and Marco Prinz. Microglia emerge from erythromyeloid precursors via pu.1- and irf8-dependent pathways. *Nature Neuroscience*, 16, 2013.

[17] Lianne van de Laar, Wouter Saelens, Sofie Prijck, Liesbet Martens, Charlotte L. Scott, Gert Van Isterdael, Eik Hoffmann, Rudi Beyaert, Yvan Saeys, Bart N. Lambrecht, and Martin Guilliams. Yolk sac macrophages, fetal liver, and adult monocytes can colonize an empty niche and develop into functional tissue-resident macrophages. *Immunity*, 44:755–768, 2016.

[18] Sophie L. Gibbings, Rajni Goyal, A. Nicole Desch, Sonia M. Leach, Miglena Prabagar, Shaikh M. Atif, Donna L. Bratton, William Janssen, and Claudia V. Jakubzick. Transcriptome analysis highlights the conserved difference between embryonic and postnatal-derived alveolar macrophages. *Blood*, 126(11):1357–1366, 2015.

[19] Charlotte L. Scott, Fang Zheng, Patrick De Baetselier, Liesbet Martens, Yvan Saeys, Sofie De Prijck, Saskia Lippens, Chloé Abels, Steve Schoonooghe, Geert Raes, Nick Devoogdt, Bart N. Lambrecht, Alain Beschin, and Martin Guilliams. Bone marrow-derived monocytes give rise to self-renewing and fully differentiated kupffer cells. *Nature Communications*, 7, 2016. http://dx.doi.org/10.1038/ncomms10321.

[20] S Jolles. Paul langerhans. *Journal of clinical pathology*, 55(4):243–243, 04 2002.

[21] Daisuke Kurotaki, Haruka Sasaki, and Tomohiko Tamura. Transcriptional control of monocyte and macrophage development. *International Immunology*, 29(3):97–107, 04 2017.

[22] Bain Calum C. and Mowat Allan McI. Intestinal macrophages – specialised adaptation to a unique environment. *European Journal of Immunology*, 41(9):2494–2498, 2011.

[23] Natalio Garbi and Bart N. Lambrecht. Location, function, and ontogeny of pulmonary macrophages during the steady state. *Pflügers Archiv - European Journal of Physiology*, 469(3):561–572, Apr 2017.

[24] David Gosselin, Dylan Skola, Nicole G. Coufal, Inge R. Holtman, Johannes C. M. Schlachetzki, Eniko Sajti, Baptiste N. Jaeger, Carolyn O'Connor, Conor Fitzpatrick, Martina P. Pasillas, Monique Pena, Amy Adair, David D. Gonda, Michael L. Levy, Richard M. Ransohoff, Fred H. Gage, and Christopher K. Glass. An environment-dependent transcriptional network specifies human microglia identity. *Science*, 356(6344), 2017.

[25] Natalie Levtova, Luke M. Healy, Catalina Marysol Carvajal Gonczi, Brandon Stopnicki, Manon Blain, Timothy E. Kennedy, Craig S. Moore, Jack P. Antel, and Peter J. Darlington. Comparative morphology and phagocytic capacity of primary human adult microglia with time-lapse imaging. *Journal of Neuroimmunology*, 310:143 – 149, 2017.

[26] Yonit Lavin, Deborah Winter, Ronnie Blecher-Gonen, Eyal David, Hadas Keren-Shaul, Miriam Merad, Steffen Jung, and Ido Amit. Tissue-resident macrophage

enhancer landscapes are shaped by the local microenvironment. *Cell*, 159(6):1312 – 1326, 2014.

[27] Masako Kohyama, Wataru Ise, Brian T. Edelson, Peter R. Wilker, Kai Hildner, Carlo Mejia, William A. Frazier, Theresa L. Murphy, and Kenneth M. Murphy. Role for spi-c in the development of red pulp macrophages and splenic iron homeostasis. *Nature*, 457, 2008.

[28] Anne Buttgereit, Iva Lelios, Xueyang Yu, Melissa Vrohlings, Natalie R Krakoski, Emmanuel L Gautier, Ryuichi Nishinakamura, Burkhard Becher, and Melanie Greter. Sall1 is a transcriptional regulator defining microglia identity and function. *Nature Immunology*, 17, 2016.

[29] Marcela Rosas, Luke C. Davies, Peter J. Giles, Chia-Te Liao, Bashar Kharfan, Timothy C. Stone, Valerie B. O'Donnell, Donald J. Fraser, Simon A. Jones, and Philip R. Taylor. The transcription factor gata6 links tissue macrophage phenotype and proliferative renewal. *Science*, 344(6184):645–648, 2014.

[30] Yasutaka Okabe and Ruslan Medzhitov. Tissue-specific signals control reversible program of localization and functional polarization of macrophages. *Cell*, 157:832–844, 2014.

[31] Antonio José Ruiz-Alcaraz, Violeta Carmona-Martínez, María Tristán-Manzano, Francisco Machado-Linde, María Luisa Sánchez-Ferrer, Pilar García-Peñarrubia, and María Martínez-Esparza. Characterization of human peritoneal monocyte/-macrophage subsets in homeostasis: Phenotype, gata6, phagocytic/oxidative activities and cytokines expression. *Scientific Reports*, 8(1):12794, 2018.

[32] Christoph Schneider, Samuel P. Nobs, Michael Kurrer, Hubert Rehrauer, Christoph Thiele, and Manfred Kopf. Induction of the nuclear receptor ppar-γ by the cytokine gm-csf is critical for the differentiation of fetal monocytes into alveolar macrophages. *Nature Immunology*, 15, 2014.

[33] Xueyang Yu, Anne Buttgereit, Iva Lelios, Sebastian G. Utz, Dilay Cansever, Burkhard Becher, and Melanie Greter. The cytokine tgf-β promotes the development and homeostasis of alveolar macrophages. *Immunity*, 47(5):903 – 912.e4, 2017.

[34] A Schridde, C C Bain, J U Mayer, J Montgomery, E Pollet, B Denecke, S W F Milling, S J Jenkins, M Dalod, S Henri, B Malissen, O Pabst, and A Mcl Mowat. Tissue-specific differentiation of colonic macrophages requires tgfβ receptor-mediated signaling. *Mucosal Immunology*, 10, 2017.

[35] Neil M. Gude, Claire T. Roberts, Bill Kalionis, and Roger G. King. Growth and function of the normal human placenta. *Thrombosis Research*, 114(5):397 – 407, 2004. Special issue - State-of-the-Art 11th International Congress on Antiphospholipid Antibodies.

[36] Meghan E Bushway, Scott A Gerber, Bruce M Fenton, Richard K Miller, Edith M Lord, and Shawn P Murphy. Morphological and phenotypic analyses of the human placenta using whole mount immunofluorescence. *Biology of Reproduction*, 90, 2014.

[37] Mi Jeong Kim, Roberto Romero, Chong Jai Kim, Adi L. Tarca, Sovantha Chhauy, Christopher LaJeunesse, Deug-Chan Lee, Sorin Draghici, Francesca Gotsch, Juan Pedro Kusanovic, Sonia S. Hassan, and Jung-Sun Kim. Villitis of unknown etiology is associated with a distinct pattern of chemokine up-regulation in the feto-maternal and placental compartments: Implications for conjoint maternal allograft rejection and maternal anti-fetal graft-versus-host disease. *The Journal of Immunology*, 182(6):3919–3927, 2009.

[38] S. Kim, J., R Romero, R. Kim, M., Y. M. Kim, L Friel, J Espinoza, and C. J. Kim. Involvement of hofbauer cells and maternal t cells in villitis of unknown aetiology. *Histopathology*, 52(4):457–464, 2018.

[39] Zhonghua Tang, Serkalem Tadesse, Errol Norwitz, Gil Mor, Vikki M. Abrahams, and Seth Guller. Isolation of hofbauer cells from human term placentas with high yield and purity. *American Journal of Reproductive Immunology (New York, N.Y.: 1989)*, 66(4):336–348, 2011.

[40] M. Castellucci, A. Celona, H. Bartels, B. Steininger, V. Benedetto, and P. Kaufmann. Mitosis of the hofbauer cell: Possible implications for a fetal macrophage. *Placenta*, 8(1):65 – 76, 1987.

[41] Y. Seval, E.T. Korgun, and R. Demir. Hofbauer cells in early human placenta: Possible implications in vasculogenesis and angiogenesis. *Placenta*, 28(8):841 – 845, 2007.

[42] E.Y. Anteby, S. Natanson-Yaron, C. Greenfield, D. Goldman-Wohl, R. Haimov-Kochman, H. Holzer, and S. Yagel. Human placental hofbauer cells express sprouty proteins: a possible modulating mechanism of villous branching. *Placenta*, 26(6):476 – 483, 2005.

[43] J Loegl, U Hiden, E Nussbaumer, C Schliefsteiner, S Cvitic, I Lang, C Wadsack, B Huppertz, and G Desoye. Hofbauer cells of m2a, m2b and m2c polarization may regulate feto-placental angiogenesis. *Reproduction*, 152(5):447–455, 2016.

[44] Sun Young Kim, Roberto Romero, Adi L. Tarca, Gaurav Bhatti, Chong Jai Kim, JoonHo Lee, Amelia Elsey, Nandor Gabor Than, Tinnakorn Chaiworapongsa, Sonia S. Hassan, Gyeong Hoon Kang, and Jung-Sun Kim. Methylome of fetal and maternal monocytes and macrophages at the feto-maternal interface. *American Journal of Reproductive Immunology*, 68(1):8–27, 2012.

[45] Omar M. Young, Zhonghua Tang, Tracy Niven-Fairchild, Serkalem Tadesse, Graciela Krikun, Errol R. Norwitz, Gil Mor, Vikki M. Abrahams, and Seth Guller. Toll-like

receptor-mediated responses by placental hofbauer cells (hbcs): A potential proinflammatory role for fetal m2 macrophages. *American Journal of Reproductive Immunology*, 73(1):22–35, 2014.

[46] Seung Woo Yang, Eun Hee Cho, So Young Choi, Yun Kyung Lee, Jae Hyun Park, Min Kyung Kim, Jin Yeon Park, Hyeong Jwa Choi, Jeong Ik Lee, Hyun Myung Ko, Seung Hwa Park, Han Sung Hwang, and Young Sun Kang. Dc-sign expression in hofbauer cells may play an important role in immune tolerance in fetal chorionic villi during the development of preeclampsia. *Journal of Reproductive Immunology*, 124:30 – 37, 2017.

[47] Sergio Romagnani. Th1/Th2 Cells. *Inflammatory Bowel Diseases*, 5(4):285–294, 11 1999.

[48] Charles D. Mills, Kristi Kincaid, Jennifer M. Alt, Michelle J. Heilman, and Annette M. Hill. M-1/m-2 macrophages and the th1/th2 paradigm. *The Journal of Immunology*, 164(12):6166–6173, 2000.

[49] Alberto Mantovani, Subhra K Biswas, Maria Rosaria Galdiero, Antonio Sica, and Massimo Locati. Macrophage plasticity and polarization in tissue repair and remodelling. *The Journal of Pathology*, 229(2):176–185, 2013.

[50] Daphne Y.S. Vogel, Judith E. Glim, Andrea W.D. Stavenuiter, Marjolein Breur, Priscilla Heijnen, Sandra Amor, Christine D. Dijkstra, and Robert H.J. Beelen. Human macrophage polarization in vitro: Maturation and activation methods compared. *Immunobiology*, 219(9):695 – 703, 2014.

[51] Derek C. Lacey, Adrian Achuthan, Andrew J. Fleetwood, Hang Dinh, John Roiniotis, Glen M. Scholz, Melody W. Chang, Sandra K. Beckman, Andrew D. Cook, and John A. Hamilton. Defining gm-csf– and macrophage-csf–dependent macrophage responses by in vitro models. *The Journal of Immunology*, 188(11):5752–5765, 2012.

[52] Giulia Chinetti-Gbaguidi and Bart Staels. Macrophage polarization in metabolic disorders: functions and regulation. *Current Opinion in Lipidology*, 22, 2011.

[53] Alberto Mantovani, Silvano Sozzani, Massimo Locati, Paola Allavena, and Antonio Sica. Macrophage polarization: tumor-associated macrophages as a paradigm for polarized m2 mononuclear phagocytes. *Trends in Immunology*, 23(11):549 – 555, 2002.

[54] Silke Reinartz, Tim Schumann, Florian Finkernagel, Annika Wortmann, Julia M. Jansen, Wolfgang Meissner, Michael Krause, Anne-Marie Schwörer, Uwe Wagner, Sabine Müller-Brüsselbach, and Rolf Müller. Mixed-polarization phenotype of ascites-associated macrophages in human ovarian carcinoma: Correlation of cd163 expression, cytokine levels and early relapse. *International Journal of Cancer*, 134(1):32–42, 2013.

[55] Fernando O. Martinez, Siamon Gordon, Massimo Locati, and Alberto Mantovani. Transcriptional profiling of the human monocyte-to-macrophage differentiation and polarization: New molecules and patterns of gene expression. *The Journal of Immunology*, 177(10):7303–7311, 2006.

[56] Gerard J. Nau, Joan F. L. Richmond, Ann Schlesinger, Ezra G. Jennings, Eric S. Lander, and Richard A. Young. Human macrophage activation programs induced by bacterial pathogens. *Proceedings of the National Academy of Sciences*, 99(3):1503–1508, 2002.

[57] Stephen A. Ramsey, Sandy L. Klemm, Daniel E. Zak, Kathleen A. Kennedy, Vesteinn Thorsson, Bin Li, Mark Gilchrist, Elizabeth S. Gold, Carrie D. Johnson, Vladimir Litvak, Garnet Navarro, Jared C. Roach, Carrie M. Rosenberger, Alistair G. Rust, Natalya Yudkovsky, Alan Aderem, and Ilya Shmulevich. Uncovering a macrophage transcriptional program by integrating evidence from motif scanning and expression dynamics. *PLOS Computational Biology*, 4(3):1–25, 03 2008.

[58] David M. Mosser and Justin P. Edwards. Exploring the full spectrum of macrophage activation. *Nature Reviews Immunology*, 8:958 EP –, 12 2008.

[59] Jia Xue, Susanne V. Schmidt, Jil Sander, Astrid Draffehn, Wolfgang Krebs, Inga Quester, Dominic De Nardo, Trupti D. Gohel, Martina Emde, Lisa Schmidleithner, Hariharasudan Ganesan, Andrea Nino-Castro, Michael R. Mallmann, Larisa Labzin, Heidi Theis, Michael Kraut, Marc Beyer, Eicke Latz, Tom C. Freeman, Thomas Ulas, and Joachim L. Schultze. Transcriptome-based network analysis reveals a spectrum model of human macrophage activation. *Immunity*, 40(2):274 – 288, 2014.

[60] Luca Magnani, Jerome Eeckhoute, and Mathieu Lupien. Pioneer factors: directing transcriptional regulators within the chromatin environment. *Trends in Genetics*, 27(11):465 – 474, 2011.

[61] Kenneth S. Zaret and Jason S. Carroll. Pioneer transcription factors: establishing competence for gene expression. *Genes & Development*, 25(21):2227–2241, 2011.

[62] McKercher S. R., Torbett B. E., Anderson K. L., Henkel G. W., Vestal D. J., Baribault H., Klemsz M., Feeney A. J., Wu G. E., Paige C. J., and Maki R. A. Targeted disruption of the pu.1 gene results in multiple hematopoietic abnormalities. *The EMBO Journal*, 15(20):5647–5658, 1996.

[63] Jonathan Back, David Allman, Susan Chan, and Philippe Kastner. Visualizing pu.1 activity during hematopoiesis. *Experimental Hematology*, 33(4):395 – 402, 2005.

[64] Stephen L. Nutt, Donald Metcalf, Angela D'Amico, Matthew Polli, and Li Wu. Dynamic regulation of pu.1 expression in multipotent hematopoietic progenitors. *Journal of Experimental Medicine*, 201(2):221–231, 2005.

[65] Michael J. Klemsz, Scott R. McKercher, Antonio Celada, Charles Van Beveren, and Richard A. Maki. The macrophage and b cell-specific transcription factor pu.1 is related to the ets oncogene. *Cell*, 61(1):113 – 124, 1990.

[66] Sven Heinz, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C. Lin, Peter Laslo, Jason X. Cheng, Cornelis Murre, Harinder Singh, and Christopher K. Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular Cell*, 38(4):576 – 589, 2010.

[67] David Gosselin, Verena M. Link, Casey E. Romanoski, Gregory J. Fonseca, Dawn Z. Eichenfield, Nathanael J. Spann, Joshua D. Stender, Hyun B. Chun, Hannah Garner, Frederic Geissmann, and Christopher K. Glass. Environment drives selection and function of enhancers controlling tissue-specific macrophage identities. *Cell*, 159(6):1327 – 1340, 2014.

[68] Swee Heng Milon Pang, Martina Minnich, Pradnya Gangatirkar, Zhiqiang Zheng, Anja Ebert, Guangchun Song, Ross A Dickins, Lynn M Corcoran, Charles G Mullighan, Meinrad Busslinger, Nicholas D Huntington, Stephen L Nutt, and Sebastian Carotta. Pu.1 cooperates with irf4 and irf8 to suppress pre-b cell leukemia. *Leukemia*, 30(6):1375–1387, 06 2016.

[69] Tomohiko Tamura, Pratima Thotakura, Tetsuya S. Tanaka, Minoru S. H. Ko, and Keiko Ozato. Identification of target genes and a unique cis element regulated by irf-8 in developing macrophages. *Blood*, 106(6):1938–1947, 2005.

[70] Carlos R. Escalante, Abraham L. Brass, Jagan M. R. Pongubala, Ella Shatova, Leyi Shen, Harinder Singh, and Aneel K. Aggarwal. Crystal structure of pu.1/irf-4/dna ternary complex. *Molecular Cell*, 10(5):1097–1105, 2019/02/14 2002.

[71] Michio Yamamoto, Takayuki Kato, Chie Hotta, Akira Nishiyama, Daisuke Kurotaki, Masahiro Yoshinari, Masamichi Takami, Motohide Ichino, Masatoshi Nakazawa, Toshifumi Matsuyama, Ryutaro Kamijo, Seiichi Kitagawa, Keiko Ozato, and Tomohiko Tamura. Shared and distinct functions of the transcription factors irf4 and irf8 in myeloid cell development. *PLOS ONE*, 6(10):1–10, 10 2011.

[72] Wolfgang Bauer, Martina Rauner, Michael Haase, Satu Kujawski, Laleh S. Arabanian, Ivonne Habermann, Lorenz C. Hofbauer, Gerhard Ehninger, and Alexander Kiani. Osteomyelosclerosis, anemia and extramedullary hematopoiesis in mice lacking the transcription factor nfatc2. *Haematologica*, 96(11):1580–1588, 2011.

[73] Fumiyo Ikeda, Riko Nishimura, Takuma Matsubara, Sakae Tanaka, Jun ichiro Inoue, Sakamuri V. Reddy, Kenji Hata, Kenji Yamashita, Toru Hiraga, Toshiyuki Watanabe, Toshio Kukita, Katsuji Yoshioka, Anjana Rao, and Toshiyuki Yoneda. Critical roles of c-jun signaling in regulation of nfat family and rankl-regulated osteoclast differentiation. *The Journal of Clinical Investigation*, 114(4):475–484, 8 2004.

[74] EK O'Shea, R Rutkowski, WF Stafford, and PS Kim. Preferential heterodimer formation by isolated leucine zippers from fos and jun. *Science*, 245(4918):646–648, 1989.

[75] E F Wagner. Functions of ap1 (fos/jun) in bone development. *Annals of the Rheumatic Diseases*, 61, 2002.

[76] D. Liebermann, B. Gregory, and B. Hoffman. Ap-1 (fos-jun) transcription factors in hematopoietic differentiation and apoptosis. *International Journal of Oncology*, 12:685–1385, 1998.

[77] K Kataoka, K T Fujiwara, M Noda, and M Nishizawa. Mafb, a new maf family transcription activator that can associate with maf and fos but not with jun. *Molecular and Cellular Biology*, 14(11):7581–7591, 1994.

[78] Louise M. Kelly, Ursula Englmeier, Isabelle Lafon, Michael H. Sieweke, and Thomas Graf. Mafb is an inducer of monocytic differentiation. *The EMBO Journal*, 19(9):1987–1997, 2000.

[79] Dhouha Daassi, Michito Hamada, Hyojung Jeon, Yuki Imamura, Mai Thi Nhu Tran, and Satoru Takahashi. Differential expression patterns of mafb and c-maf in macrophages in vivo and in vitro. *Biochemical and Biophysical Research Communications*, 473(1):118 – 124, 2016.

[80] Athar Aziz, Erinn Soucie, Sandrine Sarrazin, and Michael H. Sieweke. Mafb/c-maf deficiency enables self-renewal of differentiated functional macrophages. *Science*, 326(5954):867–871, 2009.

[81] Hirofumi Kurokawa, Hozumi Motohashi, Shinji Sueno, Momoko Kimura, Hiroaki Takagawa, Yousuke Kanno, Masayuki Yamamoto, and Toshiyuki Tanaka. Structural basis of alternative dna recognition by maf transcription factors. *Molecular and Cellular Biology*, 29(23):6232–6244, 12 2009.

[82] Hanna Krysinska, Maarten Hoogenkamp, Richard Ingram, Nicola Wilson, Hiromi Tagoh, Peter Laslo, Harinder Singh, and Constanze Bonifer. A two-step, pu.1-dependent mechanism for developmentally regulated chromatin remodeling and transcription of the c-fms gene. *Molecular and Cellular Biology*, 27(3):878–887, 2007.

[83] S Kharbanda, T Nakamura, R Stone, R Hass, S Bernstein, R Datta, V P Sukhatme, and D Kufe. Expression of the early growth response 1 and 2 zinc finger genes during induction of monocytic differentiation. *The Journal of Clinical Investigation*, 88(2):571–577, 8 1991.

[84] Hirano Seishiro, Anuradha Cunigaipur D., and Kanno Sanae. krox-20/egr-2 is upregulated following non-specific and homophilic adhesion in rat macrophages. *Immunology*, 107(1):86–92, 2002.

[85] R J Shaw, D E Doherty, A G Ritter, S H Benedict, and R A Clark. Adherence-dependent increase in human monocyte pdgf(b) mrna is associated with increases in c-fos, c-jun, and egr2 mrna. *The Journal of Cell Biology*, 111(5):2139–2148, 1990.

[86] Christian Frantz, Kathleen M. Stewart, and Valerie M. Weaver. The extracellular matrix at a glance. *Journal of Cell Science*, 123(24):4195–4200, 2010.

[87] Berisio Rita, Vitagliano Luigi, Mazzarella Lelio, and Zagari Adriana. Crystal structure of the collagen triple helix model [(pro-pro-gly)10]3. *Protein Science*, 11(2):262–270, 2009.

[88] Matthew D. Shoulders and Ronald T. Raines. Collagen structure and stability. *Annual Review of Biochemistry*, 78(1):929–958, 2009. PMID: 19344236.

[89] Jaroslava Halper and Michael Kjaer. *Basic Components of Connective Tissues and Extracellular Matrix: Elastin, Fibrillin, Fibulins, Fibrinogen, Fibronectin, Laminin, Tenascins and Thrombospondins*, pages 31–47. Springer Netherlands, Dordrecht, 2014.

[90] Alexandros Glentis, Vasily Gurchenkov, and Danijela Matic Vignjevic. Assembly, heterogeneity, and breaching of the basement membranes. *Cell Adhesion & Migration*, 8(3):236–245, 2014. PMID: 24727304.

[91] Henry M. Becker, Jacob Rullo, Mian Chen, Magar Ghazarian, Sungho Bak, Haiyan Xiao, John B. Hay, and Myron I. Cybulsky. $\alpha 1\beta 1$ integrin-mediated adhesion inhibits macrophage exit from a peripheral inflammatory lesion. *The Journal of Immunology*, 190(8):4305–4314, 2013.

[92] Luke R. Anderson, Thomas W. Owens, and Matthew J. Naylor. Structural and mechanical functions of integrins. *Biophysical Reviews*, 6(2):203–213, Jun 2014.

[93] Ambra Pozzi, Peter D. Yurchenco, and Renato V. Iozzo. The nature and biology of basement membranes. *Matrix Biology*, 57-58:1 – 11, 2017. Basement Membranes in Health and Disease.

[94] D E Hall, L F Reichardt, E Crowley, B Holley, H Moezzi, A Sonnenberg, and C H Damsky. The alpha 1/beta 1 and alpha 6/beta 1 integrin heterodimers mediate cell attachment to distinct sites on laminin. *The Journal of Cell Biology*, 110(6):2175–2184, 1990.

[95] Matti Lahti, Jyrki Heino, and Jarmo Käpylä. Leukocyte integrins αlβ2, αmβ2 and αxβ2 as collagen receptors - receptor activation and recognition of gfoger motif. *The FASEB Journal*, 29(1_supplement):LB167, 2015.

[96] Cédric Zeltz, Joseph Orgel, and Donald Gullberg. Molecular composition and function of integrin-based collagen glues—introducing colinbris. *Biochimica et Biophysica Acta (BBA) - General Subjects*, 1840(8):2533 – 2548, 2014. Matrix-mediated cell behaviour and properties.

[97] S L Newman and M A Tucci. Regulation of human monocyte/macrophage function by extracellular matrix. adherence of monocytes to collagen matrices enhances phagocytosis of opsonized bacteria by activation of complement receptors and enhancement of fc receptor function. *The Journal of Clinical Investigation*, 86(3):703–714, 9 1990.

[98] Spencer W. Galt, Stephan Lindemann, Don Medd, Loren L. Allen, Larry W. Kraiss, Estelle S. Harris, Stephen M. Prescott, Thomas M. McIntyre, Andrew S. Weyrich, and Guy A. Zimmerman. Differential regulation of matrix metalloproteinase-9 by monocytes adherent to collagen and platelets. *Circulation Research*, 89(6):509–516, 2001.

[99] Maryam G. Rohani, Ryan S. McMahan, Maria V. Razumova, Angie L. Hertz, Maryelise Cieslewicz, Suzie H. Pun, Michael Regnier, Ying Wang, Timothy P. Birkland, and William C. Parks. Mmp-10 regulates collagenolytic activity of alternatively activated resident macrophages. *Journal of Investigative Dermatology*, 135(10):2377 – 2384, 2015.

[100] Michael Schnoor, Paul Cullen, Julia Lorkowski, Katrin Stolle, Horst Robenek, David Troyer, Jürgen Rauterberg, and Stefan Lorkowski. Production of type vi collagen by human macrophages: A new dimension in macrophage functional heterogeneity. *The Journal of Immunology*, 180(8):5707–5719, 2008.

[101] Johanna A. Knipper, Sebastian Willenborg, Jürgen Brinckmann, Wilhelm Bloch, Tobias Maaß, Raimund Wagener, Thomas Krieg, Tara Sutherland, Ariel Munitz, Marc E. Rothenberg, Anja Niehoff, Rebecca Richardson, Matthias Hammerschmidt, Judith E. Allen, and Sabine A. Eming. Interleukin-4 receptor α signaling in myeloid cells controls collagen fibril assembly in skin repair. *Immunity*, 43(4):803 – 816, 2015.

[102] Y Tsukamoto, W E Helsel, and S M Wahl. Macrophage production of fibronectin, a chemoattractant for fibroblasts. *The Journal of Immunology*, 127(2):673–678, 1981.

[103] Bianco Celso. Fibrin, fibronectin, and macrophages. *Annals of the New York Academy of Sciences*, 408(1):602–609, 1983.

[104] P. R. Sudhakaran, A. Radhika, and S. S. Jacob. Monocyte macrophage differentiation in vitro: Fibronectin-dependent upregulation of certain macrophage-specific activities. *Glycoconjugate Journal*, 24(1):49–55, 01 2007.

[105] D A Norris, R A Clark, L M Swigart, J C Huff, W L Weston, and S E Howell. Fibronectin fragment(s) are chemotactic for human peripheral blood monocytes. *The Journal of Immunology*, 129(4):1612–1618, 1982.

[106] Karine Feghali and Daniel Grenier. Priming effect of fibronectin fragments on the macrophage inflammatory response: Potential contribution to periodontitis. *Inflammation*, 35(5):1696–1705, 10 2012.

[107] Atay Safinur, Gercel-Taylor Cicek, and Taylor Douglas D. Human trophoblast-derived exosomal fibronectin induces pro-inflammatory il-1β production by macrophages. *American Journal of Reproductive Immunology*, 66(4):259–269, 2011.

[108] C. H. Waddington. The epigenotype. *International Journal of Epidemiology*, 41(1):10–13, 2012.

[109] Timothy H. Bestor. The DNA methyltransferases of mammals. *Human Molecular Genetics*, 9(16):2395–2402, 10 2000.

[110] James G. Herman and Stephen B. Baylin. Gene silencing in cancer in association with promoter hypermethylation. *New England Journal of Medicine*, 349(21):2042–2054, 2003. PMID: 14627790.

[111] Maja Klug, Sandra Schmidhofer, Claudia Gebhard, Reinhard Andreesen, and Michael Rehli. 5-hydroxymethylcytosine is an essential intermediate of active dna demethylation processes in primary human monocytes. *Genome Biology*, 14(5):R46, 2013.

[112] Hong Ji, Lauren I. R. Ehrlich, Jun Seita, Peter Murakami, Akiko Doi, Paul Lindau, Hwajin Lee, Martin J. Aryee, Rafael A. Irizarry, Kitai Kim, Derrick J. Rossi, Matthew A. Inlay, Thomas Serwold, Holger Karsunky, Lena Ho, George Q. Daley, Irving L. Weissman, and Andrew P. Feinberg. Comprehensive methylome map of lineage commitment from haematopoietic progenitors. *Nature*, 467:338 EP –, 08 2010.

[113] Bryan M. Turner. *The Nucleosome: Chromatin's Structural Unit*, chapter 3, pages 44–58. Wiley-Blackwell, 2007.

[114] *Histone Tails: Modifications and Epigenetic Information*, chapter 4, pages 59–74. Wiley-Blackwell, 2007.

[115] Antti P Aalto and Amy E Pasquinelli. Small non-coding rnas mount a silent revolution in gene expression. *Current Opinion in Cell Biology*, 24(3):333 – 340, 2012. Nucleus and gene expression.

[116] Dominik J. Jedlinski, Hadi Jorjani, Rafal Gumienny, Andreas R. Gruber, Mihaela Zavolan, Jana Hertel, Stephanie Kehr, and Peter F. Stadler. An updated human snoRNAome. *Nucleic Acids Research*, 44(11):5068–5082, 05 2016.

[117] Tae-Kyung Kim, Martin Hemberg, Jesse M. Gray, Allen M. Costa, Daniel M. Bear, Jing Wu, David A. Harmin, Mike Laptewicz, Kellie Barbara-Haley, Scott Kuersten, Eirene Markenscoff-Papadimitriou, Dietmar Kuhl, Haruhiko Bito, Paul F. Worley, Gabriel Kreiman, and Michael E. Greenberg. Widespread transcription at neuronal activity-regulated enhancers. *Nature*, 465:182 EP –, 04 2010.

[118] Katsutomo Okamura, Wei-Jen Chung, J. Graham Ruby, Huili Guo, David P. Bartel, and Eric C. Lai. The drosophila hairpin rna pathway generates endogenous short interfering rnas. *Nature*, 453:803 EP –, 05 2008.

[119] Maartje J. Luteijn and RenéF. Ketting. Piwi-interacting rnas: from generation to transgenerational epigenetics. *Nature Reviews Genetics*, 14:523 EP –, 06 2013.

[120] Menno P. Creyghton, Albert W. Cheng, G. Grant Welstead, Tristan Kooistra, Bryce W. Carey, Eveline J. Steine, Jacob Hanna, Michael A. Lodato, Garrett M. Frampton, Phillip A. Sharp, Laurie A. Boyer, Richard A. Young, and Rudolf Jaenisch. Histone h3k27ac separates active from poised enhancers and predicts developmental state. *Proceedings of the National Academy of Sciences*, 107(50):21931–21936, 2010.

[121] Lionel B. Ivashkiv. Epigenetic regulation of macrophage polarization and function. *Trends in Immunology*, 34(5):216 – 223, 2013.

[122] Taeko Wada, Jiro Kikuchi, Noriko Nishimura, Rumi Shimizu, Toshio Kitamura, and Yusuke Furukawa. Expression levels of histone deacetylases determine the cell fate of hematopoietic progenitors. *Journal of Biological Chemistry*, 284(44):30673–30683, 2009.

[123] Stefan Bonn, Robert P Zinzen, Charles Girardot, E Hilary Gustafson, Alexis Perez-Gonzalez, Nicolas Delhomme, Yad Ghavi-Helm, Bartek Wilczyński, Andrew Riddell, and Eileen E M Furlong. Tissue-specific analysis of chromatin state identifies temporal signatures of enhancer activity during embryonic development. *Nature Genetics*, 44:148 EP –, 01 2012.

[124] Weiwei Yang and Patricia Ernst. Set/mll family proteins in hematopoiesis and leukemia. *International Journal of Hematology*, 105(1):7–16, Jan 2017.

[125] Jonathan M. Burg, Julie J. Gonzalez, Kenneth R. Maksimchuk, and Dewey G. McCafferty. Lysine-specific demethylase 1a (kdm1a/lsd1): Product recognition and kinetic analysis of full-length histones. *Biochemistry*, 55(11):1652–1662, 2016. PMID: 26673564.

[126] Nathaniel D Heintzman, Rhona K Stuart, Gary Hon, Yutao Fu, Christina W Ching, R David Hawkins, Leah O Barrera, Sara Van Calcar, Chunxu Qu, Keith A Ching, Wei Wang, Zhiping Weng, Roland D Green, Gregory E Crawford, and Bing Ren. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nature Genetics*, 39:311 EP –, 02 2007.

[127] Artem Barski, Suresh Cuddapah, Kairong Cui, Tae-Young Roh, Dustin E. Schones, Zhibin Wang, Gang Wei, Iouri Chepelev, and Keji Zhao. High-resolution profiling of histone methylations in the human genome. *Cell*, 129(4):823 – 837, 2007.

[128] Runsheng He and Benjamin L. Kidder. H3K4 demethylase KDM5B regulates global dynamics of transcription elongation and alternative splicing in embryonic stem cells. *Nucleic Acids Research*, 45(11):6427–6441, 04 2017.

[129] Justin S. Becker, Dario Nicetto, and Kenneth S. Zaret. H3k9me3-dependent heterochromatin: Barrier to cell fate changes. *Trends in Genetics*, 32(1):29 – 41, 2016.

[130] Panagis Filippakopoulos, Sarah Picaud, Maria Mangos, Tracy Keates, Jean-Philippe Lambert, Dalia Barsyte-Lovejoy, Ildiko Felletar, Rudolf Volkmer, Susanne Müller, Tony Pawson, Anne-Claude Gingras, Cheryl H. Arrowsmith, and Stefan Knapp. Histone recognition and large-scale structural analysis of the human bromodomain family. *Cell*, 149(1):214 – 231, 2012.

[131] Vanessa Brès, Sunnie M Yoh, and Katherine A Jones. The multi-tasking p-tefb complex. *Current Opinion in Cell Biology*, 20(3):334 – 340, 2008. Nucleus and gene expression.

[132] Christine Ender, Azra Krek, Marc R. Friedländer, Michaela Beitzinger, Lasse Weinmann, Wei Chen, Sébastien Pfeffer, Nikolaus Rajewsky, and Gunter Meister. A human snorna with microrna-like functions. *Molecular Cell*, 32(4):519 – 528, 2008.

[133] Emily F. Finnegan and Amy E. Pasquinelli. Microrna biogenesis: regulating the regulators. *Critical Reviews in Biochemistry and Molecular Biology*, 48(1):51–68, 2013.

[134] David P. Bartel. Micrornas: Target recognition and regulatory functions. *Cell*, 136(2):215 – 233, 2009.

[135] Mikiko C. Siomi, Kaoru Sato, Dubravka Pezic, and Alexei A. Aravin. Piwi-interacting small rnas: the vanguard of genome defence. *Nature Reviews Molecular Cell Biology*, 12:246 EP –, 03 2011.

[136] Alexei A. Aravin, Ravi Sachidanandam, Deborah Bourc'his, Christopher Schaefer, Dubravka Pezic, Katalin Fejes Toth, Timothy Bestor, and Gregory J. Hannon. A pirna pathway primed by individual transposons is linked to de novo dna methylation in mice. *Molecular Cell*, 31(6):785 – 799, 2008.

[137] Colin D. Malone, Julius Brennecke, Monica Dus, Alexander Stark, W. Richard McCombie, Ravi Sachidanandam, and Gregory J. Hannon. Specialized pirna pathways act in germline and somatic tissues of the drosophila ovary. *Cell*, 137(3):522 – 535, 2009.

[138] Robin Andersson, Claudia Gebhard, Irene Miguel-Escalada, Ilka Hoof, Jette Bornholdt, Mette Boyd, Yun Chen, Xiaobei Zhao, Christian Schmidl, Takahiro Suzuki, Evgenia Ntini, Erik Arner, Eivind Valen, Kang Li, Lucia Schwarzfischer, Dagmar Glatz, Johanna Raithel, Berit Lilje, Nicolas Rapin, Frederik Otzen Bagger, Mette Jørgensen, Peter Refsing Andersen, Nicolas Bertin, Owen Rackham, A. Maxwell Burroughs, J. Kenneth Baillie, Yuri Ishizu, Yuri Shimizu, Erina Furuhata, Shiori Maeda, Yutaka Negishi, Christopher J. Mungall, Terrence F. Meehan, Timo Lassmann, Masayoshi Itoh, Hideya Kawaji, Naoto Kondo, Jun Kawai, Andreas Lennartsson, Carsten O. Daub, Peter Heutink, David A. Hume, Torben Heick Jensen, Harukazu Suzuki, Yoshihide Hayashizaki, Ferenc Müller, The FANTOM Consortium, Alistair R. R. Forrest, Piero Carninci, Michael Rehli, and Albin Sandelin. An

atlas of active enhancers across human cell types and tissues. *Nature*, 507:455 EP –, 03 2014.

[139] Gioacchino Natoli and Jean-Christophe Andrau. Noncoding transcription at enhancers: General principles and functional models. *Annual Review of Genetics*, 46(1):1–19, 2012. PMID: 22905871.

[140] Minna U. Kaikkonen, Nathanael J. Spann, Sven Heinz, Casey E. Romanoski, Karmel A. Allison, Joshua D. Stender, Hyun B. Chun, David F. Tough, Rab K. Prinjha, Christopher Benner, and Christopher K. Glass. Remodeling of the enhancer landscape during macrophage activation is coupled to enhancer transcription. *Molecular Cell*, 51(3):310 – 325, 2013.

[141] Wenbo Li, Dimple Notani, Qi Ma, Bogdan Tanasa, Esperanza Nunez, Aaron Yun Chen, Daria Merkurjev, Jie Zhang, Kenneth Ohgi, Xiaoyuan Song, Soohwan Oh, Hong-Sook Kim, Christopher K. Glass, and Michael G. Rosenfeld. Functional roles of enhancer rnas for oestrogen-dependent transcriptional activation. *Nature*, 498:516 EP –, 06 2013.

[142] Michael T. Y. Lam, Han Cho, Hanna P. Lesch, David Gosselin, Sven Heinz, Yumiko Tanaka-Oishi, Christopher Benner, Minna U. Kaikkonen, Aneeza S. Kim, Mika Kosaka, Cindy Y. Lee, Andy Watt, Tamar R. Grossman, Michael G. Rosenfeld, Ronald M. Evans, and Christopher K. Glass. Rev-erbs repress macrophage gene expression by inhibiting enhancer-directed transcription. *Nature*, 498:511 EP –, 06 2013.

[143] Carlos A. Melo, Jarno Drost, Patrick J. Wijchers, Harmen van de Werken, Elzo de Wit, Joachim A.F. Oude Vrielink, Ran Elkon, Sónia A. Melo, Nicolas Léveillé, Raghu Kalluri, Wouter de Laat, and Reuven Agami. ernas are required for p53-dependent enhancer activity and gene transcription. *Molecular Cell*, 49(3):524 – 535, 2013.

[144] Yumiko Oishi, Shinichiro Hayashi, Takayuki Isagawa, Motohiko Oshima, Atsushi Iwama, Shigeki Shimba, Hitoshi Okamura, and Ichiro Manabe. Bmal1 regulates inflammatory responses in macrophages by modulating enhancer rna transcription. *Scientific Reports*, 7(1):7086, 2017.

[145] Buenrostro Jason D., Wu Beijing, Chang Howard Y., and Greenleaf William J. Atac-seq: A method for assaying chromatin accessibility genome-wide. *Current Protocols in Molecular Biology*, 109(1):21.29.1–21.29.9, 2015.

[146] M. Ryan Corces, Alexandro E. Trevino, Emily G. Hamilton, Peyton G. Greenside, Nicholas A. Sinnott-Armstrong, Sam Vesuna, Ansuman T. Satpathy, Adam J. Rubin, Kathleen S. Montine, Beijing Wu, Arwa Kathiria, Seung Woo Cho, Maxwell R. Mumbach, Ava C. Carter, Maya Kasowski, Lisa A. Orloff, Viviana I. Risca, Anshul Kundaje, Paul A. Khavari, Thomas J. Montine, William J. Greenleaf, and

Howard Y. Chang. An improved ATAC-seq protocol reduces background and enables interrogation of frozen tissues. *Nature Methods*, 2017.

[147] Parks David R., Roederer Mario, and Moore Wayne A. A new “logicle” display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. *Cytometry Part A*, 69A(6):541–551, 2006.

[148] Sven Heinz, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C. Lin, Peter Laslo, Jason X. Cheng, Cornelis Murre, Harinder Singh, and Christopher K. Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular Cell*, 38(4):576–589, 2010.

[149] M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[150] McCarthy, Davis J., Chen, Yunshun, Smyth, and Gordon K. Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic Acids Research*, 40(10):4288–4297, 2012.

[151] B. Ellis, P. Haaland, F. Hahne, N. Le Meur, N. Gopalakrishnan, J. Spidlen, and M. Jiang. *flowCore: flowCore: Basic structures for flow cytometry data*, 2017. R package version 1.44.0.

[152] R. Gentleman, F. Hahne, J. Kettman, N. Le Meur, and N. Gopalakrishnan. *flowQ: Quality control for flow cytometry*, 2017. R package version 1.38.0.

[153] J. Spidlen., N. Gopalakrishnan, F. Hahne, B. Ellis, R. Gentleman, M. Dalphin, N. Le Meur, B. Purcell, and W. Jiang. *flowUtils: Utilities for flow cytometry*, 2017. R package version 1.42.0.

[154] B. Ellis, R. Gentleman, F. Hahne, N. Le Meur, D. Sarkar, and M. Jiang. *flowViz: Visualization for flow cytometry*, 2017. R package version 1.42.0.

[155] Barret Schloerke, Jason Crowley, Di Cook, Francois Briatte, Moritz Marbach, Edwin Thoen, Amos Elberg, and Joseph Larmarange. *GGally: Extension to 'ggplot2'*, 2017. R package version 1.3.2.

[156] Mike Jiang. *ggcyto: Visualize Cytometry data with ggplot*, 2015. R package version 1.6.1.

[157] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2009.

[158] Gregory R. Warnes, Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber Andy Liaw, Thomas Lumley, Martin Maechler, Arni Magnusson, Steffen Moeller, Marc Schwartz, and Bill Venables. *gplots: Various R Programming Tools for Plotting Data*, 2016. R package version 3.0.1.

[159] Erich Neuwirth. *RColorBrewer: ColorBrewer Palettes*, 2014. R package version 1.1-2.

[160] G. Grothendieck. *sqldf: Manipulate R Data Frames Using SQL*, 2017. R package version 0.4-11.

[161] L.J.P. van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.

[162] Alboukadel Kassambara. *ggpubr: 'ggplot2' Based Publication Ready Plots*, 2018. R package version 0.1.7.

[163] Kamil Slowikowski. *ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'*, 2018. R package version 0.8.0.

[164] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature Methods*, 9, 2012.

[165] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[166] Marc R. Friedländer, Sebastian D. Mackowiak, Na Li, Wei Chen, and Nikolaus Rajewsky. mirdeep2 accurately identifies known and hundreds of novel microrna genes in seven animal clades. *Nucleic Acids Research*, 40(1):37–52, 2012.

[167] Jesse H. Krijthe. *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation*, 2015. R package version 0.13.

[168] Davis J. McCarthy, Yunshun Chen, and Gordon K. Smyth. Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic Acids Research*, 40(10):4288–4297, 2012.

[169] Aaron T. L. Lun, Yunshun Chen, and Gordon K. Smyth. *It's DE-licious: A Recipe for Differential Expression Analyses of RNA-seq Experiments Using Quasi-Likelihood Methods in edgeR*, pages 391–416. Springer New York, New York, NY, 2016.

[170] Shashank Tripathi, Marie O. Pohl, Yingyao Zhou, Ariel Rodriguez-Frandsen, Guojun Wang, David A. Stein, Hong M. Moulton, Paul DeJesus, Jianwei Che, Lubbertus C. F. Mulder, Emilio Yángüez, Dario Andenmatten, Lars Pache, Balaji Manicassamy, Randy A. Albrecht, Maria G. Gonzalez, Quy Nguyen, Abraham Brass, Stephen Elledge, Michael White, Sagi Shapira, Nir Hacohen, Alexander Karlas, Thomas F. Meyer, Michael Shales, Andre Gatorano, Jeffrey R. Johnson, Gwen Jang, Tasha Johnson, Erik Verschueren, Doug Sanders, Nevan Krogan, Megan Shaw, Renate König, Silke Stertz, Adolfo García-Sastre, and Sumit K. Chanda. Meta- and orthogonal integration of influenza &#x201c;omics&#x201d; data defines a role for ubr4 in virus budding. *Cell Host & Microbe*, 18(6):723–735, 2018/07/22 2015.

[171] Ana Lukic, Pia Larssen, Alexander Fauland, Bengt Samuelsson, Craig E. Wheelock, Susanne Gabrielsson, and Olof Radmark. Gm-csf– and m-csf–primed macrophages present similar resolving but distinct inflammatory lipid mediator signatures. *The FASEB Journal*, 31(10):4370–4381, 2017. PMID: 28637652.

[172] Marietta Margit Budai, József Tőzsér, and Szilvia Benkő. Different dynamics of nlrp3 inflammasome-mediated il-1 production in gm-csf– and m-csf–differentiated human macrophages. *Journal of Leukocyte Biology*, 101(6):1335–1347, 2017.

[173] Jennifer C Miller, Brian D Brown, Tal Shay, Emmanuel L Gautier, Vladimir Jojic, Ariella Cohain, Gaurav Pandey, Marylene Leboeuf, Kutlu G Elpek, Julie Helft, Daigo Hashimoto, Andrew Chow, Jeremy Price, Melanie Greter, Milena Bogunovic, Angelique Bellemare-Pelletier, Paul S Frenette, Gwendalyn J Randolph, Shannon J Turley, Miriam Merad, the Immunological Genome Consortium, Claudia Jakubzick, Adam J Best, Jamie Knell, Ananda Goldrath, Jennifer Miller, Brian Brown, Daphne Koller, Nadia Cohen, Patrick Brennan, Michael Brenner, Aviv Regev, Anne Fletcher, Kutlu Elpek, Deepali Malhotra, Shannon Turley, Radu Jianu, David Laidlaw, Jim Collins, Kavitha Narayan, Katelyn Sylvia, Joonsoo Kang, Roi Gazit, Derrick J Rossi, Francis Kim, Tata Nageswara Rao, Amy Wagers, Susan A Shinton, Richard R Hardy, Paul Monach, Natalie A Bezman, Joseph C Sun, Charlie C Kim, Lewis L Lanier, Tracy Heng, Taras Kreslavsky, Michio Painter, Jeffrey Ericson, Scott Davis, Diane Mathis, and Christophe Benoist. Deciphering the transcriptional network of the dendritic cell lineage. *Nature Immunology*, 13:888 EP –, 07 2012.

[174] Bindu Varghese, Erina Vlashi, Wei Xia, Wilfredo Ayala Lopez, Chrystal M. Paulos, Joseph Reddy, Le-Cun Xu, and Philip S. Low. Folate receptor-β in activated macrophages: Ligand binding and receptor recycling kinetics. *Molecular Pharmaceutics*, 11(10):3609–3616, 2014. PMID: 25166491.

[175] Geanncarlo Lugo-Villarino, Anthony Troegeler, Luciana Balboa, Claire Lastrucci, Carine Duval, Ingrid Mercier, Alan Bénard, Florence Capilla, Talal Al Saati, Renaud Poincloux, Ivanela Kondova, Frank A. W. Verreck, Céline Cougoule, Isabelle Maridonneau-Parini, Maria del Carmen Sasiain, and Olivier Neyrolles. The c-type lectin receptor dc-sign has an anti-inflammatory role in human m(il-4) macrophages in response to mycobacterium tuberculosis. *Frontiers in Immunology*, 9:1123, 2018.

[176] Joanne Tonkin, Lieve Temmerman, Robert D Sampson, Enrique Gallego-Colon, Laura Barberi, Daniel Bilbao, Michael D Schneider, Antonio Musarò, and Nadia Rosenthal. Monocyte/macrophage-derived igf-1 orchestrates murine skeletal muscle regeneration and modulates autocrine polarization. *Molecular Therapy*, 23(7):1189 – 1200, 2015.

[177] Olga Spadaro, Christina D. Camell, Lidia Bosurgi, Kim Y. Nguyen, Yun-Hee Youm, Carla V. Rothlin, and Vishwa Deep Dixit. Igf1 shapes macrophage activation in response to immunometabolic challenge. *Cell Reports*, 19(2):225–234, 2019/02/13 2017.

[178] Michelino Di Rosa, Giulia Malaguarnera, Corinne De Gregorio, Filippo Drago, and Lucia Malaguarnera. Evaluation of chi3l-1 and chit-1 expression in differentiated and polarized macrophages. *Inflammation*, 36(2):482–492, Apr 2013.

[179] Krause Stefan W., Rehli Michael, Kreutz Marina, Schwarzfischer Lucia, Paulauskis Joseph D., and Andreesen Reinhard. Differential screening identifies genetic markers of monocyte to macrophage maturation. *Journal of Leukocyte Biology*, 60(4):540–545, 1996.

[180] Michael Rehli, Hans-Helmut Niller, Christoph Ammon, Sabine Langmann, Lucia Schwarzfischer, Reinhard Andreesen, and Stefan W. Krause. Transcriptional regulation of chi3l1, a marker gene for late stages of macrophage differentiation. *Journal of Biological Chemistry*, 278(45):44058–44067, 2003.

[181] Thu-Hang Pham, Christopher Benner, Monika Lichtinger, Lucia Schwarzfischer, Yuhui Hu, Reinhard Andreesen, Wei Chen, and Michael Rehli. Dynamic epigenetic enhancer signatures reveal key transcription factors associated with monocytic differentiation states. *Blood*, 119(24):e161–e171, 2012.

[182] Ioannis Vlachos, Konstantinos Zagganas, Maria D. Paraskevopoulou, Gerogios Gerogaskilas, Diitra Karagkouni, Thanasis Vergoulis, Theodore Dalamagas, and Artemis G. Hatzigeorgiou. Diana-mirpath v3. 0: deciphering microrna function with experimental support. *Nucleic acids research*, 2015.

[183] Emilie Bourdonnay, Claudie Morzadec, Olivier Fardel, and Laurent Vernhet. Redox-sensitive regulation of gene expression in human primary macrophages exposed to inorganic arsenic. *Journal of Cellular Biochemistry*, 107(3):537–547, 2009.

[184] Attila Szanto, Balint L. Balint, Zsuzsanna S. Nagy, Endre Barta, Balazs Dezso, Attila Pap, Lajos Szeles, Szilard Poliska, Melinda Oros, Ronald M. Evans, Yaacov Barak, John Schwabe, and Laszlo Nagy. Stat6 transcription factor is a facilitator of the nuclear receptor pparγ-regulated gene expression in macrophages and dendritic cells. *Immunity*, 33(5):699–712, 2018/07/25 2010.

[185] Xiaoling Li. Sirt1 and energy metabolism. *Acta Biochimica et Biophysica Sinica*, 45(1):51–60, 2013.

[186] Malay Haldar, Masako Kohyama, Alex Yick-Lun So, Wumesh KC, Xiaodi Wu, Carlos G. Briseño, Ansuman T. Satpathy, Nicole M. Kretzer, Hisashi Arase, Namakkal S. Rajasekaran, Li Wang, Takeshi Egawa, Kazuhiko Igarashi, David Baltimore, Theresa L. Murphy, and Kenneth M. Murphy. Heme-mediated spi-c induction promotes monocyte differentiation into iron-recycling macrophages. *Cell*, 156(6):1223 – 1234, 2014.

[187] Chen Hualin, Xu Wenli, Liu Dapeng, Li Xijing, Pan Xiuhua, and Pang Qingfeng. The anti-inflammatory mechanism of heme oxygenase-1 induced by hemin in primary rat alveolar macrophages. *Inflammation*, 35(3):1087–1093, Jun 2012.

[188] Thai Hien Tu, Yeonsoo Joe, Hye-Seon Choi, Hun Taeg Chung, and Rina Yu. Induction of heme oxygenase-1 with hemin reduces obesity-induced adipose tissue inflammation via adipose macrophage phenotype switching. *Mediators of Inflammation*, 2014:290708, 2014.

[189] K Ishikawa, M Navab, N Leitinger, A M Fogelman, and A J Lusis. Induction of heme oxygenase-1 inhibits the monocyte transmigration induced by mildly oxidized ldl. *The Journal of Clinical Investigation*, 100(5):1209–1216, 9 1997.

[190] Detlef Lang, Stefan Reuter, Tania Buzescu, Christian August, and Stefan Heidenreich. Heme-induced heme oxygenase-1 (ho-1) in human monocytes inhibits apoptosis despite caspase-3 up-regulation. *International Immunology*, 17(2):155–165, 2005.

[191] Francesca Vinchi, Milene Costa da Silva, Giada Ingoglia, Sara Petrillo, Nathan Brinkman, Adrian Zuercher, Adelheid Cerwenka, Emanuela Tolosano, and Martina U. Muckenthaler. Hemopexin therapy reverts heme-induced proinflammatory phenotypic switching of macrophages in a mouse model of sickle cell disease. *Blood*, 127(4):473–486, 2016.

[192] Mateo de las Casas-Engel, Angeles Domínguez-Soto, Elena Sierra-Filardi, Rafael Bragado, Concha Nieto, Amaya Puig-Kroger, Rafael Samaniego, Mabel Loza, María Teresa Corcuera, Fernando Gómez-Aguado, Matilde Bustos, Paloma Sánchez-Mateos, and Angel L. Corbí. Serotonin skews human macrophage polarization through htr2b and htr7. *The Journal of Immunology*, 190(5):2301–2310, 2013.

[193] Silke Müller, Thomas Quast, Agnes Schröder, Stephanie Hucke, Luisa Klotz, Jonathan Jantsch, Rupert Gerzer, Ruth Hemmersbach, and Waldemar Kolanus. Salt-dependent chemotaxis of macrophages. *PLOS ONE*, 8(9):1–9, 09 2013.

[194] Nurgazieva Dinara, Mickley Amanda, Moganti Kondaiah, Ming Wen, Ovsyi Illya, Popova Anna, Sachindra, Awad Kareem, Wang Nan, Bieback Karen, Goerdt Sergij, Kzhyshkowska Julia, and Gratchev Alexei. Tgf-β1, but not bone morphogenetic proteins, activates smad1/5 pathway in primary human macrophages and induces expression of proatherogenic genes. *The Journal of Immunology*, 194(2):709–718, 2015.

[195] Elena Sierra-Filardi, Concha Nieto, Ángeles Domínguez-Soto, Rubén Barroso, Paloma Sánchez-Mateos, Amaya Puig-Kroger, María López-Bravo, Jorge Joven, Carlos Ardavín, José L. Rodríguez-Fernández, Carmen Sánchez-Torres, Mario Mellado, and Ángel L. Corbí. Ccl2 shapes macrophage polarization by gm-csf and m-csf: Identification of ccl2/ccr2-dependent gene expression profile. *The Journal of Immunology*, 192(8):3858–3867, 2014.

[196] J F Bohnsack, H K Kleinman, T Takahashi, J J O'Shea, and E J Brown. Connective tissue proteins and phagocytic cell function. laminin enhances complement and fc-mediated phagocytosis by cultured human macrophages. *Journal of Experimental Medicine*, 161(5):912–923, 1985.

[197] Pengfei Lu, Ken Takai, Valerie M. Weaver, and Zena Werb. Extracellular matrix degradation and remodeling in development and disease. *Cold Spring Harbor Perspectives in Biology*, 3(12), 2011.

[198] Reel Buket, Sala-Newby Graciela B, Huang Wei-Chun, and Newby Andrew C. Diverse patterns of cyclooxygenase-independent metalloproteinase gene regulation in human monocytes. *British Journal of Pharmacology*, 163(8):1679–1690, 2011.

[199] S Haskill, C H. Johnson, D Eierman, S Becker, and K Warren. Adherence induces selective mrna expression of monocyte mediators and proto-oncogenes. *Journal of immunology*, 140:1690–4, 04 1988.

[200] Simon L. Newman, Bindu Bhugra, Angela Holly, and Randal E. Morris. Enhanced killing of candida albicans by human macrophages adherent to type 1 collagen matrices via induction of phagolysosomal fusion. *Infection and Immunity*, 73(2):770–777, 2005.

[201] Georg Kraal, Luc J. W. van der Laan, Outi Elomaa, and Karl Tryggvason. The macrophage receptor marco. *Microbes and Infection*, 2(3):313 – 316, 2000.

[202] Mario Henrique M. Barros, Franziska Hauck, Johannes H. Dreyer, Bettina Kempkes, and Gerald Niedobitek. Macrophage polarisation: an immunohistochemical approach for identifying m1 and m2 macrophages. *PLOS ONE*, 8(11), 11 2013.

[203] A Schridde, C C Bain, J U Mayer, J Montgomery, E Pollet, B Denecke, S W F Milling, S J Jenkins, M Dalod, S Henri, B Malissen, O Pabst, and A Mcl Mowat. Tissue-specific differentiation of colonic macrophages requires tgfreceptor-mediated signaling. *Mucosal Immunology*, 10:1387 EP –, 02 2017.

[204] John F. Kuemmerle. Insulin-like growth factors in the gastrointestinal tract and liver. *Endocrinology and Metabolism Clinics of North America*, 41(2):409 – 423, 2012. Insulin-Like Growth Factors in Health and Disease.

[205] David R. Clemmons and John I. Jones. Insulin-Like Growth Factors and Their Binding Proteins: Biological Actions. *Endocrine Reviews*, 16(1):3–34, 02 1995.

[206] Joëlle Dupont and Martin Holzenberger. Biology of insulin-like growth factors in development. *Birth Defects Research Part C: Embryo Today: Reviews*, 69(4):257–271, 2003.

[207] Yu Zheng, Yongli Song, Qi Han, Wenjie Liu, Jiuzhi Xu, Zhengquan Yu, Ran Zhang, and Ning Li. Intestinal epithelial cell-specific igf1 promotes the expansion of intestinal stem cells during epithelial regeneration and functions on the intestinal immune homeostasis. *American Journal of Physiology-Endocrinology and Metabolism*, 315(4):E638–E649, 2018. PMID: 29783855.

[208] Hye Rim Chang, Hae Jin Kim, Xiaoyuan Xu, and Anthony W. Ferrante Jr. Macrophage and adipocyte igf1 maintain adipose tissue homeostasis during metabolic stresses. *Obesity*, 24(1):172–183, 2016.

[209] Hyeon-Sook Suh, Meng-Liang Zhao, Leandra Derico, Namjong Choi, and Sunhee C. Lee. Insulin-like growth factor 1 and 2 (igf1, igf2) expression in human microglia: differential regulation by inflammatory mediators. *Journal of Neuroinflammation*, 10(1):805, Mar 2013.

[210] Guobao Chen, Carol Shurong Tan, Boon King Teh, and Jinhua Lu. Molecular mechanisms for synchronized transcription of three complement c1q subunit genes in dendritic cells and macrophages. *Journal of Biological Chemistry*, 286(40):34941–34950, 2011.

[211] Ying He, Christopher R. Jones, Nobuhiro Fujiki, Ying Xu, Bin Guo, Jimmy L. Holder, Moritz J. Rossner, Seiji Nishino, and Ying-Hui Fu. The transcriptional repressor dec2 regulates sleep length in mammals. *Science*, 325(5942):866–870, 2009.

[212] Anthony H. Tsang, Carmen Sánchez-Moreno, Brid Bode, Moritz J. Rossner, Marta Garaulet, and Henrik Oster. Tissue-specific interaction of per1/2 and dec2 in the regulation of fibroblast circadian rhythms. *Journal of Biological Rhythms*, 27(6):478–489, 2012. PMID: 23223373.

[213] Naoya Matsunaga, Miki Inoue, Naoki Kusunose, Keisuke Kakimoto, Kengo Hamamura, Yuichi Hanada, Ayumi Toi, Yuji Yoshiyama, Fuyuki Sato, Katsumi Fujimoto, Satoru Koyanagi, and Shigehiro Ohdo. Time-dependent interaction between differentiated embryo chondrocyte-2 and ccaat/enhancer-binding protein α underlies the circadian expression of cyp2d6 in serum-shocked hepg2 cells. *Molecular Pharmacology*, 81(5):739–747, 2012.

[214] Sameena Azmi, Hong Sun, Anne Ozog, and Reshma Taneja. msharp-1/dec2, a basic helix-loop-helix protein functions as a transcriptional repressor of e box activity and stra13 expression. *Journal of Biological Chemistry*, 278(22):20098–20109, 2003.

[215] John H. Carter and Warren G. Tourtellotte. Early growth response transcriptional regulators are dispensable for macrophage differentiation. *The Journal of Immunology*, 178(5):3038–3047, 2007.

[216] Sina Taefehshokr, Yashar Azari Key, Mansour Khakpour, Pourya Dadebighlu, and Amin Oveisi. Early growth response 2 and egr3 are unique regulators in immune system. *Central European Journal of&nbsp;Immunology*, 42(2):205–209, 2017.

[217] Meredith Safford, Samuel Collins, Michael A Lutz, Amy Allen, Ching-Tai Huang, Jeanne Kowalski, Amanda Blackford, Maureen R Horton, Charles Drake, Ronald H Schwartz, and Jonathan D Powell. Egr-2 and egr-3 are negative regulators of t cell activation. *Nature Immunology*, 6:472 EP –, 04 2005.

[218] Hung Q. Nguyen, Barbara Hoffman-Liebermann, and Dan A. Liebermann. The zinc finger transcription factor egr-1 is essential for and restricts differentiation along the macrophage lineage. *Cell*, 72(2):197 – 209, 1993.

[219] Erin A. Jones, Sung-Wook Jang, Gennifer M. Mager, Li-Wei Chang, Rajini Srinivasan, Nolan G. Gokey, Rebecca M. Ward, Rakesh Nagarajan, and John Svaren. Interactions of sox10 and egr2 in myelin gene regulation. *Neuron Glia Biology*, 3(4):377–387, 2007.

[220] Courtney Krueger, Erin A. Jones, Guannan Sun, John J. Moran, Rajini Srinivasan, Sunduz Keles, Sung-Wook Jang, and John Svaren. Genome-wide analysis of EGR2/SOX10 binding in myelinating peripheral nerve. *Nucleic Acids Research*, 40(14):6449–6460, 04 2012.

[221] T. Sevilla, R. Sivera, D. Martínez-Rubio, V. Lupo, M. J. Chumillas, E. Calpena, J. Dopazo, J. J. Vílchez, F. Palau, and C. Espinós. The egr2 gene is involved in axonal charcotmarietooth disease. *European Journal of Neurology*, 22(12):1548–1555, 2015.

[222] E Young, D Noerenberg, L Mansouri, V Ljungström, M Frick, L-A Sutton, S J Blakemore, J Galan-Sousa, K Plevova, P Baliakas, D Rossi, R Clifford, D Roos-Weil, V Navrkalova, B Dörken, C A Schmitt, K E Smedby, G Juliusson, B Giacopelli, J S Blachly, C Belessi, P Panagiotidis, N Chiorazzi, F Davi, A W Langerak, D Oscier, A Schuh, G Gaidano, P Ghia, W Xu, L Fan, O A Bernard, F Nguyen-Khac, L Rassenti, J Li, T J Kipps, K Stamatopoulos, S Pospisilova, T Zenz, C C Oakes, J C Strefford, R Rosenquist, and F Damm. Egr2 mutations define a new clinically aggressive subgroup of chronic lymphocytic leukemia. *Leukemia*, 31:1547 EP –, 11 2016.

[223] Christopher C Oakes, Marc Seifert, Yassen Assenov, Lei Gu, Martina Przekopowitz, Amy S Ruppert, Qi Wang, Charles D Imbusch, Andrius Serva, Sandra D Koser, David Brocks, Daniel B Lipka, Olga Bogatyrova, Dieter Weichenhan, Benedikt Brors, Laura Rassenti, Thomas J Kipps, Daniel Mertens, Marc Zapatka, Peter Lichter, Hartmut Döhner, Ralf Küppers, Thorsten Zenz, Stephan Stilgenbauer, John C Byrd, and Christoph Plass. Dna methylation dynamics during b cell maturation underlie a continuum of disease phenotypes in chronic lymphocytic leukemia. *Nature Genetics*, 48:253 EP –, 01 2016.

[224] Holly Hung, Rebecca Kohnken, and John Svaren. The nucleosome remodeling and deacetylase chromatin remodeling (nurd) complex is required for peripheral nerve myelination. *Journal of Neuroscience*, 32(5):1517–1527, 2012.

[225] Peter Laslo, Chauncey J. Spooner, Aryeh Warmflash, David W. Lancki, Hyun-Jun Lee, Roger Sciammas, Benjamin N. Gantner, Aaron R. Dinner, and Harinder Singh. Multilineage transcriptional priming and determination of alternate hematopoietic cell fates. *Cell*, 126(4):755–766, 2019/02/24 2006.

[226] Seishiro Hirano, C. D. Anuradha, and Sanae Kanno. Transcription of krox-20/egr-2 is upregulated after exposure to fibrous particles and adhesion in rat alveolar macrophages. *American Journal of Respiratory Cell and Molecular Biology*, 23(3):313–319, 2000. PMID: 10970821.

[227] Kyle A. Jablonski, Stephanie A. Amici, Lindsay M. Webb, Juan de Dios Ruiz-Rosado, Phillip G. Popovich, Santiago Partida-Sanchez, and Mireia Guerau-de Arellano. Novel markers to delineate murine m1 and m2 macrophages. *PLOS ONE*, 10(12):1–25, 12 2015.

[228] Tatyana Veremeyko, Amanda W. Y. Yung, Daniel C. Anthony, Tatyana Strekalova, and Eugene D. Ponomarev. Early growth response gene-2 is essential for m1 and m2 macrophage activation and plasticity by modulation of the transcription factor cebp. *Frontiers in Immunology*, 9:2515, 2018.

[229] Yankel Gabet, Sanjeev K. Baniwal, Nathalie Leclerc, Yunfan Shi, Alice E. Kohn-Gabet, Jon Cogan, Alexis Dixon, Marilyn Bachar, Lixin Guo, Jack E. Turman, and Baruch Frenkel. Krox20/egr2 deficiency accelerates cell growth and differentiation in the monocytic lineage and decreases bone mass. *Blood*, 116(19):3964–3971, 2010.

[230] Himangi G. Marathe, Gaurav Mehta, Xiaolu Zhang, Ila Datar, Aanchal Mehrotra, Kam C. Yeung, and Ivana L. de la Serna. Swi/snf enzymes promote sox10- mediated activation of myelin gene expression. *PLOS ONE*, 8(7), 07 2013.

[231] Eyal Y. Anteby, Shira Natanson-Yaron, Yaron Hamani, Yael Sciaki, Debra Goldman-Wohl, Caryn Greenfield, Ilana Ariel, and Simcha Yagel. Fibroblast growth factor-10 and fibroblast growth factor receptors 1–4: expression and peptide localization in human decidua and placenta. *European Journal of Obstetrics & Gynecology and Reproductive Biology*, 119(1):27 – 35, 2005.

[232] Salma Khan, Hidetaka Katabuchi, Masako Araki, Ryuichiro Nishimura, and Hitoshi Okamura. Human villous macrophage-conditioned media enhance human trophoblast growth and differentiation in vitro. *Biology of Reproduction*, 62(4):1075–1083, 2000.

[233] Ursula Hiden, Elisabeth Glitzner, Michaele Hartmann, and Gernot Desoye. Insulin and the igf system in the human placenta of normal and diabetic pregnancies. *Journal of Anatomy*, 215(1):60–68, 2009.

[234] S. Sonderegger, J. Pollheimer, and M. Knöfler. Wnt signalling in implantation, decidualisation and placental differentiation – review. *Placenta*, 31(10):839 – 847, 2010.

[235] Stephanie A. Ragland and Alison K. Criss. From bacterial killing to immune modulation: Recent insights into the functions of lysozyme. *PLOS Pathogens*, 13(9):1–22, 09 2017.

[236] Harmit S. Ranhotra. The nr4a orphan nuclear receptors: mediators in metabolism and diseases. *Journal of Receptors and Signal Transduction*, 35(2):184–188, 2015.

[237] Graham D Thomas, Richard N Hanna, Neelakatan T Vasudevan, Anouk A Hamers, Casey E Romanoski, Sara McArdle, Kevin D Ross, Amy Blatchley, Deborah

Yoakum, Bruce A Hamilton, Zbigniew Mikulski, Mukesh K Jain, Christopher K Glass, and Catherine C Hedrick. Deleting an nr4a1 super-enhancer subdomain ablates ly6c(low) monocytes while preserving macrophage gene function. *Immunity*, 45(5):975–987, 11 2016.

[238] Natacha Ipseiz, Stefan Uderhardt, Carina Scholtysek, Martin Steffen, Gernot Schabbauer, Aline Bozec, Georg Schett, and Gerhard Krönke. The nuclear receptor nr4a1 mediates anti-inflammatory effects of apoptotic cells. *The Journal of Immunology*, 192(10):4852–4858, 2014.

[239] Zhonghua Tang, Tracy Niven-Fairchild, Serkalem Tadesse, Errol R Norwitz, Catalin S Buhimschi, Irina A Buhimschi, and Seth Guller. Glucocorticoids enhance cd163 expression in placental hofbauer cells. *Endocrinology*, 154(1):471–482, 01 2013.

[240] Chi-Shiun Chiang, Fang-Hsin Chen, Ji-Hong Hong, Pei-Shin Jiang, Hsiang-Ling Huang, Chun-Chieh Wang, and William H. McBride. Functional phenotype of macrophages depends on assay procedures. *International Immunology*, 20(2):215–222, 12 2007.

Chapter 8

Appendix

List of Abbreviations

Abbreviation	Full Name
7-AAD	7-Aminoactinomycin
AA	Acrylamid
APS	Ammonium peroxodisulfate
ATAC	Assay for Transposase Accessible Chromatin
bp	Base pair
BSA	Bovine Serum albumin
BMM	Bone marrow macrophages
cDNA	complementary DNA
CD	Cluster of Differentiation
ChIP	Chromatin Immunoprecipitation
CML	chronic myeloid leukemia
CNS	central nervous system
COPD	chronic obstructive pulmonary disease
cpm	counts per million
DNA	Deoxynucleic acid
DMSO	Dimethylsulfoxid
ECM	Extracellular Matrix
EGR	Early response gene
FCS	Fetal calf serum
FDR	False Discovery Rate
GM-CSF	Granulocyte-Macrophage colony stimulating factor
GO	Gene Ontology
H3K27ac	Histone 3 Lysine 27 acetylation
H3K4me1	Histone 3 Lysine 3 monomethylation
H3K4me3	Histone 3 Lysine 3 trimethylation
H3K9me3	Histone 3 Lysine 9 trimethylation
H3K27me3	Histone 3 Lysine 27 trimethylation
HC	Hofbauer Cells
HCl	Hydrogen chloride
iDC	immature Dendritic cell
Ig	Immunoglobulin
IL	Interleukin
KCl	potassium chloride
LDTF	Lineage determining transcription factor
logFC	log2 Fold Change
LYZ	Lysozyme
MARCO	macrophage receptor with collagenous structure
M-CSF	Macrophage colony stimulating factor
MFI	Median Fluorescence Intensity

$MgCl_2$	Magnesium chloride
MMP	Metalloproteinase
NaBut	Sodium butyrate
NP-40	Non-Indent P40 substitute
NaCl	sodium chloride
PAA	Polyacrylamid
PBS	Buffered phosphate solution
PBMC	Peripheral Blood Mononuclear Cells
PCR	Polymerase chain reaction
PMSF	phenylmethane sulfonyl fluoride
PRC2	Polycomb repressive complex 2
RNA	Ribonucleic acid
rpm	rotation per minute
RPKM	Reads Per Kilobase Million
RT	Room temperature
rxn	Reaction
SDS	Sodium dodecylsulfate
seq	sequencing
SNP	Single nucleotide polymorphism
TAM	Tumour associated macrophage
TEMED	Tetramethylethylenediamine

Bash and Perl Scripts

These scripts were used in the analysis of several datasets and were written by Prof. Dr. Michael Rehli.
bcl2fastq-wrap.sh:

```
#!/bin/bash
# (C) 2011 by Randy Rueckner under FreeTheWorldLicense V1.0
# edited by Julia Engelmann, Aug 3, 2012,  Dec 5, 2012, Dec 7, 2015, May 5, 2017; May 8, 2017.
#
#
# adopted to NextSeq550 (Michael Rehli) by Christian Kohler 12/17 and 01/18
#
# major changes to script:
# * skip forced parameters "--use-bases-mask" "--tiles".
# => only the run number is required
# * add "-c" parameter (c=core) to allow passing number of cores to be used
# * introduced £{_ARGSOTHER} variable that collects passed parameters
##################
# old documentation from Randy/Julia
#####################
#  bcl2fastq v2_18 requires a three field folder name- deviations may cause software to stop.
# YYMMDD_machinename_NNNN
# YYMMDD: date, NNNN: 4 digit run number
# -> move folder '150701_SNL143_0116_AC6CVDACXX_131GB' to '150701_SNL143_0116'
# Sample IDs (SampleId) may only occur once within a project, else error
# v2_18 needs boost version 1.54 or higher. It throws an error even w/out a sampleSheet
# 'bad lexical cast: source type value could not be interpreted as target'
########################
#
# test call: bash bcl2fastq_wrap.sh 2 "--barcode-mismatches 0"
#usage(){
```

```
#                 printf "usage:\t£(basename "£{0}") <NextSeqRunNum> <additional bcl2fastq parameter(s)>\n\tNextSeqRunNum ::=
↪  [0-9]{1,4}\n\tadditional bcl2fastq parameters example:  \"--barcode-mismatches 0\"  \"--processing-threads 12\", don't forget to
↪  quote\n\n";
#                 exit 1 ;
#}
DOC_BCL2FASTQ="/misc/software/ngs/bcl2fastq/bin/bcl2fastq_showParams.sh"
usage(){
                printf "usage:\t$(basename "£{0}") <NextSeqRunNum> <additional bcl2fastq parameter(s)>\n\tNextSeqRunNum ::=
                ↪  [0-9]{1,4}\n\tlist <additional bcl2fastq parameters> by running:\n\t"£{DOC_BCL2FASTQ}"\n\n\tDon't forget to quote
                ↪  each parameter separately (eg: \"--processing-threads 1\" \"--barcode-mismatches 1\"\n\n";
                exit 1 ;
}
#echo £{1} # HiScanRunNum
#echo £{2} # --use-bases-mask
#echo £{3} # --mismatches
#echo £{4} # --tiles
# uncommented [2018-01-11]
##########################
#baseMask=£2
#baseMask=£{baseMask/--use-bases-mask /}
#
#
#mismatch=£3
##****# mismatch=£{mismatch/--mismatches /} # parameter is does not exist in v2-18
#mismatch=£{mismatch/--barcode-mismatches /} # use this parameter instead
#
#
#lanetile=£4
#lanetile=£{lanetile/--tiles /}
##########################
##############################
# [2018-01-21]
# still needs testing ...
#while getopts "d:" opt
#do
#                 case £opt in
#                                 d)
#                                                 echo " bla bla "
#                                                 ;;
#                 esac
#done
#
#printf "cores: £{_CORES}\n"
#exit;
############################
#############################
# [2018-01-21]
# also set to default value of 1
#
#mismatch=£2
#mismatch=£{mismatch/--barcode-mismatches /} # only required parameter
#
#if [ -z £{mismatch} ]; then
#                 printf "\nPlease provide <--barcode-mismatches arg> parameter...\n\n"
#                 usage ;
#                 exit 2 ;
#fi
############################
#****# get remaining command-line parameters that do not necessarily require an argument
#****# (eg. "--ignore-missing-bcl" or "--create-fastq-for-index-reads")
#_ARGSOTHER="£{@:5}"
_ARGSOTHER="£{@:2}" #additional parameters start with position 2
#printf "ARGSOTHER: £{_ARGSOTHER}\n"
# uncomment for debugging
#########################
# printf "baseMask: £baseMask \n"
# printf "mismatch: £mismatch \n"
# printf "lanetile: £lanetile \n"
# printf "other args: £{_ARGSOTHER} \n"
# check the parameter(s)
# to select all tiles on all lanes (no. indicates lane): --tiles s_[1{8]
# --use-bases-mask Y*,I*    # for selecting all
#****# Note: "--singleFastQ" does not exist anymore in v2-18
#[[ "£{#}" -gt "0" && "£{#}" -lt "7" ]] || {   # more than 0 and less than 7 params
#        printf "usage:\t£(basename "£{0}") <NextSeqRunNum> <additional bcl2fastq parameter(s)>\n\tNextSeqRunNum ::=
↪  [0-9]{1,4}\n\taddParam example: \"--use-bases-mask y50n,I6n,Y50n\"  \"--barcode-mismatches 0\"  \"--tiles
↪  s_[2468]_[0-9][0-9][0-9][0-9],s_1_0001\" \" --additional parameters\", don't forget to quote\n\n"
#        exit 2
#}
#****# Note: "--singleFastQ" does not exist anymore in v2-18
# skip condition less than 7 params
#[[ "£{#}" -gt "0" && "£{#}" -lt "7" ]] || {   # more than 0 and less than 7 params
[[ "£{#}" -gt "0" ]] || {
#printf "usage:\t£(basename "£{0}") <NextSeqRunNum> <additional bcl2fastq parameter(s)>\n\tNextSeqRunNum ::=  [0-9]{1,4}\n\taddParam
↪  example:  \"--barcode-mismatches 0\"  \" --processing-threads 12\", don't forget to quote\n\n"
usage ;
exit 3 ;
}
case "£{#1}" in
            4)     _RUNNUM="£{1}"
                         ;;
            3)     _RUNNUM="0£{1}"
                         ;;
            2)     _RUNNUM="00£{1}"
                         ;;
            1)     _RUNNUM="000£{1}"
                         ;;
esac
## find the path to the "NextSeqRunNum"-th dataset
_WORKPATH=$(find /misc/rehli -maxdepth 1 -type d  -name "*_NB551283_£{_RUNNUM}*" |head -1)
[ -z "£{_WORKPATH}" ] && {
printf "sorry, no data found for NextSeqRun number £{1}\n\n"
exit 4 ;
}
###############
# [2018-01-21]
# check if additional parameters are present
if [ ! -z "£{_ARGSOTHER}" ]; then
            printf "\"should $(basename "£{_WORKPATH}") be processed with \"£{1}\" and \"£{_ARGSOTHER}\" [y/N]? "
else
            printf "\"should $(basename "£{_WORKPATH}") be processed with \"£{1}\" as only parameter? [y/N]? "
fi
#printf "\nshould £(basename "£{_WORKPATH}") be processed with \"£{2}\", \"£{3}\" , \"£{4}\" and \"£{_ARGSOTHER}\" [y/N]?  "
#printf "\nshould £(basename "£{_WORKPATH}") be processed with \"£{2}\", and \"£{_ARGSOTHER}\" [y/N]?  "
read -s -n1 _go
if [ "£{_go}" == "y" -o "£{_go}" == "Y" ]; then
            printf "\nyes, let's go...\n\n"
else
            printf "\nno, better not...\n\n"
            exit 5 ;
fi
_DATE=$(date +%s)
####
# generate the job script
#_WORKPATH="."   # for debugging
#echo £{_WORKPATH}
cat >"£{_WORKPATH}/bcl2fastq.£{1}.£{_DATE}.sh" <<EOF
#!/bin/bash

cd "£{_WORKPATH}"

###########################################
# [2018-01-21]
# still needs testing ...
##***# get nr of useable cores
_NUM_JOBS=£((£(grep -c "^processor" "/proc/cpuinfo") - £(printf "%.0f" £(awk '{print £1}' <"/proc/loadavg"))))

#printf "\£{_NUM_JOBS} CPUs available ...\n"

#***# limit to 16 cores when running on computing server
_MAX_JOBS=16
#
if [ "\£{_NUM_JOBS}" -gt "\£{_MAX_JOBS}" ]; then
            _NUM_JOBS="\£{_MAX_JOBS}"
fi
#
#printf "\£{_NUM_JOBS} CPUs actually used ...\n"
############################################

printf "starting at \£(date)\n\n"

# save output at different location (2018-01-10)
_OUTPATH="/misc/data/NextSeq/£(/usr/bin/basename £{_WORKPATH})"
#printf "\£{_OUTPATH}"

mkdir "\£{_OUTPATH}"


# convert bcl to fastq- you might have to adapt the path to bcl2fastq
#***# set absolute path to bcl2fastq
#***# parameter <demultiplexing-threads> not available in version v2.20.0.422 !
#***' => use parameter <writing-threads> instead ??
```

```
#/misc/software/ngs/bcl2fastq/src/bcl2fastq_v2.20.0.422/usr/local/bin/bcl2fastq --output-dir "\£{_OUTPATH}/FastQ.£{_DATE}"
↪  --barcode-mismatches "£mismatch" --sample-sheet "SampleSheet.csv" --fastq-compression-level 9 --processing-threads "\£{_NUM_JOBS}"
↪  --writing-threads "\£{_NUM_JOBS}" --no-lane-splitting £{_ARGSOTHER}

#***' => use parameter <writing-threads> instead ??
/misc/software/ngs/bcl2fastq/src/bcl2fastq_v2.20.0.422/usr/local/bin/bcl2fastq --output-dir "\£{_OUTPATH}/FastQ.£{_DATE}"
↪  --barcode-mismatches 1 --sample-sheet "SampleSheet.csv" --fastq-compression-level 9  --no-lane-splitting £{_ARGSOTHER}

#cd "Data/Intensities/FastQ.£{_DATE}"
cd "\£{_OUTPATH}/FastQ.£{_DATE}"

make -j\£{_NUM_JOBS}


#cd "\£{_WORKPATH}"
cd "£{_WORKPATH}"
printf "\nfinishing at \£(date)\n"

cp -r "£{_WORKPATH}/InterOp" "\£{_OUTPATH}/FastQ.£{_DATE}"
cp "£{_WORKPATH}/RunInfo.xml" "£{_WORKPATH}/RunParameters.xml" "\£{_OUTPATH}/FastQ.£{_DATE}"

## generate a link to the output, set permissions and cleanup
[ -L "FastQ" ] && rm "FastQ"
#ln -s "Data/Intensities/FastQ.£{_DATE}" "FastQ"
ln -s "\£{_OUTPATH}/FastQ.£{_DATE}" "FastQ"

#setfacl -R -m u:rea62300:rwx "Data/Intensities/FastQ.£{_DATE}"
setfacl -R -m u:rea62300:rwx "\£{_OUTPATH}/FastQ.£{_DATE}"
EOF
chmod 750 £{_WORKPATH}/bcl2fastq.£{1}.£{_DATE}.sh
_OUTPATH2="/misc/data/NextSeq/$(/usr/bin/basename £{_WORKPATH})"
mkdir -p £{_OUTPATH2}/FastQ.£{_DATE}
cp £{_WORKPATH}/bcl2fastq.£{1}.£{_DATE}.sh £{_OUTPATH2}/FastQ.£{_DATE}/
#***# run job in screen mode and log output (-L)
# use DATE as unique identifier in case a <run> is executed more than once...
screen -L £{_WORKPATH}/screenlog.£{1}.£{_DATE} -S "run£{_RUNNUM}.£{_DATE}" £{_WORKPATH}/bcl2fastq.£{1}.£{_DATE}.sh
```

NEXTSEQcopy.sh:

```
#!/bin/bash
# by Michael Rehli, Febr 24, 2018
# edited on March 5,2018 to fix FastQC processing for '*'
# bash script to copy our fastq files from the NEXTSEQ directory into our rawData directory,
# and to run FASTQC
#
args=("$@")                                              # writes command line into array
if [ £{#args[@]} != 3 ] ; then                        # checkes whether enough arguments are given
        echo -e "\nMissing or too many arguments! \n"
        echo "Correct usage:"                              # prints to STDERR the correct usage
        echo "NEXTSEQcopy.sh <run number> <target subfolder in /misc/data/rawData> <sample name> "
        echo "Example:"
        echo "NEXTSEQcopy.sh 10 RNA/MOMACDC Mock_iDCs_7d_S3_R1_001"
        echo -e "To copy all fastq.gz files in the directory use '*' (including the '') as sample name. \n"
        exit
fi
# defining variables from input
RUN=£{args[0]}
TARGETFOLDER="/misc/data/rawData/£{args[1]}"
SAMPLE=£{args[2]}.fastq.gz
# finding the NextSeq Folder -
# each folder has a specific four-field structure:
# YYMMDD_machinename_NNNN_flowcellID, where NNNN is the
# four digit run number
# converting Run number to 4 digit string
case "£{#RUN}" in
                4)      _RUN="£{RUN}"
                                ;;
                3)      _RUN="0£{RUN}"
                                ;;
                2)      _RUN="00£{RUN}"
                                ;;
                1)      _RUN="000£{RUN}"
                                ;;
esac
# find the path to the dataset of the run
NEXTSEQPATH=$(find /misc/data/NextSeq -maxdepth 1 -type d  -name "*_NB551283_£{_RUN}*" |head -1)
[ -z "£{NEXTSEQPATH}" ] && {
printf "Sorry, no data found for NextSeqRun number £{RUN}\n\n"
```

```
exit 1 ;
}
# find the path to the subfolder containing the fastq data of the run
FASTQPATH=$(find "£{NEXTSEQPATH}" -maxdepth 1 -type d  -name "FastQ*" |head -1)
[ -z "£{FASTQPATH}" ] && {
printf "Sorry, no data found in the Folder for NextSeqRun number £{RUN}\n\n"
exit 1 ;
}
echo -e "\nNEXTSEQcopy.sh will move data from the /misc/data/NextSeq folder to the"
echo -e "given subfolder in /misc/data/rawData , and run FASTQC. \n"

if [ ! -d "$TARGETFOLDER" ]; then                        #looks for the desired output folder - if not available, creates it
        mkdir -p "£{TARGETFOLDER}/FastQC"
        echo "generating the folder $TARGETFOLDER"
        echo " "
fi
cp $FASTQPATH/$SAMPLE $TARGETFOLDER
#echo -e "Running fastqc \n"
for ENTRY in £{FASTQPATH}/£{SAMPLE}
        do
        SAMPLEBASE=£{ENTRY##*/}
        fastqc-0.11.7 -o "£{TARGETFOLDER}/FastQC" "£{TARGETFOLDER}/£{SAMPLEBASE}"
done
```

getBSFconvertSE.sh:

```
#!/bin/bash
# by Michael Rehli, Febr 5, 2018
# bash script to download and check md5 sum for bam-files produced at the BSF in Vienna
args=("$@")                                                # writes command line into array
if [ £{#args[@]} != 7 ] ; then                                # checkes whether enough parameters are given
        echo -e "\nMissing or too many arguments! \n"
        echo "Correct usage:"                                  # prints to STDERR the correct usage
        echo "getBSFconvertSE.sh <user name> <password> <flowcell ID> <lane number> <sample name> <output subdirectory in
        ↪  misc/data/rawdata> <new sample name>"
        echo "Example:"
        echo -e "getBSFconvertSE.sh mrehli xyz BSF_0411_HNH5NBBXX 4 S_10_ChIP_DD_S33767 chromatin/ChIP/MOMACDC 3dDCs_DSG+FA_R2_EGR2_A
        ↪  \n"
        exit
fi
# defining variables per default and operator input
WEBFOLDER="https://biomedical-sequencing.at/downloads/download/BSF_downloads/group_Michael_Rehli"
USER=£{args[0]}
PASSW=£{args[1]}
EXPERIMENT=£{args[2]}
LANE="£{args[3]}"
SAMPLE=£{args[4]}
RUN=$(echo $EXPERIMENT| cut -d'_' -f 2)
PREFIX=£{args[2]}_£{args[3]}_
BSFFOLDER="/misc/data/rawData/BSF/run£{RUN}"
echo -e "\ngetBSFconvert.sh will download data from the Biomedical Sequencing Facility in Vienna,"
echo -e "convert it into FASTQ format, run FASTQC.\n"
BSFBAM=$WEBFOLDER/£{args[0]}/£{args[2]}/$PREFIX£{args[4]}.bam
BSFMD5=$WEBFOLDER/£{args[0]}/£{args[2]}/$PREFIX£{args[4]}.bam.md5
OWNBAM=$BSFFOLDER/$PREFIX£{args[4]}.bam
OWNMD5=$BSFFOLDER/$PREFIX£{args[4]}.bam.md5
if [ ! -d "$BSFFOLDER" ]; then                       #looks for the desired BSF output folder - if not available, creates it
        mkdir "$BSFFOLDER"
        echo -e "generating the folder $BSFFOLDER \n"
fi
TARGETFOLDER="/misc/data/rawData/£{args[5]}"
TARGET=$TARGETFOLDER/£{args[6]}.fastq
if [ ! -d "$TARGETFOLDER" ]; then                        #looks for the desired output folder - if not available, creates it
        mkdir -p "£{TARGETFOLDER}/FastQC"
        echo -e "generating the folder $TARGETFOLDER \n"
fi
# downloading the .bam and .bam.md5 files
echo -e "downloading the bam-file for $SAMPLE\n"
curl -u $USER:$PASSW $BSFBAM -o $OWNBAM --insecure
echo -e "downloading the md5-file for $SAMPLE\n"
curl -u $USER:$PASSW $BSFMD5 -o $OWNMD5 --insecure
#comparing checksums created in Vienna and here
if [ $(md5sum "$OWNBAM" | cut -b-32) == $(cat "$OWNMD5") ]; then
    echo "MD5 checksum matches"
        # provided that the checksums match, proceed with converting into fastq, FastQC, and gzip
    echo -e "\nconverting the bam-file into fastq format \n"
    /usr/bin/java -jar /misc/software/ngs/picard/src/v2.17.3/picard.jar SamToFastq I=$OWNBAM FASTQ=$TARGET
    echo -e "running fastqc \n"
```

```
        fastqc-0.11.7 -o "£{TARGETFOLDER}/FastQC" $TARGET
        echo -e "zipping the fastq file \n"
        gzip $TARGET
else
        echo "MD5 checksum different! Try downloading again!"
fi
```

mapATAC.sh:

```
#!/bin/bash
# by Michael Rehli, Febr 23, 2018
# bash script to map ATAC-seq data, generate tagDirectories, coverage bigWigs,
# and to find ATAC-seq peaks
#
#setting homer environment
DIR_PKG="/misc/software/ngs"
PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin
export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH
# Defining the program versions (needs to be adjusted to the server/workstation used)
# rhskl17 version
SKEWER="/misc/software/ngs/skewer/skewer-0.2.2/skewer"
BOWTIE="/misc/software/ngs/bowtie/bowtie2-2.3.4-linux-x86_64/bowtie2"
SAMTOOLS="/misc/software/ngs/samtools/samtools-1.6/bin/samtools"
PICARD="/usr/bin/java -jar /misc/software/ngs/picard/src/v2.17.3/picard.jar"
IGVTOOLS="/usr/bin/java -Xmx4g -Djava.awt.headless=true -jar /misc/software/viewer/IGV/IGVTools_2.3.98/igvtools.jar"
# required input:
# -f <fastq or fastq.gz> forward read
# -r <fastq or fastq.gz> reverse read (if not available, will run as single end)
# -g <genome>  available genomes include hg19, GRCh38, mm9, mm10
# -o <output subdirectory> only subdirectories above ATAC-folder are given
# -n <sample name> used for sam, bigwig and tag directory
# optional:
# -t <number of threads>
# -s <path to adapter sequences> (or type NEXTERA to use Nextera-adapters)
# Set Script Name variable
SCRIPT=`basename £{BASH_SOURCE[0]}`
# Set fonts for Help.
NORM=`tput sgr0`
BOLD=`tput bold`
REV=`tput smso`
# USAGE function
function USAGE {
  echo -e \\n"Help documentation for £{BOLD}£{SCRIPT}.£{NORM}"\\n
  echo -e "£{REV}Basic usage:£{NORM}\n£{BOLD}$SCRIPT -g <genome> -f <fastq> -r <fastq> -o <subdirectory> -n <name> -t
 ↪  <threads>£{NORM}"\\n
  echo "Required:"
  echo "£{BOLD}-g£{NORM} <genome>  available genomes include hg19, GRCh38, mm9, mm10"
  echo "£{BOLD}-f£{NORM} <fastq or fastq.gz> forward read"
  echo "£{BOLD}-r£{NORM} <fastq or fastq.gz> reverse read (if not available, will run as single end)"
  echo "£{BOLD}-d£{NORM} <output subdirectory> only subdirectories above ATAC-folder are given"
  echo "£{BOLD}-n£{NORM} <sample name> used for sam, bigwig and tag directory"
  echo -e "£{BOLD}-t£{NORM} <number of threads> (optional/default 16)"\\n
  echo -e "£{BOLD}-s£{NORM} <path to adapter sequences> (or type NEXTERA to use Nextera-adapters)"\\n
  echo -e "Example: £{BOLD}$SCRIPT -g hg19 -f readA.fastq -r readB.fastq -d MOMACDC -n test -t 32 -s NEXTERA£{NORM}"\\n
  exit 1
}
#Check the number of arguments. If none are passed, print help and exit.
NUMARGS=$#
if [ $NUMARGS -eq 0 ]; then
  USAGE
fi
# Defining available genomes
GENOMES=(hg19 GRCh38 hg38 mm9 mm10)
# Set default number of threads to 16
THREADS=16
TRIM=0
# Parse command line options
while getopts :g:f:r:d:n:t:s:h OPTIONS; do
  case $OPTIONS in
    g)  #set option "g" - Bowtie will will only accept GRCh38 while HOMER needs hg38
      GENOME=$OPTARG
      HOMERGENOME=$OPTARG
      ;;
    f)  #set option "f"
```

```
      FASTQR1=$OPTARG
      ;;
    r)  #set option "r"
      FASTQR2=$OPTARG
      ;;
    d)  #set option "d"
      DIRECTORY=$OPTARG
      ;;
    n)  #set option "n"
      SAMPLENAME=$OPTARG
      ;;
    t)  #set option "t"
      THREADS=$OPTARG
      SAMTHREADS=$((THREADS-1))
      ;;
    s)  #set option "s"
      ADAPTER=$OPTARG
      TRIM=1
      ;;
    h)  #show help
      USAGE
      ;;
    \?) #unrecognized option - show USAGE
      echo -e \\n"Option -£{BOLD}$OPTARG£{NORM} not allowed."
      USAGE
      ;;
  esac
done
shift $((OPTIND-1))
# Sanity checks
# Check whether a reasonable number of cores is given
if [ $THREADS -lt 1 ] || [ "$THREADS" -gt 63 ]; then
        echo "Number of threads should be in the range of 1 - 63!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 9
fi
# Check whether GENOME is available
INARRAY=$(echo £{GENOMES[@]} | grep -o "$GENOME" | wc -w)
if [ $INARRAY == 0 ] ; then
      echo -e \\n"Genome -£{BOLD}$GENOME£{NORM} not available."
      echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
      exit 2
fi
GENOMEPATH=$GENOME
HOMERGENOME=$GENOME
BIGWIGGENOME=$GENOME
# path to chromosome sizes
CHROMSIZES="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes/$GENOME.chrom.sizes"
if [ $GENOME == "hg38" ] || [ $GENOME == "GRCh38" ] ; then
      GENOME="hg38"
      GENOMEPATH="GRCh38.PRI_p10"
      BIGWIGGENOME="GRCh38"
      HOMERGENOME="hg38"
      CHROMSIZES="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes/GRCh38.PRI_p10.chrom.sizes"
fi
#bowtie index
INDEX="$GENOMEPATH/$GENOME"
# Check whether one or two reads are given, whether files are available, non-identical and have the right extensions
if [ ! -f "$FASTQR1" ]; then
    echo "Fastq file (read 1) not found!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 3
fi
case "$FASTQR1" in
    *.fastq.gz) ;;
    *.fastq)    ;;
    *.fq.gz)    ;;
    *.fq)       ;;
    *)          echo "Fastq file (read 1) has wrong extension (should be fastq, fq, fastq.gz or fq.gz!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 4
    ;;
esac
if [ -z "$FASTQR2" ]; then
    SEQTYPE=1
    TYPE="SE"
    SKEWTYPE="any"
    echo -e \\n"Data processing for £{BOLD}single-end£{NORM} reads"
elif [ ! -f "$FASTQR2" ]; then
    echo "Fastq file (read 2) not found!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 5
```

```
elif [ "$FASTQR1" == "$FASTQR2" ]; then
    echo "Fastq files for read 1 and 2 are identical!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 6
else
        SEQTYPE=2
        TYPE="PE"
        SKEWTYPE="pe"
        echo -e \\n"Data Processing for £{BOLD}paired-end£{NORM} reads"
        case "$FASTQR2" in
       *.fastq.gz) ;;
       *.fastq)    ;;
       *.fq.gz)    ;;
       *.fq)       ;;
       *)          echo "Fastq file (read 2) has wrong extension (should be fastq, fq, fastq.gz or fq.gz!"
       echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
       exit 7
       ;;
    esac
fi
# Check wether the subfolders are already there - if not ask whether path should be made
if [ -z "$DIRECTORY" ]; then
        echo "Subdirectory is missing!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 7
fi
# rhskl17 version
MAPPINGDIR="/misc/data/processedData/mapping/chromatin/£{BIGWIGGENOME}/ATAC/£{DIRECTORY}"
TAGDIRDIR="/misc/data/processedData/tagDir/chromatin/£{BIGWIGGENOME}/ATAC/£{DIRECTORY}"
BIGWIGDIR="/misc/data/processedData/bigWig/chromatin/£{BIGWIGGENOME}/ATAC/£{DIRECTORY}"
if [ ! -d "$MAPPINGDIR" ]; then                        #looks for the mapping output folder - if not available, askes whether to make
↪  it
        echo "$MAPPINGDIR"
        read -p "Do you want to create this mapping folder (y/n)?" choice
        case "$choice" in
                  y|Y )         mkdir -p "$MAPPINGDIR"
                ;;
                  n|N )   echo "Didn't create the folder and stopped!"
                    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
                    exit 8
                ;;
                  * ) echo "invalid response";;
        esac
fi
if [ ! -d "$TAGDIRDIR" ]; then                        #looks for the tagDir output folder - if not available, askes whether to make
↪  it
        echo "$TAGDIRDIR"
        read -p "Do you want to create this tagDirectory folder (y/n)?" choice
        case "$choice" in
                  y|Y )         mkdir -p "$TAGDIRDIR"
                ;;
                  n|N )   echo "Didn't create the folder and stopped!"
                    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
                    exit 8
                ;;
                  * ) echo "invalid response";;
        esac
fi
if [ ! -d "$BIGWIGDIR" ]; then                        #looks for the bigWig output folder - if not available, askes whether to make
↪  it
        echo "$BIGWIGDIR"
        read -p "Do you want to create this bigWig folder (y/n)?" choice
        case "$choice" in
                  y|Y )         mkdir -p "$BIGWIGDIR"
                ;;
                  n|N )   echo "Didn't create the folder and stopped!"
                    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
                    exit 8
                ;;
                  * ) echo "invalid response";;
        esac
fi
# Check whether sample name is given
if [ -z "$SAMPLENAME" ]; then
        echo "Sample name is missing!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 9
fi
SAMPLENAME="£{SAMPLENAME// /_}"  # replaces spaces with underscores in sample name, just in case...
# Check if path for trimming is given or set for NEXTERA
if [ $TRIM == 1 ] && [ $ADAPTER == "NEXTERA" ]; then
    ADAPTER="/misc/software/ngs/skewer/adapterFiles/nextera_adapters.fa"
```

```
elif [ $TRIM == 1 ] && [ ! -f "$ADAPTER" ]; then
    echo "Adapter file not found!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 11
fi
if [ ! -d "£{MAPPINGDIR}/logs" ]; then                        #looks for the logs output folder - if not available, makes it
   mkdir "£{MAPPINGDIR}/logs"
fi
# If everything passed, create a log file for all the commands that are executed during the processing
LOGFILE="£{MAPPINGDIR}/logs/£{SAMPLENAME}.commands.txt"
touch $LOGFILE
# Function to append command lines into logfile and then execute
exe() { echo -e "\£ $@"\\n >> $LOGFILE; "$@" ; }
# Generating a random number file name for temporary files
TEMPFILES="/loctmp/$RANDOM.tmp"
# Optionally starting the pipeline by trimming the reads with skewer
if [ $TRIM == 1 ] ; then
   case "$SEQTYPE" in
   1 )  exe $SKEWER -f sanger -t $THREADS -m any -x $ADAPTER -o $TEMPFILES $FASTQR1
        FASTQR1="$TEMPFILES-trimmed.fastq"
        mv "$TEMPFILES-trimmed.log" "£{MAPPINGDIR}/logs/£{SAMPLENAME}.trimming.log"
        ;;
   2 )  exe $SKEWER -f sanger -t $THREADS -m pe -x $ADAPTER -o $TEMPFILES $FASTQR1 $FASTQR2
        FASTQR1="$TEMPFILES-trimmed-pair1.fastq"
        FASTQR2="$TEMPFILES-trimmed-pair2.fastq"
        mv "$TEMPFILES-trimmed.log" "£{MAPPINGDIR}/logs/£{SAMPLENAME}.trimming.log"
        ;;
   esac
fi
# Running Bowtie2
echo -e \\n"Running Bowtie2 on $SAMPLENAME."\\n
case "$SEQTYPE" in
        1 )  exe $BOWTIE --very-sensitive -p $THREADS -x $INDEX --met-file $MAPPINGDIR/logs/$SAMPLENAME.aln_metrics.txt -U $FASTQR1
        ↪  2> $MAPPINGDIR/logs/$SAMPLENAME.aln_rates.txt -S $TEMPFILES.sam
        ;;
        2 )  exe $BOWTIE --very-sensitive --no-discordant -p $THREADS -x $INDEX --met-file
        ↪  $MAPPINGDIR/logs/$SAMPLENAME.aln_metrics.txt -1 $FASTQR1 -2 $FASTQR2 2> $MAPPINGDIR/logs/$SAMPLENAME.aln_rates.txt -S
        ↪  $TEMPFILES.sam
    ;;
esac
# Sorting and reducing the SAM-file to q>10
echo -e "Sorting and reducing $SAMPLENAME to q>10."\\n
exe $SAMTOOLS view -h -S -q 10 $TEMPFILES.sam -@ $SAMTHREADS -o $TEMPFILES.red.sam
exe $SAMTOOLS sort -O sam -@ $SAMTHREADS -T $TEMPFILES -o $TEMPFILES.sorted.sam $TEMPFILES.red.sam
# Shifting the read positions for ATAC
echo -e "Shifting the read positions."\\n
exe myATACshiftSAM.pl "$TEMPFILES.sorted.sam" "$TEMPFILES.shifted" "$TYPE"
echo -e "Generating TagDir, BigWig, and TDF and finding Peaks."\\n
# Generating a HOMER tag directory
exe makeTagDirectory $TAGDIRDIR/$SAMPLENAME $TEMPFILES.shifted.sam -genome $HOMERGENOME -checkGC
# Making a UCSC browser track
exe makeUCSCfile $TAGDIRDIR/$SAMPLENAME -bigWig $CHROMSIZES -o $BIGWIGDIR/$SAMPLENAME.bigwig
# Making an IGV browser track
exe $IGVTOOLS count -e 150 $TEMPFILES.shifted.sam $BIGWIGDIR/$SAMPLENAME.cov.tdf $CHROMSIZES
# Finding peaks
exe findPeaks $TAGDIRDIR/$SAMPLENAME -region -size 150 -o auto
if [ "$SEQTYPE" == 2 ]; then
exe $PICARD CollectInsertSizeMetrics \
      I="$TEMPFILES.shifted.sam" \
      O="$TAGDIRDIR/$SAMPLENAME/insert_size_metrics.txt" \
      H="$TAGDIRDIR/$SAMPLENAME/insert_size_histogram.pdf" \
      W=500
fi
exe $SAMTOOLS view -bS -@ $SAMTHREADS -o "£{MAPPINGDIR}/£{SAMPLENAME}.shifted.bam" $TEMPFILES.shifted.sam
# removing tmp files
rm £{TEMPFILES}.*
```

findATACpeaksL4.sh:

```
#!/bin/bash
# by Michael Rehli, March 9, 2018

##################
# script for peak finding in ATACseq data
# will return three files - one with centered peak, one with broad region and one intersect

##################
```

```
#setting homer environment

DIR_PKG="/misc/software/ngs"

PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin

export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH

BEDTOOLS="/misc/software/ngs/bedtools/bedtools2-2.27.1/bin/bedtools"


args=("$@")                                              # writes command line into array

# required input:
#
# <tagdir> <outputdir>

# Set Script Name variable
SCRIPT=`basename £{BASH_SOURCE[0]}`

# Set fonts for Help.
NORM=`tput sgr0`
BOLD=`tput bold`
REV=`tput smso`

# USAGE function
function USAGE {
  echo -e \\n"Help documentation for £{BOLD}£{SCRIPT}.£{NORM}"\\n
  echo -e "£{REV}Basic usage:£{NORM}\n£{BOLD}$SCRIPT <tagdir> <outputdir>£{NORM}"\\n
  echo -e "Example: £{BOLD}$SCRIPT MO_18h /misc/data/analysis/someProject/ATACpeaks{NORM}"\\n
  exit 1
}

#Check the number of arguments. If none are passed, print help and exit.
NUMARGS=$#
if [ $NUMARGS -eq 0 ]; then
  USAGE
fi

TAGDIR="£{args[0]}"
PEAKDIR="£{args[1]}"
SAMPLEBASE=£{TAGDIR##*/}

echo £{TAGDIR}

# Stitching to regions with nucleosome size peak sizes with stringent fdr cut-off

findPeaks £{TAGDIR} -size 150 -minDist 250 -L 4 -fdr 0.00001 -region -o
↪→  "£{PEAKDIR}/£{SAMPLEBASE}.s150.md250.L4.fdr-5.region.peaks.txt"
pos2bed.pl "£{PEAKDIR}/£{SAMPLEBASE}.s150.md250.L4.fdr-5.region.peaks.txt" >
↪→  "£{PEAKDIR}/£{SAMPLEBASE}.s150.md250.L4.fdr-5.region.peaks.bed"

# centered factor peaks with 250bp peak sizes with stringent fdr cut-off and peak size equal minDist

findPeaks £{TAGDIR} -size 250 -minDist 250 -L 4 -fdr 0.00001 -style factor -o
↪→  "£{PEAKDIR}/£{SAMPLEBASE}.s250.md250.L4.fdr-5.factor.peaks.txt"
pos2bed.pl "£{PEAKDIR}/£{SAMPLEBASE}.s250.md250.L4.fdr-5.factor.peaks.txt" >
↪→  "£{PEAKDIR}/£{SAMPLEBASE}.s250.md250.L4.fdr-5.factor.peaks.bed"

# intersecting bed files

$BEDTOOLS intersect -a "£{PEAKDIR}/£{SAMPLEBASE}.s150.md250.L4.fdr-5.region.peaks.bed" -b
↪→  "£{PEAKDIR}/£{SAMPLEBASE}.s250.md250.L4.fdr-5.factor.peaks.bed" -u > "£{PEAKDIR}/£{SAMPLEBASE}.L4.intersect.peaks.bed"
bed2pos.pl "£{PEAKDIR}/£{SAMPLEBASE}.L4.intersect.peaks.bed" > "£{PEAKDIR}/£{SAMPLEBASE}.L4.intersect.peaks.txt"
```

myATACshiftSAM.pl:

```
#!/usr/bin/perl
use strict;
use warnings;
no warnings 'experimental::smartmatch';
# This program will take a BAM alignment file and adjust it as is appropriate for ATAC-seq:
# If the read is on the positive strand (as determined by the sam flag) it will add 4 to the start, and subtract 5 from the partner
↪→  start
```

```
# if the read is on the negative strand, 5 will subtracted from it's start and 4 added to its mate start.
# the length and read type will be adjusted in both cases and the read and quality string trimmed appropriately
# this is done because Tn5 has a 9bp binding site, and it binds in the middle.  Functionally that means that the DNA had to be
↪  accessible at least 4.5bp on either site of the insertion.
# to my understanding the 4 for positive and 5 for negative was chosen at random, that could be wrong
# for the negative strand the start position is the 5'-most, which doesn't actually change, it's the 3'-position that changes, and to
↪  modify that you only need to change the read sequence and size.
# example of how a BAM file will be altered
# Original:
# HWI-ST374:226:C24HPACXX:4:2214:15928:96004      99      chrI    427     255     101M    =       479     153 # should also have read
↪  and quality, but were omitted for readability
# HWI-ST374:226:C24HPACXX:4:2214:15928:96004      147     chrI    479     255     101M    =       427     -153 # should also have read
↪  and quality, but were omitted for readability
# Altered:
# HWI-ST374:226:C24HPACXX:4:2214:15928:96004      99      chrI    431     255     97M     =       474     144 # read and quality were
↪  moitted for readability, but they would each be trimmed 4bp
# HWI-ST374:226:C24HPACXX:4:2214:15928:96004      147     chrI    479     255     96M     =       431     -144 # read and quality were
↪  moitted for readability, but they would each be trimmed 5bp
# make sure the right arguments are being passed
if (@ARGV < 3) {
        print STDERR "usage: myATACshiftSAM.pl <input BAM/SAM file> <outputfilename> <PE or SE>\n";
        exit;
}
my $bamFile = $ARGV[0];
my $out = $ARGV[1];
my $samFile = $out.".sam";
my $type = $ARGV[2];
my $samtools = "samtools-1.6" ;
# only reads mapping with a quality of 10 will be considered
if ($bamFile=~m/.bam£/){
    die "cannot open £bamFile.\n" unless(open BAM,"£samtools view -h -q 10 £bamFile |");
}elsif($bamFile=~m/.sam£/){
    die "cannot open £bamFile.\n" unless(open BAM,"£samtools view -h -S -q 10 £bamFile |");
}else{
    die "Input alignment file is not a .sam or .bam file\n";
}
die "cannot open £samFile.\n" unless(open TEMP,">£samFile");
my %flags=samFlags(); # this is a little sub that generates a hash with keys of 'unmapped', 'positive' and 'negative'; that is, if the
↪  read maps, and if so, to which strand, regardless of anything else.
while(<BAM>){
    chomp;  # removing new line
    # adding the header lines (if you used -h in the samools command), we'll want it later, and going on to the reads
    print TEMP $_."\n" if(/^(\@)/);
    next if(/^(\@)/);
    my @line = split(/\t/);  # splitting SAM line into array
    # skip reads if both both pairs were not
    # the specific read is unmapped,
    # or map to the mitochondria.
    next unless ($line[1] ~~ $flags{'properlyPaired'} || $type eq "SE");
    #next if(£line[2] ~~ m/^chrM/ || £line[2] ~~ m/_random£/ || £line[2] ~~m/^chrUn/);
    #next if(£line[2] ~~ m/_random£/ || £line[2] ~~m/^chrUn/);
    if($line[1] ~~ $flags{'negative'}){ # if the read is mapped to the reverse strand
        $line[7]=$line[7]+4 unless ($type eq "SE"); # move the mate start 4 bp
        $line[8]=$line[8]+9 unless ($type eq "SE"); # adjust the inferred insert size
        $line[9] =substr($line[9],0,-5); # remove 5 bp from the reads
        $line[10] =substr($line[10],0,-5);# remove the 5bp from quality as well
        # this is ugly, but this is a way to change the CIGAR to reflect that I'm shortening the read
        # for the negative stran I want to subtract the 5p from the last entry (which should be M, for match)
        $line[5] = &CIGARtrim($line[5],'-',5); # from the CIGAR string, trim 5, and we're on the negtive strand
    }elsif($line[1] ~~ $flags{'positive'}){ # if the read is mapped to the positive strand
        $line[3]=$line[3]+4 ; # move the start 4 bp
        $line[8]=$line[8]-9 unless ($type eq "SE");# adjust the inferred insert size for paired-end reads
        $line[9] =substr($line[9],4);# remove 4bp from the reads
        $line[10] =substr($line[10],4);# remove the 4bp from quality as well
        # this is ugly, but this is a way to change the CIGAR to reflect that I'm shortening the read
        # for the positive strand I want to subtract the 4 from the first entry (which should be M, for match)
        $line[5] = &CIGARtrim($line[5],'+',4); # from the CIGAR string, trim 4, and we're on the positive strand
    }
    print TEMP join("\t",@line)."\n";
}
close BAM;
close TEMP;
exit;
##### Sub routines #####
# this generates a hash filled with the most common sam flags and wether the reads are mapped, and if so to which strand
# NOTE this does not address any of the other issues that can be interpretted from the flag (correct orientation within or ourside of
↪  the insert size)
sub samFlags{
    my %flags=(
        'unmapped' => [133,165,181,101,117,69,77,141],
        'properlyPaired'=>[99,147,83,163,81,161,97,145],
        'improperInsert' => [81,161,97,145],
        'positive' => [0,73,89,137,99,163,67,131,161,97,65,129],
        'negative' => [16,121,153,185,147,83,115,179,81,145,113,177]
        );
    return (%flags);
}
# this cuts off the first number and letter from a CIGAR string
# and returns the trimmed CIGAR plus the number and letter trimmed off
sub trimFirstCIGARunit{
    my $cigar = shift;
    my $firstChar = (split/\d+/, $cigar)[1];
    #print STDERR "£cigar\n£firstChar\n";
    my $firstNum = (split/£firstChar/, $cigar)[0];
    my $length =length($firstNum.$firstChar);
    return(substr($cigar,$length), $firstNum, $firstChar);
}
# this cuts off the last number and letter from a CIGAR string
# and returns the trimmed CIGAR plus the number and letter trimmed off
sub trimLastCIGARunit{
    my $cigar = shift;
    my $lastChar = substr($cigar,-1); # split was acting weird, so this should grab the last character
    my $lastNum = (split/[MINDS]/, $cigar)[-1]; # these are the possible letters equalling Match, Insert, N, Deletion, Shoft clip
    my $length =length($lastNum.$lastChar);
    return(substr($cigar,0,(-1*$length)), $lastNum, $lastChar);
}
# this will take a cigar string, the strand the read maps to and a number of reads to cut off
# it returns the properly trimmed CIGAR string
sub CIGARtrim{
    my ($cigar, $strand, $num) = @_;
    my $toReturn;
        if ($strand ~~ '-'){
            my ($cigarToKeep, $lastNum, $lastChar) = &trimLastCIGARunit($cigar);
            my $newLastNum = $lastNum - $num;

            if ($lastChar ~~ 'D'){ # skip to the next letter because the deletion is completely skipped over
                ($cigarToKeep, $lastNum, $lastChar) = &trimLastCIGARunit($cigarToKeep);
                $newLastNum = $lastNum - $num;
            }
            if ($newLastNum==0){
                my $nextChar = substr($cigarToKeep,-1);
                if ($nextChar ~~ 'D'){ # skip to the next letter because a read shouldn't end in a deletion
                    ($cigarToKeep, $lastNum, $lastChar) = &trimLastCIGARunit($cigarToKeep);
                }
                $toReturn = $cigarToKeep;
            }elsif($newLastNum<0){ # Not everything has been trimmed off, so iterate over this again
                $toReturn=&CIGARtrim($cigarToKeep,$strand,-1*$newLastNum);
            }else{
                $toReturn=$cigarToKeep.$newLastNum.$lastChar;
            }
        }elsif($strand ~~ '+'){
           # print STDERR "£cigar\n";
            my ($cigarToKeep, $firstNum, $firstChar) = &trimFirstCIGARunit($cigar);
            my $newFirstNum = $firstNum - $num;
            if ($firstChar ~~ 'D'){ # skip to the next letter because the deletion is completely skipped over
                ($cigarToKeep, $firstNum, $firstChar) = &trimFirstCIGARunit($cigarToKeep);
                $newFirstNum = $firstNum - $num;
            }
            if ($newFirstNum==0){
                my $nextChar = substr($cigarToKeep,-1);
                if ($nextChar ~~ 'D'){ # skip to the next letter because a read shouldn't end in a deletion
                    ($cigarToKeep, $firstNum, $firstChar) = &trimFirstCIGARunit($cigarToKeep);
                }
                $toReturn = $cigarToKeep;
            }elsif($newFirstNum<0){ # Not everything has been trimmed off, so iterate over this again
                $toReturn=&CIGARtrim($cigarToKeep,$strand,-1*$newFirstNum);
            }else{
                $toReturn=$newFirstNum.$firstChar.$cigarToKeep;
            }
        }else{
            die ("strand in CIGARtrim must be '+' or '-'\n");
        }
        return ($toReturn);
}
```

mySummarizeMotifResults.pl:

```
#!/usr/bin/perl

use warnings;
use Math::VecStat qw(min max average sum);
use Statistics::Basic qw(median);

if (@ARGV < 1) {
        print STDERR "\n\tUsage: mySummarizeMotifResults.pl <list of knownResult.txt files> [options]\n";
        print STDERR "\n\tProgram summarizing motif enrichments across multiple searches\n";
        print STDERR "\n\tOptions:\n";
        print STDERR "\n\t-outDir <path> (output directory, default is tmp) \n";
        print STDERR "\n\t-hc (cluster columns in heatmap, default is not) \n";
        print STDERR "\n\t-minp minimum p-value to be included (default 0.0001)\n";
        print STDERR "\n\t-minr minimum enrichment (-fold above expected) to be included (default 2)\n";
        print STDERR "\n\t-limit upper/lower limit for heatmap scale (default 2)\n";
        print STDERR "\n\t-maxnorm normalizes enrichment to the max enrichment value across samples\n\n";

        print STDERR "\n\tadditional options for output:\n";
        print STDERR "\n\t-pValueTables (pValue tables for individual colums instead of a single table)\n";
        print STDERR "\n\t-enrichTables (enrichment tables for individual colums instead of a single table)\n";

        exit;
}

my $rand = rand();
my $count = 0;
my $outDir = "/misc/data/tmp/";
my $tempFile = "/loctmp/" . $rand . ".file.tmp";
my $tempSort = "/loctmp/" . $rand . ".sort.tmp";
my $tempSort2 = "/loctmp/" . $rand . ".sort2.tmp";
my $tempout0 = "/loctmp/" . $rand . ".out0.tmp";
my $tempout1 = "/loctmp/" . $rand . ".out1.tmp";
my $tempout2 = "/loctmp/" . $rand . ".out2.tmp";
my $temptempout0 = "/loctmp/" . $rand . ".out0.tmp.tmp";
my $temptempout1 = "/loctmp/" . $rand . ".out1.tmp.tmp";
my $temptempout2 = "/loctmp/" . $rand . ".out2.tmp.tmp";
my $tempIDs = "/loctmp/" . $rand . ".IDs.tmp";
my $tempout = "/loctmp/" . $rand . ".out.tmp";
my $tmpScript = "/loctmp/" . $rand . ".R";
my $ratio = 1;
my $minThresh = 0.0001;
my $maxThresh = 2;
my $limit = 2;
my $maxnorm = 0;
my $pValueTable = 0;
my $enrichTable = 0;
my $pos = 0;
my $qpos = 0;
my $hc = 0;
my $names = 0;
my $namesFile = '';
my @sampleNames;


for (my $i=0;$i<@ARGV;$i++) {
        if ($ARGV[$i] eq '-minp') {
                $minThresh = $ARGV[++$i];
        } elsif ($ARGV[$i] eq '-minr') {
                $maxThresh = $ARGV[++$i];
        } elsif ($ARGV[$i] eq '-limit') {
                $limit = $ARGV[++$i];
        } elsif ($ARGV[$i] eq '-maxnorm') {
                $maxnorm = 1 ;
        } elsif ($ARGV[$i] eq '-pValueTables') {
                $pValueTable = 1 ;
        } elsif ($ARGV[$i] eq '-enrichTables') {
                $enrichTable = 1 ;
        } elsif ($ARGV[$i] eq '-hc') {
                $hc = 1 ;
        } elsif ($ARGV[$i] eq '-namesFile') {
                $namesFile = $ARGV[++$i] ;
                $names = 1;
        } elsif ($ARGV[$i] eq '-outDir') {
                $outDir = $ARGV[++$i] ;
                $outDir .= '/' unless $outDir =~ m(/£);
        }
}


my $out0 = $outDir . "pValueTable.txt";
my $out1 = $outDir . "qValueTable.txt";
my $out2 = $outDir . "ratioTable.txt";
my $pq = $outDir . "pqValueTable.txt";
my $finalout = $outDir . "cleanedRatioTable.txt";
my $finalout2 = $outDir . "cleanedqValueTable.txt";
my $l1 = $limit;
my $l2 = ( int ($limit / 0.03) / 100 ) + 0.001 ;
my $l3 = int ($limit / 0.03) / 100 ;
my $l4 = - int ($limit / 0.03) / 100 ;
my $l5 = - ( int ($limit / 0.03) / 100 ) - 0.001 ;
my $l6 = - $limit;



if ($names == 1) {
        $namesFile =~ s/\R//g;
        open NF, $namesFile or die "Can't open £namesFile for read: £!";
        while (<NF>) {
                push (@sampleNames, $_);
        }
        close NF;
}

open FH, $ARGV[0] or die "Can't open £ARGV[0] for read: £!";
my @filenames;
while (<FH>) {
    push (@filenames, $_);
}
close FH or die "Cannot close £ARGV[0]: £!";

my $clusterCount = scalar(@filenames);


#print STDERR @filenames;

print STDERR "\n\tno. of clusters: £clusterCount\n";


foreach $file ( @filenames ) {

        $file =~ s/\R//g;
           `(head -n 1 £file && tail -n +2 £file | sort) | cut -f1,3,5,7,9 > £tempFile`;

        open IN, $tempFile or die "Could not open file: \"£file\"\n";

        $count++;

           open OUT0, ">£tempout0";
           open OUT1, ">£tempout1";
        open OUT2, ">£tempout2";
        if ($count == 1) {
            open OUT4, ">£tempIDs";
            }


        while ( <IN> ) {
        chomp;
        s/\r//g;
         my @line= split /[%\t]+/;

           if ($count == 1) {
                   print OUT4 "£line[0]\n";
        }

        if ($. == 1) {

                print OUT0 "cluster£count\n";
                print OUT1 "cluster£count\n";
                print OUT2 "cluster£count\n";

        } else {
                if ($line[4] == 0) {
                        $ratio = 1 ;
                        } else {
                    $ratio = $line[3] / $line[4];
                    }
                   print OUT0 "£line[1]\n";
                   print OUT1 "£line[2]\n";
                print OUT2 "£ratio\n";
                }
        }

        close IN;
        close OUT0;
        close OUT1;
           close OUT2;
           if ($count == 1) {
                   close OUT4;
            `cp -f £tempIDs £temptempout0 ; cp -f £tempIDs £temptempout1 ; cp -f £tempIDs £temptempout2`;
                   }

           `paste £temptempout0 £tempout0 > £out0 ; paste £temptempout1 £tempout1 > £out1 ; paste £temptempout2 £tempout2 > £out2`;
        `cp -f £out0 £temptempout0 ; cp -f £out1 £temptempout1 ; cp -f £out2 £temptempout2`;
        `rm -f £tempout0 ; rm -f £tempout1 ; rm -f £tempout2`;

}

`paste £out1 £out2 > £tempout`;
`paste £out0 £out1 > £pq`;

# print STDERR @sampleNames;


if ($pValueTable == 0 && $enrichTable == 0) {



open TMP, $tempout;

open OP, ">£finalout";
open OQ, ">£finalout2";

        while ( <TMP> ) {
                chomp;
                s/\r//g;
                my @line= split /\t/;
                my @array;
                my @data;

                for ($i = 1; $i < ($clusterCount+1) ; $i++) {
                        $array[($i-1)] = $line[$i] ;
                        }
                for ($i = ($clusterCount+2); $i < ($clusterCount+$clusterCount+2) ; $i++) {
                        $data[($i-$clusterCount-2)] = $line[$i] ;
                        }


                if ($. == 1) {
                        print OP $line[$clusterCount+1] ;
                        print OQ $line[$clusterCount+1] ;
                        print STDERR $line[$clusterCount+1] ;
                        if ($names ==1) {
                                 for ($i = 0; $i < $clusterCount ; $i++) {
                                         $sampleNames[$i] =~ s/\R//g;
                                        print OP "\t£sampleNames[£i]" ;
                                        print OQ "\t£sampleNames[£i]" ;
                                }
                        } else {
                                for ($i = ($clusterCount+2); $i < ($clusterCount+$clusterCount+2) ; $i++) {
                                        print OP "\t£line[£i]" ;
                                        print OQ "\t£line[£i]" ;
                                }
                        }
                        print OP "\n" ;
                        print OQ "\n" ;
                } else {
                        my $med = 0;
                        $med = median(@array) ;
                        my $ave = average(@array);
                        my $min = min(@array);
                        my $max = max(@data);
                        if ($max == 1) {$max = 1.001;}
                        my $logmaxfactor = 2 / log2($max);

                        if ($min <= $minThresh && $max >=$maxThresh) {
                                print OP $line[$clusterCount+1] ;
                                print OQ $line[$clusterCount+1] ;
                                for ($i = ($clusterCount+2); $i < ($clusterCount+$clusterCount+2) ; $i++) {
                                        $y= $i-($clusterCount+1);
                                        if ($maxnorm == 0) {
                                                print OP "\t£line[£i]" ;
                                                print OQ "\t£line[£y]" ;
                                        } else {
                                                $normvalue = log2($line[$i]+0.001) * $logmaxfactor;
                                                print OP "\t£normvalue" ;
                                                print OP "\t£line[£y]" ;
                                        }
                                }
                        print OP "\n" ;
                        print OQ "\n" ;
                        }
                }
        }
        close TMP;
        close OP;
        close OQ;

        open OUT, ">£tmpScript";
        print OUT "setwd(\"£outDir\")\n";
        print OUT "library(gplots)\n";
        print OUT "library(RColorBrewer)\n";
        print OUT "data <- read.table(\"£finalout\", header=T, sep=\"        \", check.names=T, row.names=\"Motif.Name\")\n";
        print OUT "x  <- data.matrix(data)\n";
        if ($maxnorm == 0) {
        print OUT "logx <- log2(x + 0.0001)\n";
        } else {
        print OUT "logx <- data.matrix(data)\n";
        }
        if ($hc == 1) {
        print OUT "hc <- hclust(dist(t(logx), method = \"manhattan\"), method=\"ward.D2\")\n"; }
        print OUT "hr <- hclust(dist(logx, method = \"manhattan\"), method=\"ward.D2\")\n";
        print OUT "mycol <- colorRampPalette(c(\"darkcyan\",\"white\",\"chocolate3\"))(299)\n";
        print OUT "col_breaks = c(seq(£l6,£l5,length=100), seq(£l4,£l3,length=100), seq(£l2,£l1,length=100))\n";
        print OUT "pdf(file=\"ratioTable_clusteredHeatmap.pdf\", height=8, width=8)\n";
        if ($hc == 1) {
        print OUT "heatmap.2(logx, Rowv=as.dendrogram(hr), Colv=as.dendrogram(hc), col = mycol, breaks=col_breaks, scale=\"none\",
        ↪  margins=c(5,5), cexRow=0.5, cexCol=1.0, key=TRUE, density.info=\"none\", trace=\"none\", symm=FALSE,
        ↪  symkey=F,symbreaks=TRUE)\n";
        } else {
        print OUT "heatmap.2(logx, Rowv=as.dendrogram(hr), Colv=NA, dendrogram = \"row\", col = mycol, breaks=col_breaks,
        ↪  scale=\"none\", margins=c(5,5), cexRow=0.5, cexCol=1.0, key=TRUE, density.info=\"none\", trace=\"none\", symm=FALSE,
        ↪  symkey=F,symbreaks=TRUE)\n";
         }
         print OUT "dev.off()\n";
#        print OUT "write.table(logx[rev(hr\£labels[hr\£order]), hc\£labels[hc\£order]], file = \"RatioTable_clearedClustered.txt\",
↪   sep = \"        \", col.names=NA)\n";
        close OUT;

        print STDERR "\t\trunning hierarchical clustering in R\n";

        `R --no-save < "£tmpScript"`;

} elsif ($enrichTable == 1) {

        my $filelength = $clusterCount +1;

        for ($j = 1; $j <= $clusterCount ; $j++) {
                $pos = $j + 1;
                $qpos = $j + $clusterCount +2;
                `myFilterFile.pl £tempout -header -column £pos -upperlimit £minThresh > £tempSort`;
                my $qoutputFile = "cluster" . $j . "enrichmentTable.txt";
                my $startCol = $filelength + 1;
                my $endCol = $filelength *2 ;

                `cut -f£startCol-£endCol £tempSort > £tempSort2`;


                open TMPS, $tempSort2;

                open OPF, ">£qoutputFile";

                while ( <TMPS> ) {
                        chomp;
                        s/\r//g;
                        my @line= split /\t/;
                        my @array;

                        for ($i = 1; $i <= $clusterCount ; $i++) {
                                $array[($i-1)] = $line[$i] ;
                                }

                        if ($. == 1) {
                                print OPF $line[0] ;
                                if ($names ==1) {
                                         for ($i = 0; $i < $clusterCount ; $i++) {
                                                 $sampleNames[$i] =~ s/\R//g;
                                                print OP "\t£sampleNames[£i]" ;
                                        }
                                } else {
                                        for ($i = 1; $i <= $clusterCount ; $i++) {
                                                print OPF "\t£line[£i]" ;
                                                }
                                        }
                                print OPF "\n" ;
                        } else {
                                my $med = 0;
                                $med = median(@array) ;
                                my $ave = average(@array);
                                my $max = max(@array);
                                my $logmaxfactor = 2 / log2($max);

#                                  print STDERR "\nmaximum ratio / factor : £max / £logmaxfactor , average £ave, median £med, value
↪   £line[£j] ";
                                if ($line[$j] > $med && $line[$j] > $ave) {
                                        print OPF $line[0] ;
                                        for ($i = 1; $i <= $clusterCount ; $i++) {
                                                if ($maxnorm == 0) {
                                                        print OPF "\t£line[£i]" ;
                                                } else {
                                                        $normvalue = log2($line[$i]+0.001) * $logmaxfactor ;
                                                        print OPF "\t£normvalue" ;
                                                }
                                        }
                                        print OPF "\n" ;
                                }
                        }
                }

        close TMPS;
        close OPF;

        open OUT, ">£tmpScript";
        print OUT "setwd(\"£outDir\")\n";
        print OUT "library(gplots)\n";
        print OUT "library(RColorBrewer)\n";
        print OUT "data <- read.table(\"£qoutputFile\", header=T, sep=\"        \", check.names=T, row.names=\"Motif.Name\")\n";
        print OUT "x  <- data.matrix(data)\n";
        if ($maxnorm == 0) {
        print OUT "logx <- log2(x + 0.0001)\n";
        } else {
        print OUT "logx <- data.matrix(data)\n";
        }
        if ($hc == 1) {
        print OUT "hc <- hclust(dist(t(logx), method = \"manhattan\"), method=\"ward.D2\")\n"; }
        print OUT "hr <- hclust(dist(logx, method = \"manhattan\"), method=\"ward.D2\")\n";
        print OUT "mycol <- colorRampPalette(c(\"darkcyan\",\"white\",\"chocolate3\"))(299)\n";
        print OUT "col_breaks = c(seq(£l6,£l5,length=100), seq(£l4,£l3,length=100), seq(£l2,£l1,length=100))\n";
        print OUT "pdf(file=\"enrichmentHeatmapCluster£j.pdf\", height=4, width=8)\n";
        if ($hc == 1) {
        print OUT "heatmap.2(logx, Rowv=as.dendrogram(hr), Colv=as.dendrogram(hc), col = mycol, breaks=col_breaks, scale=\"none\",
        ↪   margins=c(5,5), cexRow=0.5, cexCol=1.0, key=TRUE, density.info=\"none\", trace=\"none\", symm=FALSE,
        ↪   symkey=F,symbreaks=TRUE)\n";
        } else {
        print OUT "heatmap.2(logx, Rowv=as.dendrogram(hr), Colv=NA, dendrogram = \"row\", col = mycol, breaks=col_breaks,
        ↪   scale=\"none\", margins=c(5,5), cexRow=0.5, cexCol=1.0, key=TRUE, density.info=\"none\", trace=\"none\", symm=FALSE,
        ↪   symkey=F,symbreaks=TRUE)\n";
        }
        print OUT "dev.off()\n";
        close OUT;

        print STDERR "\t\trunning hierarchical clustering in R\n";

        `R --no-save < "£tmpScript"`;

        }

} else {

         my $filelength = $clusterCount +1;
        for ($i = 1; $i <= ($clusterCount) ; $i++) {
                $pos = $i + 1;
                $qpos = $i + $clusterCount +2;
                   `myFilterFile.pl £pq -header -column £qpos -upperlimit £minThresh > £tempSort2`;
                my $qoutputFile = "cluster" . $i . "pValueTable.txt";
            `cut -f1-£filelength £tempSort2 > £qoutputFile`;

                open OUT, ">£tmpScript";
                print OUT "setwd(\"£outDir\")\n";
                print OUT "library(gplots)\n";
                print OUT "library(RColorBrewer)\n";
                print OUT "data <- read.table(\"£qoutputFile\", header=T, sep=\"        \", check.names=T,
               ↪  row.names=\"Motif.Name\")\n";
                print OUT "x  <- data.matrix(data)\n";
                print OUT "logx <- -log10(x)\n";
                if ($hc == 1) {
                print OUT "hc <- hclust(dist(t(logx), method = \"manhattan\"), method=\"ward.D2\")\n"; }
                print OUT "hr <- hclust(dist(logx, method = \"manhattan\"), method=\"ward.D2\")\n";
                print OUT "mycol <-
               ↪  colorRampPalette(c(\"floralwhite\",\"deepskyblue4\",\"darkviolet\",\"firebrick\",\"gold\"))(499)\n";
                print OUT "col_breaks = c(seq(0,5,length=100), seq(5.01,10,length=100), seq(10.01,15,length=100),
               ↪  seq(15.01,20,length=100), seq(20.01,25,length=100))\n";
                print OUT "pdf(file=\"pValueHeatmapCluster£i.pdf\", height=8, width=8)\n";
                if ($hc == 1) {
                print OUT "heatmap.2(logx, Rowv=as.dendrogram(hr), Colv=as.dendrogram(hc), col = mycol, breaks=col_breaks,
               ↪  scale=\"none\", margins=c(5,5), cexRow=0.5, cexCol=1.0, key=TRUE, density.info=\"none\", trace=\"none\",
               ↪  symm=FALSE, symkey=F,symbreaks=TRUE)\n";
                } else {
                print OUT "heatmap.2(logx, Rowv=as.dendrogram(hr), Colv=NA, dendrogram = \"row\", col = mycol, breaks=col_breaks,
               ↪  scale=\"none\", margins=c(5,5), cexRow=0.5, cexCol=1.0, key=TRUE, density.info=\"none\", trace=\"none\",
               ↪  symm=FALSE, symkey=F,symbreaks=TRUE)\n";
                }
                print OUT "dev.off()\n";
                print OUT "dev.off()\n";
#                print OUT "write.table(logx[rev(hr\£labels[hr\£order]), hc\£labels[hc\£order]], file =
↪   \"cluster.£i.pValueTable.short.txt\", sep = \"        \", col.names=NA)\n";
                close OUT;

        print STDERR "\t\trunning hierarchical clustering in R\n";
        `R --no-save < "£tmpScript"`;
        }
}

`rm -f \"£tmpScript\" `;

`rm -f /loctmp/£rand.*`;

sub log2 {
        my $n = shift;
        return log($n)/log(2);
    }

sub log10 {
        my $n = shift;
        return log($n)/log(10);
    }

exit;
```

mapRNAseq.sh:

```
#!/bin/bash
# by Michael Rehli, Febr 23, 2018
# bash script to map RNA-seq data, generate a count table, and coverage bigWigs
# will only keep unique.aligned.bam, bigwig and tdf files as well as gene count data
# IF YOU REQUIRE MULTIMAPPERS, OR ANY OTHER ORIGINAL FILES, PLEASE RUN STAR MANUALLY
# modified by MR, March16, 2018 to incorporate genome GRCm38
#setting homer environment
DIR_PKG="/misc/software/ngs"
PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin
export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH
# Defining the program versions (needs to be adjusted to the server/workstation used)
# rhskl17 version
STAR="/misc/software/ngs/STAR/v2.5.3a/STAR/source/STAR"
SAMTOOLS="/misc/software/ngs/samtools/samtools-1.6/bin/samtools"
PICARD="/usr/bin/java -jar /misc/software/ngs/picard/src/v2.17.3/picard.jar"
IGVTOOLS="/usr/bin/java -Xmx4g -Djava.awt.headless=true -jar /misc/software/viewer/IGV/IGVTools_2.3.98/igvtools.jar"
INDEXSDIR="/misc/software/ngs/genome/index/STAR"
# required input:
#
# -f <fastq or fastq.gz> forward read
# -r <fastq or fastq.gz> reverse read (if not available, or set "NA" will run as single end)
# -g <genome>  available genomes include hg19, GRCh38, GRCm38
# -l <read length>
# -o <output subdirectory> only subdirectories above RNA-folder are given
# -n <sample name> used for sam, bigwig and tag directory
#
# optional:
#
# -t <number of threads>
# -c <read length> trim reads to the given length
# Set Script Name variable
SCRIPT=`basename £{BASH_SOURCE[0]}`
# Set fonts for Help.
NORM=`tput sgr0`
BOLD=`tput bold`
REV=`tput smso`
# USAGE function
function USAGE {
  echo -e \\n"Help documentation for £{BOLD}£{SCRIPT}.£{NORM}"\\n
  echo -e "£{REV}Basic usage:£{NORM}\n£{BOLD}$SCRIPT -g <genome> -f <fastq> -r <fastq> -o <subdirectory> -n <name> -t <threads> -c
 ↪  <trim length>£{NORM}"\\n
  echo "Required:"
  echo "£{BOLD}-g£{NORM} <genome>  available genomes include hg19, GRCh38, GRCm38"
  echo "£{BOLD}-f£{NORM} <fastq or fastq.gz> forward read"
  echo "£{BOLD}-r£{NORM} <fastq or fastq.gz> reverse read (if not available, or set \"NA\" will run as single end)"
  echo "£{BOLD}-d£{NORM} <output subdirectory> only subdirectories above RNA-folder are given"
  echo "£{BOLD}-n£{NORM} <sample name> used for sam, bigwig and tag directory"
  echo "£{BOLD}-t£{NORM} <number of threads> (optional/default 16)"
  echo "£{BOLD}-l£{NORM} <read length> used for index"
  echo "£{BOLD}-c£{NORM} <trim length> trims Fastq file to the given length before processing (optional)"
  echo "Available indices:"
  for entry in $INDEXSDIR/*
        do
    echo $entry
  done
  echo -e \\n"Example: £{BOLD}$SCRIPT -g hg19 -l 50 -f read1.fastq.gz -r read2.fastq.gz -d test -n testrun -t 24£{NORM} -c 43"\\n
  exit 1
}
#Check the number of arguments. If none are passed, print help and exit.
NUMARGS=$#
if [ $NUMARGS -eq 0 ]; then
  USAGE
fi
# Defining available genomes
GENOMES=(hg19 GRCh38 hg38 GRCm38)
# Set default number of threads to 16
THREADS=16
TRIM=0
# Parse command line options
while getopts :g:f:r:d:n:t:l:c:h OPTIONS; do
  case $OPTIONS in
    g)  #set option "g" - Bowtie will will only accept GRCh38 while HOMER needs hg38
      GENOME=$OPTARG
      HOMERGENOME=$OPTARG
      ;;
    f)  #set option "f"
      FASTQR1=$OPTARG
      ;;
    r)  #set option "r"
      FASTQR2=$OPTARG
      ;;
    d)  #set option "d"
      DIRECTORY=$OPTARG
      ;;
    n)  #set option "n"
      SAMPLENAME=$OPTARG
      ;;
    t)  #set option "t"
      THREADS=$OPTARG
      SAMTHREADS=$((THREADS-1))
      ;;
    l)  #set option "l"
      LENGTH=$OPTARG
      ;;
    c)  #set option "c"
      TRIMLENGTH=$OPTARG
      TRIM=1
      ;;
    h)  #show help
      USAGE
      ;;
    \?) #unrecognized option - show USAGE
      echo -e \\n"Option -£{BOLD}$OPTARG£{NORM} not allowed."
      USAGE
      ;;
  esac
```

```
done
shift $((OPTIND-1))
# Sanity checks
# Check whether a reasonable number of cores is given
if [ $THREADS -lt 1 ] || [ "$THREADS" -gt 63 ]; then
        echo "Number of threads should be in the range of 1 - 63!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 9
fi
# Check whether GENOME is available
INARRAY=$(echo £{GENOMES[@]} | grep -o "$GENOME" | wc -w)
if [ $INARRAY == 0 ] ; then
      echo -e \\n"Genome -£{BOLD}$GENOME£{NORM} not available."
      echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
      exit 2
fi
GENOMEPATH=$GENOME
BIGWIGGENOME=$GENOME
# path to chromosome sizes
CHROMSIZES="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes/$GENOME.chrom.sizes"
# STAR index
INDEX="$INDEXSDIR/index_£{GENOME}_£{LENGTH}"
if [ $GENOME == "hg38" ] || [ $GENOME == "GRCh38" ] ; then
      GENOME="hg38"
      INDEX="$INDEXSDIR/index_GRCh38.PRI.p10_$LENGTH"
      BIGWIGGENOME="GRCh38"
      CHROMSIZES="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes/GRCh38.PRI_p10.chrom.sizes"
fi
if [ $GENOME == "mm10" ] || [ $GENOME == "GRCm38" ] ; then
      GENOME="GRCm38"
      INDEX="$INDEXSDIR/index_GRCm38.PRI.p5_$LENGTH"
      BIGWIGGENOME="GRCm38"
      CHROMSIZES="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes/GRCm38.PRI_p5.chrom.sizes"
fi
#check if index is available
if [ -z "$INDEX" ]; then
        echo "Index is missing!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 7
fi
# Check whether one or two reads are given, whether files are available, non-identical and have the right extensions
if [ ! -f "$FASTQR1" ]; then
    echo "Fastq file (read 1) not found!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 3
fi
case "$FASTQR1" in
    *.fastq.gz) ZIP="--readFilesCommand gunzip -c "
    ;;
    *.fastq)    ZIP=""
    ;;
    *.fq.gz)    ZIP="--readFilesCommand gunzip -c "
    ;;
    *.fq)       ZIP=""
    ;;
    *)          echo "Fastq file (read 1) has wrong extension (should be fastq, fq, fastq.gz or fq.gz!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 4
    ;;
esac
if [ -z "$FASTQR2" ] || [ "$FASTQR2" == "NA" ] || [ "$FASTQR2" == "na" ] ; then
    SEQTYPE=1
    TYPE="SE"
    SKEWTYPE="any"
    echo -e \\n"Data processing for £{BOLD}single-end£{NORM} reads"
elif [ ! -f "$FASTQR2" ]; then
    echo "Fastq file (read 2) not found!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 5
elif [ "$FASTQR1" == "$FASTQR2" ]; then
    echo "Fastq files for read 1 and 2 are identical!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 6
else
        SEQTYPE=2
        TYPE="PE"
        SKEWTYPE="pe"
        echo -e \\n"Data Processing for £{BOLD}paired-end£{NORM} reads"
        case "$FASTQR2" in
       *.fastq.gz) ZIP="--readFilesCommand gunzip -c "
       ;;
       *.fastq)    ZIP=""
       ;;
       *.fq.gz)    ZIP="--readFilesCommand gunzip -c "
       ;;
       *.fq)       ZIP=""
       ;;
       *)          echo "Fastq file (read 2) has wrong extension (should be fastq, fq, fastq.gz or fq.gz!"
                          echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
                   exit 7
       ;;
    esac
fi
# Check whether the subfolders are already there - if not ask whether path should be made
if [ -z "$DIRECTORY" ]; then
        echo "Subdirectory is missing!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 7
fi
# rhskl17 version
MAPPINGDIR="/misc/data/processedData/mapping/RNA/£{BIGWIGGENOME}/RNAseq/£{DIRECTORY}"
BIGWIGDIR="/misc/data/processedData/bigWig/RNA/£{BIGWIGGENOME}/RNAseq/£{DIRECTORY}"
if [ ! -d "$MAPPINGDIR" ]; then                        #looks for the mapping output folder - if not available, askes whether to make
↪  it
        echo "$MAPPINGDIR"
        read -p "Do you want to create this mapping folder (y/n)?" choice
        case "$choice" in
                  y|Y )         mkdir -p "$MAPPINGDIR"
                ;;
                  n|N )   echo "Didn't create the folder and stopped!"
                    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
                    exit 8
                ;;
                  * ) echo "invalid response";;
        esac
fi
if [ ! -d "$BIGWIGDIR" ]; then                         #looks for the bigWig output folder - if not available, askes whether to make
↪  it
        echo "$BIGWIGDIR"
        read -p "Do you want to create this bigWig folder (y/n)?" choice
        case "$choice" in
                  y|Y )         mkdir -p "$BIGWIGDIR"
                ;;
                  n|N )   echo "Didn't create the folder and stopped!"
                    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
                    exit 8
                ;;
                  * ) echo "invalid response";;
        esac
fi
# Check whether sample name is given
if [ -z "$SAMPLENAME" ]; then
        echo "Sample name is missing!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 9
fi
SAMPLENAME="£{SAMPLENAME// /_}"  # replaces spaces with underscores in sample name, just in case...
# Creating the folder for STAR
if [ ! -d "£{MAPPINGDIR}/£{SAMPLENAME}" ]; then                         #looks for the mapping output folder - if not available,
↪  askes whether to make it
   mkdir "£{MAPPINGDIR}/£{SAMPLENAME}"
fi
if [ ! -d "£{MAPPINGDIR}/logs" ]; then                        #looks for the logs output folder - if not available, makes it
   mkdir "£{MAPPINGDIR}/logs"
fi
# If everything passed, create a log file for all the commands that are executed during the processing
LOGFILE="£{MAPPINGDIR}/logs/£{SAMPLENAME}.commands.txt"
touch $LOGFILE
# Function to append command lines into logfile and then execute
exe() { echo -e "\£ $@"\\n >> $LOGFILE; "$@" ; }
# Generating a random number file name for temporary files
TEMPFILES="/loctmp/$RANDOM.tmp"
# Optionally starting the pipeline by trimming the reads with homerTools
if [ $TRIM == 1 ] ; then
   ZIP=""
   case "$SEQTYPE" in
   1 )  exe homerTools trim -len $TRIMLENGTH $FASTQR1
        exe mv "$FASTQR1.trimmed" "$TEMPFILES-trimmed.fastq"
        FASTQR1="$TEMPFILES-trimmed.fastq"
 #       mv "£FASTQR1.lengths" "£{MAPPINGDIR}/logs/£{SAMPLENAME}.trimming.log"
        ;;
   2 )  exe homerTools trim -len $TRIMLENGTH $FASTQR1
        exe mv "$FASTQR1.trimmed" "$TEMPFILES-trimmed-pair1.fastq"
        FASTQR1="$TEMPFILES-trimmed-pair1.fastq"
```

```
#         mv "£FASTQR1.lengths" "£{MAPPINGDIR}/logs/£{SAMPLENAME}.R1.trimming.log"
        exe homerTools trim -len $TRIMLENGTH $FASTQR2
        exe mv "$FASTQR2.trimmed" "$TEMPFILES-trimmed-pair2.fastq"
        FASTQR2="$TEMPFILES-trimmed-pair2.fastq"
#         mv "£FASTQR2.lengths" "£{MAPPINGDIR}/logs/£{SAMPLENAME}.R2.trimming.log"
        ;;
   esac
fi
# Running STAR
echo -e \\n"Running STAR on $SAMPLENAME."\\n
cd "£{MAPPINGDIR}/£{SAMPLENAME}"
case "$SEQTYPE" in
        1 )  exe $STAR --runThreadN $THREADS \
        --genomeDir $INDEX \
        --readFilesIn $FASTQR1 \
        $ZIP\
        --outFilterIntronMotifs RemoveNoncanonicalUnannotated \
        --outReadsUnmapped Fastq \
        --alignSJoverhangMin 8 \
        --alignSJDBoverhangMin 1 \
        --alignMatesGapMax 1000000 \
        --alignIntronMax 1000000 \
        --outSAMtype BAM SortedByCoordinate \
        --quantMode GeneCounts \
        --outWigType bedGraph \
        --outWigStrand Stranded
        ;;
        2 )  exe $STAR --runThreadN $THREADS \
        --genomeDir $INDEX \
        --readFilesIn $FASTQR1 $FASTQR2 \
        $ZIP\
        --outFilterIntronMotifs RemoveNoncanonicalUnannotated \
        --outReadsUnmapped Fastq \
        --alignSJoverhangMin 8 \
        --alignSJDBoverhangMin 1 \
        --alignMatesGapMax 1000000 \
        --alignIntronMax 1000000 \
        --outSAMtype BAM SortedByCoordinate \
        --quantMode GeneCounts \
        --outWigType bedGraph \
        --outWigStrand Stranded
        ;;
esac
# Sorting bedgraph files
echo -e \\n"Sorting the bedGraph files for $SAMPLENAME."\\n
LC_COLLATE=C sort -k1,1 -k2,2n $MAPPINGDIR/$SAMPLENAME/Signal.Unique.str1.out.bg -o $TEMPFILES.str1.sorted.bg
LC_COLLATE=C sort -k1,1 -k2,2n $MAPPINGDIR/$SAMPLENAME/Signal.Unique.str2.out.bg -o $TEMPFILES.str2.sorted.bg
echo -e "\£ LC_COLLATE=C sort -k1,1 -k2,2n $MAPPINGDIR/$SAMPLENAME/Signal.Unique.str1.out.bg -o $TEMPFILES.str1.sorted.bg"\\n >>
↪  $LOGFILE
echo -e "\£ LC_COLLATE=C sort -k1,1 -k2,2n $MAPPINGDIR/$SAMPLENAME/Signal.Unique.str2.out.bg -o $TEMPFILES.str2.sorted.bg"\\n >>
↪  $LOGFILE
# inverting values for the second strand
exe awk 'BEGIN{FS=OFS="\t"}{print £1,£2,£3,£4*(-1)}' $TEMPFILES.str2.sorted.bg > $TEMPFILES.str2.inv.bg
# generating bigWigs for both strands
echo -e \\n"Generating bigWig files for $SAMPLENAME."\\n
exe bedGraphToBigWig $TEMPFILES.str1.sorted.bg $CHROMSIZES $BIGWIGDIR/$SAMPLENAME.Unique.for.bigwig
exe bedGraphToBigWig $TEMPFILES.str2.inv.bg $CHROMSIZES $BIGWIGDIR/$SAMPLENAME.Unique.rev.bigwig
# generating corresponding tdf files for IGV browser
echo -e "Generating TDF files for $SAMPLENAME."\\n
exe bigWigToWig $BIGWIGDIR/$SAMPLENAME.Unique.for.bigwig $TEMPFILES.str1.sorted.wig -udcDir=/loctmp
exe bigWigToWig $BIGWIGDIR/$SAMPLENAME.Unique.rev.bigwig $TEMPFILES.str2.inv.wig -udcDir=/loctmp
exe $IGVTOOLS toTDF $TEMPFILES.str1.sorted.wig $BIGWIGDIR/$SAMPLENAME.Unique.for.cov.tdf $CHROMSIZES
exe $IGVTOOLS toTDF $TEMPFILES.str2.inv.wig $BIGWIGDIR/$SAMPLENAME.Unique.rev.cov.tdf $CHROMSIZES
# move and rename gene count table and bam file
case "$TRIM" in
        1 )  exe mv £{MAPPINGDIR}/£{SAMPLENAME}/Aligned.sortedByCoord.out.bam £{MAPPINGDIR}/£{SAMPLENAME}.trimmed.sorted.bam
         exe mv £{MAPPINGDIR}/£{SAMPLENAME}/ReadsPerGene.out.tab £{MAPPINGDIR}/£{SAMPLENAME}.trimmed.ReadsPerGene.txt
         exe mv £{MAPPINGDIR}/£{SAMPLENAME}/Log.final.out £{MAPPINGDIR}/logs/£{SAMPLENAME}.trimmed.summary.log.txt
         exe mv £{MAPPINGDIR}/£{SAMPLENAME}/Log.out £{MAPPINGDIR}/logs/£{SAMPLENAME}.trimmed.processing.log.txt
         ;;
    0 )  exe mv £{MAPPINGDIR}/£{SAMPLENAME}/Aligned.sortedByCoord.out.bam £{MAPPINGDIR}/£{SAMPLENAME}.sorted.bam
         exe mv £{MAPPINGDIR}/£{SAMPLENAME}/ReadsPerGene.out.tab £{MAPPINGDIR}/£{SAMPLENAME}.ReadsPerGene.txt
         exe mv £{MAPPINGDIR}/£{SAMPLENAME}/Log.final.out £{MAPPINGDIR}/logs/£{SAMPLENAME}.summary.log.txt
         exe mv £{MAPPINGDIR}/£{SAMPLENAME}/Log.out £{MAPPINGDIR}/logs/£{SAMPLENAME}.processing.log.txt
         ;;
esac
# removing tmp files
exe rm -r "£{MAPPINGDIR}/£{SAMPLENAME}"
exe rm £{TEMPFILES}.*
```

mapChIP.sh:

```
#!/bin/bash
# by Michael Rehli, Febr 23, 2018, mod. March 5, 2018, mod. March 11, 2018, mod. March 26, 2016
# bash script to map ChIPseq-like data (SE- only), generate tagDirectories, coverage bigWigs and TDF files
# optional for input data: copy number alterations
#
#setting homer environment
DIR_PKG="/misc/software/ngs"
PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin
export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH
# Defining the program versions (needs to be adjusted to the server/workstation used)
# rhskl17 version
BOWTIE="/misc/software/ngs/bowtie/bowtie2-2.3.4-linux-x86_64/bowtie2"
SAMTOOLS="/misc/software/ngs/samtools/samtools-1.6/bin/samtools"
SAMBAMBA="/misc/software/ngs/sambamba/v0.6.7/sambamba"
BEDTOOLS="/misc/software/ngs/bedtools/bedtools2-2.27.1/bin/bedtools"
PICARD="/usr/bin/java -jar /misc/software/ngs/picard/src/v2.17.3/picard.jar"
IGVTOOLS="/usr/bin/java -Xmx4g -Djava.awt.headless=true -jar /misc/software/viewer/IGV/IGVTools_2.3.98/igvtools.jar"
# required input:
#
# -f <fastq or fastq.gz>
# -g <genome>  available genomes include hg19, GRCh38, mm9, mm10
# -o <output subdirectory> only subdirectories above ChIP- or Input-folder are given
# -n <sample name> used for sam, bigwig and tag directory
#
# optional:
# -i indicates that the sample is input
# -m indicates that the sample is MCIp
# -t <number of threads>
# options specific for input samples:
# -c <read length for mappability> generates copy number profiles and annotates CNVs with Cancer Genes from Cosmic Database (Available
↪  mappability data for 50, 75, and 100 bp)
# -s <XX/XY> gender XX female, XY male (default) - only necessary if -c is given
# -a <tissue> E (epithelial), L (leukaemia/lymphoma), M (mesenchymal), O (other) will output Cancer Genes only for given tissues (only
↪  human!)
# Set Script Name variable
SCRIPT=`basename £{BASH_SOURCE[0]}`
# Set fonts for Help.
NORM=`tput sgr0`
BOLD=`tput bold`
REV=`tput smso`
# USAGE function
function USAGE {
  echo -e \\n"Help documentation for £{BOLD}£{SCRIPT}.£{NORM}"\\n
  echo -e "£{REV}Basic usage:£{NORM}\n£{BOLD}$SCRIPT -g <genome> -f <fastq> -o <subdirectory> -n <name> -t <threads>£{NORM} "\\n
  # -c <read length> -s <gender>"\\n
  echo "Required:"
  echo "£{BOLD}-g£{NORM} <genome>  available genomes include hg19, GRCh38, mm9, mm10"
  echo "£{BOLD}-f£{NORM} <fastq or fastq.gz> forward read"
  echo "£{BOLD}-d£{NORM} <output subdirectory> only subdirectories above ChIP- or Input-folder are given"
  echo "£{BOLD}-n£{NORM} <sample name> used for bam, bigwig and tag directory, etc."
  echo -e \\n"Optional:"
  echo "£{BOLD}-t£{NORM} <number of threads> (optional/default 16)"
  echo "£{BOLD}-i£{NORM} treat sample as genomic input (stored in DNA directory)"
  echo "£{BOLD}-m£{NORM} treat sample as MCIp (stored in DNA directory, TagDir with -keepOne)"
  echo "£{BOLD}-c£{NORM} <read length> will use FreeC to generate CNV profiles of Input files. Read length refers to the mappability
  ↪  files available for a given genome,"
  echo "   which are found in /misc/software/ngs/genome/sequence/XXXX (XXXX = genome)"
  echo "£{BOLD}-s£{NORM} <gender> accepts either XX or XY (default) - only necessary if -c is given"
  echo "£{BOLD}-a£{NORM} <tissue> E (epithelial), L (leukaemia/lymphoma), M (mesenchymal), O (other) will output COSMIC Cancer Genes
  ↪  for given tissues £{BOLD}HUMAN ONLY!£{NORM}"
  echo -e \\n"Example: £{BOLD}$SCRIPT -g hg19 -f TFread.fastq.gz -d MOMACDC -n TF -t 32 £{NORM}"\\n
  exit 1
}
#Check the number of arguments. If none are passed, print help and exit.
NUMARGS=$#
if [ $NUMARGS -eq 0 ]; then
  USAGE
fi
# Defining available genomes
GENOMES=(hg19 GRCh38 hg38 mm9 mm10)
TISSUETYPES=(E L M O)
KEEP=""
# Set defaults
GENDER="XY"
REF="ref"
THREADS=16
```

```
RUNFREEC=0
ANNOTATE=0
INPUT=0
MCIP=0
FOLDER="ChIP"
TOPFOLDER="chromatin"
# Parse command line options
while getopts :g:f:d:n:t:c:s:a:imh OPTIONS; do
  case $OPTIONS in
    g)  #set option "g" - Bowtie will will only accept GRCh38 while HOMER needs hg38
      GENOME=$OPTARG
      HOMERGENOME=$OPTARG
      ;;
    f)  #set option "f"
      FASTQR1=$OPTARG
      ;;
    d)  #set option "d"
      DIRECTORY=$OPTARG
      ;;
    n)  #set option "n"
      SAMPLENAME=$OPTARG
      ;;
    t)  #set option "t"
      THREADS=$OPTARG
      SAMTHREADS=$((THREADS-1))
      ;;
    c)  #set option "c"
      MAPPAB=$OPTARG
      RUNFREEC=1
      ;;
    s)  #set option "s"
      GENDER=$OPTARG
      ;;
    h)  #show help
      USAGE
      ;;
    i)  #set option "i"
      echo -e \\n"Sample is genomic input."\\n
      INPUT=1
      FOLDER="Input"
      TOPFOLDER="DNA"
      ;;
    m)  #set option "m"
      echo -e \\n"Sample is MCIp."\\n
      MCIP=1
      FOLDER="MCIp"
      TOPFOLDER="DNA"
      KEEP="-keepOne "
      ;;
    a)  #set option "a"
          ANNOTATE=1
          TISSUE=$OPTARG
      ;;
    \?) #unrecognized option - show USAGE
      echo -e \\n"Option -£{BOLD}$OPTARG£{NORM} not allowed."
      USAGE
      ;;
  esac
done
shift $((OPTIND-1))
# Sanity checks
# excude using -m and -i together
if [ $INPUT == 1 ] && [ $MCIP == 1 ]; then
      echo -e \\n"£{BOLD}Use either -i or -m.{NORM}"
      echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
      exit 2
fi
# Check whether Tissues are selected and correct
if [ $ANNOTATE == 1 ] && [ $INPUT == 1 ]; then
        INTISSUE=$(echo £{TISSUETYPES[@]} | grep -o "$TISSUE" | wc -w)
        if [ $INTISSUE == 0 ] ; then
      echo -e \\n"Tissue -£{BOLD}$TISSUE£{NORM} not available (Options include E, M, L & O)."
      echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
      exit 2
        fi
fi
# Check whether a reasonable number of cores is given
if [ $THREADS -lt 1 ] || [ "$THREADS" -gt 63 ]; then
        echo "Number of threads should be in the range of 1 - 63!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 9
fi
```

```
# Check whether GENOME is available
INARRAY=$(echo £{GENOMES[@]} | grep -o "$GENOME" | wc -w)
if [ $INARRAY == 0 ] ; then
      echo -e \\n"Genome -£{BOLD}$GENOME£{NORM} not available."
      echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
      exit 2
fi
GENOMEPATH=$GENOME
HOMERGENOME=$GENOME
BIGWIGGENOME=$GENOME
# path to chromosome sizes
CHROMSIZES="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes/$GENOME.chrom.sizes"
# path to GEM files (mappability info), and chromosome sizes for FreeC
# needs two different one's due to a bug in FreeC
GEMPATH="/misc/software/ngs/genome/sequence/£{GENOME}/mappability"
GEM="£{GEMPATH}/£{GENOME}_£{MAPPAB}.mappability"
case $GENDER in
        XY)   REF="ref"
        ;;
        XX)          REF="ref-Y"
        ;;
esac
REDCHROMSIZES="/misc/software/ngs/freec/FREEC-11.0/data/£{GENOME}.£{REF}.chrom.sizes"
if [ $GENOME == "hg38" ] || [ $GENOME == "GRCh38" ] ; then
      GENOME="hg38"
      GENOMEPATH="GRCh38.PRI_p10"
      BIGWIGGENOME="GRCh38"
      HOMERGENOME="hg38"
      CHROMSIZES="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes/GRCh38.PRI_p10.chrom.sizes"
      GEMPATH="/misc/software/ngs/genome/sequence/GRCh38.PRI_p10/mappability"
      GEM="£{GEMPATH}/GRCh38_£{MAPPAB}.mappability"
      REDCHROMSIZES="/misc/software/ngs/freec/FREEC-11.0/data/£{GENOME}.£{REF}.chrom.sizes"
fi
#bowtie index
INDEX="$GENOMEPATH/$GENOME"
# if -c is given, check whether mappability track is available
if [ "$RUNFREEC" == 1 ] && [ ! -f "$GEM" ]; then
    echo "Gem file for £{GENOME} not found!"
    echo -e "Available gem files for £{GENOME} are:"
    for entry in $GEMPATH/*
           do
       echo $entry
    done
    echo -e \\n"Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 3
fi
# Check whether fastq file is available and has the right extension
if [ ! -f "$FASTQR1" ]; then
    echo "Fastq file (read 1) not found!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 3
fi
case "$FASTQR1" in
    *.fastq.gz) ;;
    *.fastq)    ;;
    *.fq.gz)    ;;
    *.fq)       ;;
    *)          echo "Fastq file (read 1) has wrong extension (should be fastq, fq, fastq.gz or fq.gz!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 4
    ;;
esac
# Check whether the subfolders are given and already there - if not ask whether path should be made
if [ -z "$DIRECTORY" ]; then
        echo "Subdirectory is missing!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 7
fi
# rhskl17 version
MAPPINGDIR="/misc/data/processedData/mapping/£{TOPFOLDER}/£{BIGWIGGENOME}/£{FOLDER}/£{DIRECTORY}"
TAGDIRDIR="/misc/data/processedData/tagDir/£{TOPFOLDER}/£{BIGWIGGENOME}/£{FOLDER}/£{DIRECTORY}"
BIGWIGDIR="/misc/data/processedData/bigWig/£{TOPFOLDER}/£{BIGWIGGENOME}/£{FOLDER}/£{DIRECTORY}"
if [ ! -d "$MAPPINGDIR" ]; then                         #looks for the mapping output folder - if not available, askes whether to make
↪  it
        echo "$MAPPINGDIR"
        read -p "Do you want to create this mapping folder (y/n)?" choice
        case "$choice" in
                  y|Y )         mkdir -p "$MAPPINGDIR"
                ;;
                  n|N )   echo "Didn't create the folder and stopped!"
                    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
                    exit 8
                ;;
                  * ) echo "invalid response";;
        esac
fi
if [ ! -d "$TAGDIRDIR" ]; then                         #looks for the tagDir output folder - if not available, askes whether to make
↪  it
        echo "$TAGDIRDIR"
        read -p "Do you want to create this tagDirectory folder (y/n)?" choice
        case "$choice" in
                  y|Y )         mkdir -p "$TAGDIRDIR"
                ;;
                    n|N )   echo "Didn't create the folder and stopped!"
                    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
                    exit 8
                ;;
                  * ) echo "invalid response";;
        esac
fi
if [ ! -d "$BIGWIGDIR" ]; then                         #looks for the bigWig output folder - if not available, askes whether to make
↪  it
        echo "$BIGWIGDIR"
        read -p "Do you want to create this bigWig folder (y/n)?" choice
        case "$choice" in
                  y|Y )         mkdir -p "$BIGWIGDIR"
                ;;
                  n|N )   echo "Didn't create the folder and stopped!"
                    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
                    exit 8
                ;;
                  * ) echo "invalid response";;
        esac
fi
# Check whether sample name is given
if [ -z "$SAMPLENAME" ]; then
        echo "Sample name is missing!"
    echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
    exit 9
fi
SAMPLENAME="£{SAMPLENAME// /_}"  # replaces spaces with underscores in sample name, just in case...
if [ ! -d "£{MAPPINGDIR}/logs" ]; then                         #looks for the logs output folder - if not available, makes it
   mkdir "£{MAPPINGDIR}/logs"
fi
# If everything passed, create a log file for all the commands that are executed during the processing
LOGFILE="£{MAPPINGDIR}/logs/£{SAMPLENAME}.commands.txt"
touch $LOGFILE
# Function to append command lines into logfile and then execute
exe() { echo -e "\£ $@"\\n >> $LOGFILE; "$@" ; }
# Generating a random number file name for temporary files
TEMPFILES="/loctmp/$RANDOM.tmp"
# Running Bowtie2
echo -e \\n"Running Bowtie2 on $SAMPLENAME."\\n
exe $BOWTIE --very-sensitive -p $THREADS -x $INDEX --met-file $MAPPINGDIR/logs/$SAMPLENAME.aln_metrics.txt -U $FASTQR1 2>
↪  $MAPPINGDIR/logs/$SAMPLENAME.aln_rates.txt -S $TEMPFILES.sam
###################################################
#modified on March 26, 2018
#MCIp should not be filtered for mapping quality
case "$MCIP" in
        0 )         # Sorting and reducing the SAM-file to q>10
                echo -e "Sorting and reducing $SAMPLENAME to q>10."\\n
                exe $SAMTOOLS view -h -S -q 10 $TEMPFILES.sam -@ $SAMTHREADS -o $TEMPFILES.red.sam
                exe $SAMTOOLS sort -O sam -@ $SAMTHREADS -T $TEMPFILES -o $TEMPFILES.sorted.sam $TEMPFILES.red.sam
        ;;
        1 )  # Sorting
                echo -e "Sorting $SAMPLENAME."\\n
                exe $SAMTOOLS sort -O sam -@ $SAMTHREADS -T $TEMPFILES -o $TEMPFILES.sorted.sam $TEMPFILES.sam
        ;;
esac
###################################################
echo -e "Generating TagDir, BigWig, and TDF and finding Peaks."\\n
# Generating a HOMER tag directory
exe makeTagDirectory $TAGDIRDIR/$SAMPLENAME $TEMPFILES.sorted.sam $KEEP -genome $HOMERGENOME -checkGC
# Making a UCSC browser track
exe makeUCSCfile $TAGDIRDIR/$SAMPLENAME -bigWig $CHROMSIZES -o $BIGWIGDIR/$SAMPLENAME.bigwig
# Making an IGV browser track
exe $IGVTOOLS count $TEMPFILES.sorted.sam $BIGWIGDIR/$SAMPLENAME.cov.tdf $CHROMSIZES
# Run FreeC if requested
if [ $RUNFREEC == 1 ]; then
    if [ ! -d "£{MAPPINGDIR}/CNVdata" ]; then                         #looks for the CNVdata output folder - if not available,
    ↪  makes it
       mkdir "£{MAPPINGDIR}/CNVdata"
    fi
    if [ ! -d "£{MAPPINGDIR}/CNVprofiles" ]; then                        #looks for the CNVprofile output folder - if not available,
   ↪  makes it
       mkdir "£{MAPPINGDIR}/CNVprofiles"
    fi
    exe mv $TEMPFILES.sorted.sam "£{MAPPINGDIR}/£{SAMPLENAME}.sam"
    case "$GENOME" in
                  hg19 )         COSMIC="/misc/software/ngs/freec/FREEC-11.0/data/COSMIC_Cancer_Gene_Census_GRCh37v84_2018.tsv"
                ;;
                  hg38 )  COSMIC="/misc/software/ngs/freec/FREEC-11.0/data/COSMIC_Cancer_Gene_Census_GRCh38v84_2018.tsv"
                ;;
        esac
    exe runFreeC.pl £{MAPPINGDIR}/CNVdata £{GENDER} £{GENOME} £{MAPPINGDIR} £{SAMPLENAME} £{REDCHROMSIZES} £{GEM}
        # annotation of CNVs with COSMIC genes
        exe FreeC2bed.pl £{MAPPINGDIR}/CNVdata/£{SAMPLENAME}.sam_CNVs £{TEMPFILES}.CNV.bed
        exe mv £{MAPPINGDIR}/CNVdata/£{SAMPLENAME}.pdf £{MAPPINGDIR}/CNVprofiles
        if [ $GENOME == "hg38" ] || [ $GENOME == "GRCh38" ] || [ $GENOME == "hg19" ]; then
                case "$ANNOTATE" in
                        0 )  echo -e "\£ ConvertCOSMIC.pl £{COSMIC} >£{TEMPFILES}.cosmic.bed"\\n  >> $LOGFILE
                                 ConvertCOSMIC.pl £{COSMIC} > £{TEMPFILES}.cosmic.bed
                                 echo -e "\£ £{BEDTOOLS} intersect -a £{TEMPFILES}.CNV.bed -b £{TEMPFILES}.cosmic.bed -wa -wb
                                ↪  >£{TEMPFILES}.inter.bed"\\n >> $LOGFILE
                                 £{BEDTOOLS} intersect -a £{TEMPFILES}.CNV.bed -b £{TEMPFILES}.cosmic.bed -wa -wb
                                ↪  >£{TEMPFILES}.inter.bed
                                 exe reformatCOSMICannotation.pl £{TEMPFILES}.inter.bed
                                ↪  £{MAPPINGDIR}/CNVdata/£{SAMPLENAME}.annotated_CNVs.txt
                        ;;
                        1 )  echo -e "\£ ConvertCOSMIC.pl £{COSMIC} -tissues £{TISSUE} >£{TEMPFILES}.cosmic.bed"\\n >> $LOGFILE
                                 ConvertCOSMIC.pl £{COSMIC} -tissues £{TISSUE} > £{TEMPFILES}.cosmic.bed
                                 echo -e "\£ £{BEDTOOLS} intersect -a £{TEMPFILES}.CNV.bed -b £{TEMPFILES}.cosmic.bed -wa -wb
                                ↪  >£{TEMPFILES}.inter.bed"\\n >> $LOGFILE
                                 £{BEDTOOLS} intersect -a £{TEMPFILES}.CNV.bed -b £{TEMPFILES}.cosmic.bed -wa -wb
                                ↪  >£{TEMPFILES}.inter.bed
                                 exe reformatCOSMICannotation.pl £{TEMPFILES}.inter.bed
                                ↪  £{MAPPINGDIR}/CNVdata/£{SAMPLENAME}.£{TISSUE}.annotated_CNVs.txt
                        ;;
                esac
        fi
    #convert sam to bam
    exe $SAMTOOLS view -bS -@ $SAMTHREADS -o "£{MAPPINGDIR}/£{SAMPLENAME}.bam" "£{MAPPINGDIR}/£{SAMPLENAME}.sam"
    exe rm "£{MAPPINGDIR}/£{SAMPLENAME}.sam"
else
   # just convert sam to bam
   exe $SAMTOOLS view -bS -@ $SAMTHREADS -o "£{MAPPINGDIR}/£{SAMPLENAME}.bam" $TEMPFILES.sorted.sam
fi
# removing tmp files
rm £{TEMPFILES}.*
```

filter4Mappability.sh:

```
#!/bin/bash
# by Michael Rehli, March 17, 2018
# bash script to filter-out regions with low mappability
#
#setting homer environment
DIR_PKG="/misc/software/ngs"
PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin
export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH
# Defining the program versions (needs to be adjusted to the server/workstation used)
# rhskl17 version
BOWTIE="/misc/software/ngs/bowtie/bowtie2-2.3.4-linux-x86_64/bowtie2"
SAMTOOLS="/misc/software/ngs/samtools/samtools-1.6/bin/samtools"
SAMBAMBA="/misc/software/ngs/sambamba/v0.6.7/sambamba"
BEDTOOLS="/misc/software/ngs/bedtools/bedtools2-2.27.1/bin/bedtools"
PICARD="/usr/bin/java -jar /misc/software/ngs/picard/src/v2.17.3/picard.jar"
IGVTOOLS="/usr/bin/java -Xmx4g -Djava.awt.headless=true -jar /misc/software/viewer/IGV/IGVTools_2.3.98/igvtools.jar"
# required input:
#
# -p <peak file> accepts position or bed files
# -g <genome> available genomes include hg19, GRCh38, mm9, mm10, default GRCh38
# -s <mapping size> currently accepts 50, 75, 100 (available mappability traacks)
# -f <cut-off> filter for mappability score of regions [range:0-1], default 0.8
#
# Set Script Name variable
SCRIPT=`basename £{BASH_SOURCE[0]}`
```

```
# Set fonts for Help.
NORM=`tput sgr0`
BOLD=`tput bold`
REV=`tput smso`
# USAGE function
function USAGE {
  echo -e \\n"Help documentation for £{BOLD}£{SCRIPT}.£{NORM}"\\n
  echo -e "£{REV}Basic usage:£{NORM}\n£{BOLD}$SCRIPT -p <peak file> -g <genome> -f <cut-off> -s <mapping size>"\\n
  echo "Required:"
  echo "£{BOLD}-p£{NORM} <peak/region file> accepts position or bed files"
  echo "£{BOLD}-s£{NORM} <mapping size> currently accepts 50, 75, 100 (available mappability tracks), default:50"
  echo -e \\n"Optional:"
  echo "£{BOLD}-g£{NORM} <genome>  available genomes include hg19, GRCh38, mm9, mm10, GRCm38 default:GRCh38"
  echo "£{BOLD}-f£{NORM} <cut-off> filter for mappability score of regions [range:0-1], default:0.8"
  echo -e \\n"Example: £{BOLD}$SCRIPT -p peaks.txt -g hg19 -f 0.9 -s 50 £{NORM}"\\n
  exit 1
}
#Check the number of arguments. If none are passed, print help and exit.
NUMARGS=$#
if [ $NUMARGS -eq 0 ]; then
  USAGE
fi
# Defining available data sets
GENOMES=(hg19 GRCh38 hg38 mm9 mm10 GRCm38)
MAPPABILITY=(50 75 100)
# Set defaults
GENOME="GRCh38"
CUTOFF=0.8
SIZE=50
# set path
MAPDIR="/misc/data/analysis/generalStuff/mappability"
SIZES_DIR="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes"
TEMPFILES="/loctmp/$RANDOM.tmp"
# Parse command line options
while getopts :g:f:p:s:h OPTIONS; do
  case $OPTIONS in
    p)  #set option "p"
      PEAKS=$OPTARG
      ;;
    g)  #set option "g" - Bowtie will will only accept GRCh38 while HOMER needs hg38
      GENOME=$OPTARG
      ;;
    f)  #set option "f"
      CUTOFF=$OPTARG
      ;;
    s)  #set option "s"
      SIZE=$OPTARG
      ;;
    h)  #show help
      USAGE
      ;;
    \?) #unrecognized option - show USAGE
      echo -e \\n"Option -£{BOLD}$OPTARG£{NORM} not allowed."
      USAGE
      ;;
  esac
done
shift $((OPTIND-1))
# Sanity checks
# Check whether GENOME is available
INARRAY=$(echo £{GENOMES[@]} | grep -o "$GENOME" | wc -w)
if [ $INARRAY == 0 ] ; then
      echo -e \\n"Genome -£{BOLD}$GENOME£{NORM} not available."
      echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
      exit 2
fi
INMAP=$(echo £{MAPPABILITY[@]} | grep -o "$SIZE" | wc -w)
if [ $INMAP == 0 ] ; then
      echo -e \\n"Mappability for size £{BOLD}$SIZE£{NORM} not available."
      echo -e "Use £{BOLD}$SCRIPT -h£{NORM} to see the help documentation."\\n
      exit 2
fi
# Define mappability track and chrom.sizes and genome version for available genomes
case $GENOME in
        hg19)   MAPTRACK=£{MAPDIR}/hg19.mappability_£{SIZE}.bedGraph
                        HOMERGENOME=hg19
        ;;
        mm9)   MAPTRACK=£{MAPDIR}/mm9.mappability_£{SIZE}.bedGraph
                        HOMERGENOME=mm9
        ;;
        hg38|GRCh38)   MAPTRACK=£{MAPDIR}/hg38.mappability_£{SIZE}.bedGraph
                        HOMERGENOME=hg38
        ;;
        mm10|GRCm38)   MAPTRACK=£{MAPDIR}/mm10.mappability_£{SIZE}.bedGraph
                        HOMERGENOME=mm10
        ;;
esac
# Run annotatePeaks.pl
annotatePeaks.pl £{PEAKS} £{HOMERGENOME} -size given -bedGraph £{MAPTRACK} -nogene -noann -cpu 3 > £{TEMPFILES}.ann.peaks.txt
# Filter out mappability below cut-off
FILTEREDPEAKS=£{PEAKS%.*}.mapScoreFiltered.txt
myFilterFile.pl £{TEMPFILES}.ann.peaks.txt -column 8 -lowerlimit £{CUTOFF} > £{FILTEREDPEAKS}
LENGTHA="$(wc -l <"$PEAKS")"
LENGTHB="$(wc -l <"$FILTEREDPEAKS")"
DIFFERENCE=$((LENGTHA-LENGTHB))
echo -e \\n"Done. Filtered out £{DIFFERENCE} regions with mappability scores below £{CUTOFF}, leaving £{LENGTHB} regions."\\n
# removing tmp files
rm £{TEMPFILES}.*
```

This script was used for the FACS analysis in R:

```
#!/usr/bin/env bash
## Script to Analyse FACS Data derived from nine colour panel and 16 macrophage conditions ##

## Set Directories ##
DATAexp93="/Users/claudiakiesewetter/Documents/Doktorarbeit/FACS/FCS_Files/exp93"
DATAexp95="/Users/claudiakiesewetter/Documents/Doktorarbeit/FACS/FCS_Files/exp95"
DATAexp98="/Users/claudiakiesewetter/Documents/Doktorarbeit/FACS/FCS_Files/exp98"

OUTDIR="/Users/claudiakiesewetter/Documents/Doktorarbeit/FACS/Analysis"
mkdir -p $OUTDIR/FSCSSC
mkdir -p $OUTDIR/LiveDead
mkdir -p $OUTDIR/singlet
FSCDIR="$OUTDIR/FSCSSC"
LIVEDIR="$OUTDIR/LiveDead"
SINGLEDIR="$OUTDIR/singlet"

### Setting colors ###

colCol1="deeppink"
colCol4="violetred3"
colABTcPlastic="slateblue3"
colLam="orchid1"
colMono="forestgreen"
colAdipo="khaki2"
colSero="lightsalmon2"
colTGF="darkgoldenrod3"
colHermin="brown4"
colNaCl="gold2"
colIgG="orange1"
colFib="plum3"
colMCSF="cornflowerblue"
coliDC="dodgerblue3"
colGMCSF="cadetblue3"
colNONAD="hotpink"
colABTeflon="wheat3"

### Specifications ###

#Samples were dyed with 9 colours for surface receptors:
#PE: CD16
#PerCP: CD19, CD56, 7AAD --> Dump channel to exclude dead cells and lymphocytes
#PE-Cy7: HLA-DR
#APC: CD11b
#APC-H7: CD71
#BV421: CD115
#BV510: CD64
#BV650: CD163
#BV711: CD14

#######################################################
### This R script contains the basic analysis steps ###
#######################################################

cat > ~/Documents/Doktorarbeit/FACS.R <<EOF
library("flowCore")
library("flowViz")
library("ggcyto")
library("ggplot2")
library("RColorBrewer")
library("ggpubr")

########################################
### Load Data from FCS Files into R ###
########################################

### exp 93 M1 ###

setwd("£DATAexp93")

exp93_AB <- read.flowSet(c("exp93_AB_MCSF_unstained_016.fcs", "exp93_AB_MCSF_Panel_017.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp93_ColI <- read.flowSet(c("exp93_Collagen_I_unstained_034.fcs", "exp93_Collagen_I_Panel_035.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp93_ColIV <- read.flowSet(c("exp93_Collagen_IV_unstained_036.fcs", "exp93_Collagen_IV_Panel_037.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp93_Fib <- read.flowSet(c("exp93_Fibronektin_unstained_038.fcs", "exp93_Fibronektin_Panel_039.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp93_Lam <- read.flowSet(c("exp93_Laminin_unstained_040.fcs", "exp93_Laminin_Panel_041.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp93_Non <- read.flowSet(c("exp93_Rollerflaschen_unstained_042.fcs", "exp93_Rollerflaschen_Panel_043.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp93_iDC <- read.flowSet(c("exp93_iDC_unstained_022.fcs", "exp93_iDC_Panel_023.fcs"), alter.names = TRUE, transformation = FALSE)

exp93_GMCSF <- read.flowSet(c("exp93_GMCSF_unstained_020.fcs", "exp93_GMCSF_Panel_021.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp93_MCSF <- read.flowSet(c("exp93_MCSF_FCS_unstained_018.fcs", "exp93_MCSF_FCS_Tube_001_019.fcs"), alter.names = TRUE, transformation
↪  = FALSE)

exp93_ABT <- read.flowSet(c("exp93_AB_Serum_unstained_001.fcs", "exp93_AB_Serum_Dump_003.fcs", "exp93_AB_Serum_FMO_CD115_009.fcs",
↪  "exp93_AB_Serum_FMO_CD11b_007.fcs", "exp93_AB_Serum_FMO_CD15_004.fcs", "exp93_AB_Serum_FMO_CD64_010.fcs",
↪  "exp93_AB_Serum_FMO_CD16_005.fcs", "exp93_AB_Serum_FMO_CD71_008.fcs", "exp93_AB_Serum_FMO_CD163_011.fcs",
↪  "exp93_AB_Serum_FMO_CD14_012.fcs", "exp93_AB_Serum_FMO_HLADR_006.fcs", "exp93_AB_Serum_Panel_013.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp93_Adipo <- read.flowSet(c("exp93_Adiponektin_unstained_028.fcs", "exp93_Adiponektin_Panel_029.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp93_Sero <- read.flowSet(c("exp93_Serotonin_unstained_030.fcs", "exp93_Serotonin_Panel_031.fcs"), alter.names = TRUE, transformation
↪  = FALSE)

exp93_Hermin <- read.flowSet(c("exp93_Hermin_unstained_026.fcs", "exp93_Hermin_Panel_027.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp93_IgG <- read.flowSet(c("exp93_IgG_unstained_032.fcs", "exp93_IgG_Panel_033.fcs"), alter.names = TRUE, transformation = FALSE)

exp93_NaCl <- read.flowSet(c("exp93_NaCl_unstained_014.fcs", "exp93_NaCl_Panel_015.fcs"), alter.names = TRUE, transformation = FALSE)

exp93_TGF <- read.flowSet(c("exp93_TGF_unstained_024.fcs", "exp93_TGF_Panel_025.fcs"), alter.names = TRUE, transformation = FALSE)

### exp 95 M2 ###

setwd("£DATAexp95")
exp95_AB <- read.flowSet(c("exp95_AB_MCSF_unstained_016.fcs", "exp95_AB_MCSF_Panel_017.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp95_ColI <- read.flowSet(c("exp95_Collagen_I_unstained_037.fcs", "exp95_Collagen_I_Panel_038.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp95_ColIV <- read.flowSet(c("exp95_Collagen_IV_unstained_039.fcs", "exp95_Collagen_IV_Panel_040.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp95_Fib <- read.flowSet(c("exp95_Fibronektin_unstained_044.fcs", "exp95_Fibronektin_Panel_045.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp95_Lam <- read.flowSet(c("exp95_Laminin_unstained_046.fcs", "exp95_Laminin_Panel_047.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp95_Non <- read.flowSet(c("Rollerflaschen_unstained_048.fcs", "Rollerflaschen_Panel_049.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp95_iDC <- read.flowSet(c("iDC_unstained_022.fcs", "iDC_Panel_023.fcs"), alter.names = TRUE, transformation = FALSE)

exp95_GMCSF <- read.flowSet(c("GMCSF_unstained_020.fcs", "GMCSF_Panel_021.fcs"), alter.names = TRUE, transformation = FALSE)
```

```
exp95_MCSF <- read.flowSet(c("MCSF_FCS_unstained_018.fcs", "MCSF_FCS_Panel_019.fcs"), alter.names = TRUE, transformation = FALSE)

exp95_ABT <- read.flowSet(c("AB_Serum_unstained_001.fcs", "AB_Serum_Dump_003.fcs", "AB_Serum_FMO_CD115_009.fcs",
↪  "AB_Serum_FMO_CD11b_007.fcs", "AB_Serum_FMO_CD15_004.fcs", "AB_Serum_FMO_CD64_010.fcs", "exp95_AB_Serum_FMO_CD16_005.fcs",
↪  "exp95_AB_Serum_FMO_CD71_008.fcs", "exp95_AB_Serum_FMO_CD163_011.fcs", "exp95_AB_Serum_FMO_CD14_012.fcs",
↪  "AB_Serum_FMO_HLADR_006.fcs", "AB_Serum_Panel_013.fcs"), alter.names = TRUE, transformation = FALSE)

exp95_Adipo <- read.flowSet(c("Adiponektin_unstained_031.fcs", "Adiponektin_Panel_032.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp95_Sero <- read.flowSet(c("Serotonin_unstained_033.fcs", "Serotonin_Panel_034.fcs"), alter.names = TRUE, transformation = FALSE)

exp95_Hermin <- read.flowSet(c("Hermin_unstained_026.fcs", "Hermin_FMO CD15_028.fcs", "Hermin_FMO CD16_029.fcs", "Hermin_FMO
↪  CD64_030.fcs", "Hermin_Panel_027.fcs"), alter.names = TRUE, transformation = FALSE)

exp95_IgG <- read.flowSet(c("IgG_unstained_035.fcs", "IgG_Panel_036.fcs"), alter.names = TRUE, transformation = FALSE)

exp95_NaCl <- read.flowSet(c("NaCl_unstained_014.fcs", "NaCl_Panel_015.fcs"), alter.names = TRUE, transformation = FALSE)

exp95_TGF <- read.flowSet(c("TGF_unstained_024.fcs", "TGF_Panel_025.fcs"), alter.names = TRUE, transformation = FALSE)

### exp 98 M3 ###

setwd("£DATAexp98")
exp98_AB <- read.flowSet(c("exp98_AB_MCSF_unstained_016.fcs", "exp98_AB_MCSF_Panel_017.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp98_ColI <- read.flowSet(c("exp98_Collagen_I_unstained_038.fcs", "exp98_Collagen_I_Panel_039.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp98_ColIV <- read.flowSet(c("exp98_Collagen_IV_unstained_040.fcs", "exp98_Collagen_IV_Panel_041.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp98_Fib <- read.flowSet(c("exp98_Fibronektin_unstained_042.fcs", "exp98_Fibronektin_Panel_043.fcs"), alter.names = TRUE,
↪  transformation = FALSE)

exp98_Lam <- read.flowSet(c("exp98_Laminin_unstained_044.fcs", "exp98_Laminin_Panel_045.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp98_Non <- read.flowSet(c("Rollerflaschen_unstained_046.fcs", "Rollerflaschen_Panel_047.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp98_iDC <- read.flowSet(c("iDC_unstained_022.fcs", "iDC_Panel_023.fcs"), alter.names = TRUE, transformation = FALSE)

exp98_GMCSF <- read.flowSet(c("GMCSF_unstained_020.fcs", "GMCSF_Panel_021.fcs"), alter.names = TRUE, transformation = FALSE)

exp98_MCSF <- read.flowSet(c("MCSF_FCS_unstained_018.fcs", "MCSF_FCS_Panel_019.fcs"), alter.names = TRUE, transformation = FALSE)

exp98_ABT <- read.flowSet(c("AB_Serum_unstained_001.fcs", "AB_Serum_Dump_003.fcs", "AB_Serum_FMO_CD115_009.fcs",
↪  "AB_Serum_FMO_CD11b_007.fcs", "AB_Serum_FMO_CD15_004.fcs", "AB_Serum_FMO_CD64_010.fcs", "exp98_AB_Serum_FMO_CD16_005.fcs",
↪  "exp98_AB_Serum_FMO_CD71_008.fcs", "exp98_AB_Serum_FMO_CD163_011.fcs", "exp98_AB_Serum_FMO_CD14_012.fcs",
↪  "AB_Serum_FMO_HLADR_006.fcs", "AB_Serum_Panel_013.fcs"), alter.names = TRUE, transformation = FALSE)

exp98_Adipo <- read.flowSet(c("Adiponektin_unstained_032.fcs", "Adiponektin_Panel_033.fcs"), alter.names = TRUE, transformation =
↪  FALSE)

exp98_Sero <- read.flowSet(c("Serotonin_unstained_034.fcs", "Serotonin_Panel_035.fcs"), alter.names = TRUE, transformation = FALSE)

exp98_Hermin <- read.flowSet(c("Hermin_unstained_026.fcs", "Hermin_FMO CD15_028.fcs", "Hermin_FMO CD16_029.fcs", "Hermin_FMO
↪  CD64_030.fcs", "Hermin_Panel_031.fcs"), alter.names = TRUE, transformation = FALSE)

exp98_IgG <- read.flowSet(c("IgG_unstained_036.fcs", "IgG_Panel_037.fcs"), alter.names = TRUE, transformation = FALSE)

exp98_NaCl <- read.flowSet(c("NaCl_unstained_014.fcs", "NaCl_Panel_015.fcs"), alter.names = TRUE, transformation = FALSE)

exp98_TGF <- read.flowSet(c("TGF_unstained_024.fcs", "TGF_Panel_025.fcs"), alter.names = TRUE, transformation = FALSE)

###################
### Compensation ###
###################

### AB Teflon ###

exp93T_AB_Comp <- description(exp93_ABT[[12]])\£SPILL
exp93_ABT <- compensate(exp93_ABT, spillover = exp93T_AB_Comp)

exp95T_AB_Comp <- description(exp95_ABT[[12]])\£SPILL
exp95_ABT <- compensate(exp95_ABT, spillover = exp95T_AB_Comp)

exp98T_AB_Comp <- description(exp98_ABT[[12]])\£SPILL
exp98_ABT <- compensate(exp98_ABT, spillover = exp98T_AB_Comp)
```

```
### AB Plastic ###

exp93_AB <- compensate(exp93_AB, spillover = exp93T_AB_Comp)
exp95_AB <- compensate(exp95_AB, spillover = exp95T_AB_Comp)
exp98_AB <- compensate(exp98_AB, spillover = exp98T_AB_Comp)

### Collagen I ###

exp93_ColI <- compensate(exp93_ColI, spillover = exp93T_AB_Comp)
exp95_ColI <- compensate(exp95_ColI, spillover = exp95T_AB_Comp)
exp98_ColI <- compensate(exp98_ColI, spillover = exp98T_AB_Comp)

### Collagen IV ###

exp93_ColIV <- compensate(exp93_ColIV, spillover = exp93T_AB_Comp)
exp95_ColIV <- compensate(exp95_ColIV, spillover = exp95T_AB_Comp)
exp98_ColIV <- compensate(exp98_ColIV, spillover = exp98T_AB_Comp)

### Fibronectin ###

exp93_Fib <- compensate(exp93_Fib, spillover = exp93T_AB_Comp)
exp95_Fib <- compensate(exp95_Fib, spillover = exp95T_AB_Comp)
exp98_Fib <- compensate(exp98_Fib, spillover = exp98T_AB_Comp)

### Laminin ###

exp93_Lam <- compensate(exp93_Lam, spillover = exp93T_AB_Comp)
exp95_Lam <- compensate(exp95_Lam, spillover = exp95T_AB_Comp)
exp98_Lam <- compensate(exp98_Lam, spillover = exp98T_AB_Comp)

### Non-Adherent ###

exp93_Non <- compensate(exp93_Non, spillover = exp93T_AB_Comp)
exp95_Non <- compensate(exp95_Non, spillover = exp95T_AB_Comp)
exp98_Non <- compensate(exp98_Non, spillover = exp98T_AB_Comp)

### iDC ###

exp93_iDC <- compensate(exp93_iDC, spillover = exp93T_AB_Comp)
exp95_iDC <- compensate(exp95_iDC, spillover = exp95T_AB_Comp)
exp98_iDC <- compensate(exp98_iDC, spillover = exp98T_AB_Comp)

### GMCSF ###

exp93_GMCSF <- compensate(exp93_GMCSF, spillover = exp93T_AB_Comp)
exp95_GMCSF <- compensate(exp95_GMCSF, spillover = exp95T_AB_Comp)
exp98_GMCSF <- compensate(exp98_GMCSF, spillover = exp98T_AB_Comp)

### MCSF ###

exp93_MCSF <- compensate(exp93_MCSF, spillover = exp93T_AB_Comp)
exp95_MCSF <- compensate(exp95_MCSF, spillover = exp95T_AB_Comp)
exp98_MCSF <- compensate(exp98_MCSF, spillover = exp98T_AB_Comp)

### Adiponectin ###

exp93_Adipo <- compensate(exp93_Adipo, spillover = exp93T_AB_Comp)
exp95_Adipo <- compensate(exp95_Adipo, spillover = exp95T_AB_Comp)
exp98_Adipo <- compensate(exp98_Adipo, spillover = exp98T_AB_Comp)

### Serotonin ###

exp93_Sero <- compensate(exp93_Sero, spillover = exp93T_AB_Comp)
exp95_Sero <- compensate(exp95_Sero, spillover = exp95T_AB_Comp)
exp98_Sero <- compensate(exp98_Sero, spillover = exp98T_AB_Comp)

### Hermin ###

exp93_Hermin <- compensate(exp93_Hermin, spillover = exp93T_AB_Comp)
exp95_Hermin <- compensate(exp95_Hermin, spillover = exp95T_AB_Comp)
exp98_Hermin <- compensate(exp98_Hermin, spillover = exp98T_AB_Comp)

### IgG ###

exp93_IgG <- compensate(exp93_IgG, spillover = exp93T_AB_Comp)
exp95_IgG <- compensate(exp95_IgG, spillover = exp95T_AB_Comp)
exp98_IgG <- compensate(exp98_IgG, spillover = exp98T_AB_Comp)

### NaCl ###

exp93_NaCl <- compensate(exp93_NaCl, spillover = exp93T_AB_Comp)
exp95_NaCl <- compensate(exp95_NaCl, spillover = exp95T_AB_Comp)
```

```
exp98_NaCl <- compensate(exp98_NaCl, spillover = exp98T_AB_Comp)

### TGF ###

exp93_TGF <- compensate(exp93_TGF, spillover = exp93T_AB_Comp)
exp95_TGF<- compensate(exp95_TGF, spillover = exp95T_AB_Comp)
exp98_TGF <- compensate(exp98_TGF, spillover = exp98T_AB_Comp)

############################################
### Logicle Transformation of Flow Data ###
############################################

### AB-Teflon ###

lgcl <- estimateLogicle(exp93_ABT[[12]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_ABT_lgcl <- transform(exp93_ABT, lgcl)
lgcl <- estimateLogicle(exp95_ABT[[12]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_ABT_lgcl <- transform(exp95_ABT, lgcl)
lgcl <- estimateLogicle(exp98_ABT[[12]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_ABT_lgcl <- transform(exp98_ABT, lgcl)

### AB Plastic ###

lgcl <- estimateLogicle(exp93_AB[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_AB_lgcl <- transform(exp93_AB, lgcl)
lgcl <- estimateLogicle(exp95_AB[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_AB_lgcl <- transform(exp95_AB, lgcl)
lgcl <- estimateLogicle(exp98_AB[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_AB_lgcl <- transform(exp98_AB, lgcl)

### Collagen I ###

lgcl <- estimateLogicle(exp93_ColI[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_ColI_lgcl <- transform(exp93_ColI, lgcl)
lgcl <- estimateLogicle(exp95_ColI[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_ColI_lgcl <- transform(exp95_ColI, lgcl)
lgcl <- estimateLogicle(exp98_ColI[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_ColI_lgcl <- transform(exp98_ColI, lgcl)

### Collagen IV ###

lgcl <- estimateLogicle(exp93_ColIV[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_ColIV_lgcl <- transform(exp93_ColIV, lgcl)
lgcl <- estimateLogicle(exp95_ColIV[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_ColIV_lgcl <- transform(exp95_ColIV, lgcl)
lgcl <- estimateLogicle(exp98_ColIV[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_ColIV_lgcl <- transform(exp98_ColIV, lgcl)

### Fibronectin ###

lgcl <- estimateLogicle(exp93_Fib[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_Fib_lgcl <- transform(exp93_Fib, lgcl)
lgcl <- estimateLogicle(exp95_Fib[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_Fib_lgcl <- transform(exp95_Fib, lgcl)
lgcl <- estimateLogicle(exp98_Fib[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_Fib_lgcl <- transform(exp98_Fib, lgcl)

### Laminin ###

lgcl <- estimateLogicle(exp93_Lam[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_Lam_lgcl <- transform(exp93_Lam, lgcl)
lgcl <- estimateLogicle(exp95_Lam[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_Lam_lgcl <- transform(exp95_Lam, lgcl)
lgcl <- estimateLogicle(exp98_Lam[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
```

```
exp98_Lam_lgcl <- transform(exp98_Lam, lgcl)

### Non-Adherent ###

lgcl <- estimateLogicle(exp93_Non[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_Non_lgcl <- transform(exp93_Non, lgcl)
lgcl <- estimateLogicle(exp95_Non[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_Non_lgcl <- transform(exp95_Non, lgcl)
lgcl <- estimateLogicle(exp98_Non[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_Non_lgcl <- transform(exp98_Non, lgcl)

### iDC ###

lgcl <- estimateLogicle(exp93_iDC[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_iDC_lgcl <- transform(exp93_iDC, lgcl)
lgcl <- estimateLogicle(exp95_iDC[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_iDC_lgcl <- transform(exp95_iDC, lgcl)
lgcl <- estimateLogicle(exp98_iDC[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_iDC_lgcl <- transform(exp98_iDC, lgcl)

### GMCSF ###

lgcl <- estimateLogicle(exp93_GMCSF[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_GMCSF_lgcl <- transform(exp93_GMCSF, lgcl)
lgcl <- estimateLogicle(exp95_GMCSF[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_GMCSF_lgcl <- transform(exp95_GMCSF, lgcl)
lgcl <- estimateLogicle(exp98_GMCSF[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_GMCSF_lgcl <- transform(exp98_GMCSF, lgcl)

### MCSF ###

lgcl <- estimateLogicle(exp93_MCSF[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_MCSF_lgcl <- transform(exp93_MCSF, lgcl)
lgcl <- estimateLogicle(exp95_MCSF[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_MCSF_lgcl <- transform(exp95_MCSF, lgcl)
lgcl <- estimateLogicle(exp98_MCSF[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_MCSF_lgcl <- transform(exp98_MCSF, lgcl)

### Adiponectin ###

lgcl <- estimateLogicle(exp93_Adipo[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_Adipo_lgcl <- transform(exp93_Adipo, lgcl)
lgcl <- estimateLogicle(exp95_Adipo[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_Adipo_lgcl <- transform(exp95_Adipo, lgcl)
lgcl <- estimateLogicle(exp98_Adipo[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_Adipo_lgcl <- transform(exp98_Adipo, lgcl)

### Serotonin ###

lgcl <- estimateLogicle(exp93_Sero[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_Sero_lgcl <- transform(exp93_Sero, lgcl)
lgcl <- estimateLogicle(exp95_Sero[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_Sero_lgcl <- transform(exp95_Sero, lgcl)
lgcl <- estimateLogicle(exp98_Sero[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_Sero_lgcl <- transform(exp98_Sero, lgcl)

### Hermin ###

lgcl <- estimateLogicle(exp93_Hermin[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_Hermin_lgcl <- transform(exp93_Hermin, lgcl)
lgcl <- estimateLogicle(exp95_Hermin[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_Hermin_lgcl <- transform(exp95_Hermin, lgcl)
```

```
lgcl <- estimateLogicle(exp98_Hermin[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_Hermin_lgcl <- transform(exp98_Hermin, lgcl)

### IgG ###

lgcl <- estimateLogicle(exp93_IgG[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_IgG_lgcl <- transform(exp93_IgG, lgcl)
lgcl <- estimateLogicle(exp95_IgG[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_IgG_lgcl <- transform(exp95_IgG, lgcl)
lgcl <- estimateLogicle(exp98_IgG[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_IgG_lgcl <- transform(exp98_IgG, lgcl)

### NaCl ###

lgcl <- estimateLogicle(exp93_NaCl[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_NaCl_lgcl <- transform(exp93_NaCl, lgcl)
lgcl <- estimateLogicle(exp95_NaCl[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_NaCl_lgcl <- transform(exp95_NaCl, lgcl)
lgcl <- estimateLogicle(exp98_NaCl[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_NaCl_lgcl <- transform(exp98_NaCl, lgcl)

### TGF ###

lgcl <- estimateLogicle(exp93_TGF[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp93_TGF_lgcl <- transform(exp93_TGF, lgcl)
lgcl <- estimateLogicle(exp95_TGF[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp95_TGF_lgcl <- transform(exp95_TGF, lgcl)
lgcl <- estimateLogicle(exp98_TGF[[2]], channels = c("FITC.A", "PE.A", "PerCP.Cy5.5.A", "PE.Cy7.A", "APC.A", "APC.H7.A", "BV421.A",
↪  "BV510.A", "BV650.A", "BV711.A"))
exp98_TGF_lgcl <- transform(exp98_TGF, lgcl)

#######################
### Gating Strategy ###
#######################

####################
## FSC-SSC Gating ##
####################

setwd("£FSCDIR")

##Setting up the lattice plot ##

settings <- flowViz.par.get()
settings\£panel.background\£col <- c("white")
settings\£axis.line\£col <- c("black")
settings\£axis.text\£col <- c("black")
flowViz.par.set(settings)

### AB Teflon ###
ABT <- norm2Filter("FSC.A", "SSC.A", filterId = "ABT", scale = 1.6)
pdf(file = "£FSCDIR/FSCSSC.ABTeflon.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_ABT_lgcl[[12]], smooth = FALSE, pos = c(0.1, 0.9), filter = ABT, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_ABT_lgcl <- Subset(exp93_ABT_lgcl, ABT)
exp93_ABT_morph <- as.numeric(fsApply(exp93_ABT_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.ABTeflon.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_ABT_lgcl[[12]], smooth = FALSE, pos = c(0.1, 0.9), filter = ABT, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_ABT_lgcl <- Subset(exp95_ABT_lgcl, ABT)
exp95_ABT_morph <- as.numeric(fsApply(exp95_ABT_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.ABTeflon.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_ABT_lgcl[[12]], smooth = FALSE, pos = c(0.1, 0.9), filter = ABT, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_ABT_lgcl <- Subset(exp98_ABT_lgcl, ABT)
exp98_ABT_morph <- as.numeric(fsApply(exp98_ABT_lgcl, nrow, use.exprs = TRUE))

### AB Plastic ###
```

```
AB <- norm2Filter("FSC.A", "SSC.A", filterId = "AB", scale = 1.8)
pdf(file = "£FSCDIR/FSCSSC.ABTcPlastic.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_AB_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = AB, stats = TRUE, xlim = c(0,266144), ylim
↪   = c(0,266144))
dev.off()
exp93_AB_lgcl <- Subset(exp93_AB_lgcl, AB)
exp93_AB_morph <- as.numeric(fsApply(exp93_AB_lgcl, nrow, use.exprs = TRUE))

AB <- norm2Filter("FSC.A", "SSC.A", filterId = "AB", scale = 1.6)
pdf(file = "£FSCDIR/FSCSSC.ABTcPlastic.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_AB_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = AB, stats = TRUE, xlim = c(0,266144), ylim
↪   = c(0,266144))
dev.off()
exp95_AB_lgcl <- Subset(exp95_AB_lgcl, AB)
exp95_AB_morph <- as.numeric(fsApply(exp95_AB_lgcl, nrow, use.exprs = TRUE))

AB <- norm2Filter("FSC.A", "SSC.A", filterId = "AB", scale = 1.45)
pdf(file = "£FSCDIR/FSCSSC.ABTcPlastic.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_AB_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = AB, stats = TRUE, xlim = c(0,266144), ylim
↪   = c(0,266144))
dev.off()
exp98_AB_lgcl <- Subset(exp98_AB_lgcl, AB)
exp98_AB_morph <- as.numeric(fsApply(exp98_AB_lgcl, nrow, use.exprs = TRUE))

### Collagen I ###

ColI <- norm2Filter("FSC.A", "SSC.A", filterId = "ColI", scale = 1.8)
pdf(file = "£FSCDIR/FSCSSC.CollagenI.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_ColI_lgcl[[1]], smooth = FALSE, pos = c(0.9, 0.9), filter = ColI, stats = TRUE, xlim = c(0,266144),
↪   ylim = c(0,266144))
dev.off()
exp93_ColI_lgcl <- Subset(exp93_ColI_lgcl, ColI)
exp93_ColI_morph <- as.numeric(fsApply(exp93_ColI_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.CollagenI.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_ColI_lgcl[[1]], smooth = FALSE, pos = c(0.9, 0.9), filter = ColI, stats = TRUE, xlim = c(0,266144),
↪   ylim = c(0,266144))
dev.off()
exp95_ColI_lgcl <- Subset(exp95_ColI_lgcl, ColI)
exp95_ColI_morph <- as.numeric(fsApply(exp95_ColI_lgcl, nrow, use.exprs = TRUE))

ColI <- norm2Filter("FSC.A", "SSC.A", filterId = "ColI", scale = 1.6)
pdf(file = "£FSCDIR/FSCSSC.CollagenI.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_ColI_lgcl[[1]], smooth = FALSE, pos = c(0.9, 0.9), filter = ColI, stats = TRUE, xlim = c(0,266144),
↪   ylim = c(0,266144))
dev.off()
exp98_ColI_lgcl <- Subset(exp98_ColI_lgcl, ColI)
exp98_ColI_morph <- as.numeric(fsApply(exp98_ColI_lgcl, nrow, use.exprs = TRUE))

### Collagen IV ###

ColIV <- norm2Filter("FSC.A", "SSC.A", filterId = "ColIV", scale = 1.8)
pdf(file = "£FSCDIR/FSCSSC.CollagenIV.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_ColIV_lgcl[[1]], smooth = FALSE, pos = c(0.9, 0.9), filter = ColIV, stats = TRUE, xlim = c(0,266144),
↪   ylim = c(0,266144))
dev.off()
exp93_ColIV_lgcl <- Subset(exp93_ColIV_lgcl, ColIV)
exp93_ColIV_morph <- as.numeric(fsApply(exp93_ColIV_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.CollagenIV.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_ColIV_lgcl[[1]], smooth = FALSE, pos = c(0.9, 0.9), filter = ColIV, stats = TRUE, xlim = c(0,266144),
↪   ylim = c(0,266144))
dev.off()
exp95_ColIV_lgcl <- Subset(exp95_ColIV_lgcl, ColIV)
exp95_ColIV_morph <- as.numeric(fsApply(exp95_ColIV_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.CollagenIV.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_ColIV_lgcl[[1]], smooth = FALSE, pos = c(0.9, 0.9), filter = ColIV, stats = TRUE, xlim = c(0,266144),
↪   ylim = c(0,266144))
dev.off()
exp98_ColIV_lgcl <- Subset(exp98_ColIV_lgcl, ColIV)
exp98_ColIV_morph <- as.numeric(fsApply(exp98_ColIV_lgcl, nrow, use.exprs = TRUE))

### Fibronectin ###

Fib <- norm2Filter("FSC.A", "SSC.A", filterId = "Fib", scale = 1.8)
pdf(file = "£FSCDIR/FSCSSC.Fibronectin.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_Fib_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Fib, stats = TRUE, xlim = c(0,266144),
↪   ylim = c(0,266144))
dev.off()
exp93_Fib_lgcl <- Subset(exp93_Fib_lgcl, Fib)
```

```
exp93_Fib_morph <- as.numeric(fsApply(exp93_Fib_lgcl, nrow, use.exprs = TRUE))

Fib <- norm2Filter("FSC.A", "SSC.A", filterId = "Fib", scale = 1.3)
pdf(file = "£FSCDIR/FSCSSC.Fibronectin.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_Fib_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Fib, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_Fib_lgcl <- Subset(exp95_Fib_lgcl, Fib)
exp95_Fib_morph <- as.numeric(fsApply(exp95_Fib_lgcl, nrow, use.exprs = TRUE))

Fib <- norm2Filter("FSC.A", "SSC.A", filterId = "Fib", scale = 1.6)
pdf(file = "£FSCDIR/FSCSSC.Fibronectin.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_Fib_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Fib, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_Fib_lgcl <- Subset(exp98_Fib_lgcl, Fib)
exp98_Fib_morph <- as.numeric(fsApply(exp98_Fib_lgcl, nrow, use.exprs = TRUE))

### Laminin ###

Lam <- norm2Filter("FSC.A", "SSC.A", filterId = "Lam", scale = 1.5)
pdf(file = "£FSCDIR/FSCSSC.Laminin.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_Lam_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Lam, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_Lam_lgcl <- Subset(exp93_Lam_lgcl, Lam)
exp93_Lam_morph <- as.numeric(fsApply(exp93_Lam_lgcl, nrow, use.exprs = TRUE))

Lam <- norm2Filter("FSC.A", "SSC.A", filterId = "Lam", scale = 1.2)
pdf(file = "£FSCDIR/FSCSSC.Laminin.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_Lam_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Lam, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_Lam_lgcl <- Subset(exp95_Lam_lgcl, Lam)
exp95_Lam_morph <- as.numeric(fsApply(exp95_Lam_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.Laminin.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_Lam_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Lam, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_Lam_lgcl <- Subset(exp98_Lam_lgcl, Lam)
exp98_Lam_morph <- as.numeric(fsApply(exp98_Lam_lgcl, nrow, use.exprs = TRUE))

### Non-Adherent ###

NonAd <- norm2Filter("FSC.A", "SSC.A", filterId = "NonAd", scale = 1.8)
pdf(file = "£FSCDIR/FSCSSC.NonAdherent.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_Non_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = NonAd, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_Non_lgcl <- Subset(exp93_Non_lgcl, NonAd)
exp93_Non_morph <- as.numeric(fsApply(exp93_Non_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.NonAdherent.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_Non_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = NonAd, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_Non_lgcl <- Subset(exp95_Non_lgcl, NonAd)
exp95_Non_morph <- as.numeric(fsApply(exp95_Non_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.NonAdherent.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_Non_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = NonAd, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_Non_lgcl <- Subset(exp98_Non_lgcl, NonAd)
exp98_Non_morph <- as.numeric(fsApply(exp98_Non_lgcl, nrow, use.exprs = TRUE))

### iDC ###

iDC <- norm2Filter("FSC.A", "SSC.A", filterId = "iDC", scale = 1.5)
pdf(file = "£FSCDIR/FSCSSC.iDC.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_iDC_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = iDC, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_iDC_lgcl <- Subset(exp93_iDC_lgcl, iDC)
exp93_iDC_morph <- as.numeric(fsApply(exp93_iDC_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.iDC.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_iDC_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = iDC, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_iDC_lgcl <- Subset(exp95_iDC_lgcl, iDC)
```

```
exp95_iDC_morph <- as.numeric(fsApply(exp95_iDC_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.iDC.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_iDC_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = iDC, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_iDC_lgcl <- Subset(exp98_iDC_lgcl, iDC)
exp98_iDC_morph <- as.numeric(fsApply(exp98_iDC_lgcl, nrow, use.exprs = TRUE))

### GMCSF ###

GMCSF <- norm2Filter("FSC.A", "SSC.A", filterId = "GMCSF", scale = 1.5)
pdf(file = "£FSCDIR/FSCSSC.GMCSF.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_GMCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = GMCSF, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_GMCSF_lgcl <- Subset(exp93_GMCSF_lgcl, GMCSF)
exp93_GMCSF_morph <- as.numeric(fsApply(exp93_GMCSF_lgcl, nrow, use.exprs = TRUE))

m <- matrix(c(60000, 100000, 150000, 175000, 165000, 150000, 130000, 100000, 60000,
              30000, 15000, 30000, 50000, 110000, 120000, 120000, 120000, 75000), nrow = 9)
colnames(m) <- c("FSC.A", "SSC.A")
morph_Gate <- polygonGate(m)
exp95_GMCSF_lgcl <- Subset(exp95_GMCSF_lgcl, morph_Gate)

pdf(file = "£FSCDIR/FSCSSC.GMCSF.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_GMCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = GMCSF, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_GMCSF_lgcl <- Subset(exp95_GMCSF_lgcl, GMCSF)
exp95_GMCSF_morph <- as.numeric(fsApply(exp95_GMCSF_lgcl, nrow, use.exprs = TRUE))

GMCSF <- norm2Filter("FSC.A", "SSC.A", filterId = "GMCSF", scale = 1.8)
pdf(file = "£FSCDIR/FSCSSC.GMCSF.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_GMCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = GMCSF, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_GMCSF_lgcl <- Subset(exp98_GMCSF_lgcl, GMCSF)
exp98_GMCSF_morph <- as.numeric(fsApply(exp98_GMCSF_lgcl, nrow, use.exprs = TRUE))

### MCSF ###
MCSF <- norm2Filter("FSC.A", "SSC.A", filterId = "MCSF", scale = 1.7)

pdf(file = "£FSCDIR/FSCSSC.MCSF.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_MCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = MCSF, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_MCSF_lgcl <- Subset(exp93_MCSF_lgcl, MCSF)
exp93_MCSF_morph <- as.numeric(fsApply(exp93_MCSF_lgcl, nrow, use.exprs = TRUE))

exp95_MCSF_lgcl <- Subset(exp95_MCSF_lgcl, morph_Gate)
pdf(file = "£FSCDIR/FSCSSC.MCSF.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_MCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = MCSF, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_MCSF_lgcl <- Subset(exp95_MCSF_lgcl, MCSF)
exp95_MCSF_morph <- as.numeric(fsApply(exp95_MCSF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.MCSF.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_MCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = MCSF, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_MCSF_lgcl <- Subset(exp98_MCSF_lgcl, MCSF)
exp98_MCSF_morph <- as.numeric(fsApply(exp98_MCSF_lgcl, nrow, use.exprs = TRUE))

### Adiponectin ###

Adipo <- norm2Filter("FSC.A", "SSC.A", filterId = "Adipo", scale = 1.6)

pdf(file = "£FSCDIR/FSCSSC.Adiponectin.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_Adipo_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Adipo, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_Adipo_lgcl <- Subset(exp93_Adipo_lgcl, Adipo)
exp93_Adipo_morph <- as.numeric(fsApply(exp93_Adipo_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.Adiponectin.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_Adipo_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Adipo, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_Adipo_lgcl <- Subset(exp95_Adipo_lgcl, Adipo)
exp95_Adipo_morph <- as.numeric(fsApply(exp95_Adipo_lgcl, nrow, use.exprs = TRUE))
```

```
pdf(file = "£FSCDIR/FSCSSC.Adiponectin.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_Adipo_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Adipo, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_Adipo_lgcl <- Subset(exp98_Adipo_lgcl, Adipo)
exp98_Adipo_morph <- as.numeric(fsApply(exp98_Adipo_lgcl, nrow, use.exprs = TRUE))

### Serotonin ###

Sero <- norm2Filter("FSC.A", "SSC.A", filterId = "Sero", scale = 1.6)

pdf(file = "£FSCDIR/FSCSSC.Serotonin.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_Sero_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Sero, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_Sero_lgcl <- Subset(exp93_Sero_lgcl, Sero)
exp93_Sero_morph <- as.numeric(fsApply(exp93_Sero_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.Serotonin.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_Sero_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Sero, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_Sero_lgcl <- Subset(exp95_Sero_lgcl, Sero)
exp95_Sero_morph <- as.numeric(fsApply(exp95_Sero_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.Serotonin.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_Sero_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Sero, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_Sero_lgcl <- Subset(exp98_Sero_lgcl, Sero)
exp98_Sero_morph <- as.numeric(fsApply(exp98_Sero_lgcl, nrow, use.exprs = TRUE))

### Hermin ###

Heme <- norm2Filter("FSC.A", "SSC.A", filterId = "Heme", scale = 1.6)

pdf(file = "£FSCDIR/FSCSSC.Hermin.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_Hermin_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Heme, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_Hermin_lgcl <- Subset(exp93_Hermin_lgcl, Heme)
exp93_Hermin_morph <- as.numeric(fsApply(exp93_Hermin_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.Hermin.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_Hermin_lgcl[[5]], smooth = FALSE, pos = c(0.9, 0.9), filter = Heme, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_Hermin_lgcl <- Subset(exp95_Hermin_lgcl, Heme)
exp95_Hermin_morph <- as.numeric(fsApply(exp95_Hermin_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.Hermin.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_Hermin_lgcl[[5]], smooth = FALSE, pos = c(0.9, 0.9), filter = Heme, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_Hermin_lgcl <- Subset(exp98_Hermin_lgcl, Heme)
exp98_Hermin_morph <- as.numeric(fsApply(exp98_Hermin_lgcl, nrow, use.exprs = TRUE))

### IgG ###

IgG <- norm2Filter("FSC.A", "SSC.A", filterId = "IgG", scale = 1.6)

pdf(file = "£FSCDIR/FSCSSC.IgG.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_IgG_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = IgG, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_IgG_lgcl <- Subset(exp93_IgG_lgcl, IgG)
exp93_IgG_morph <- as.numeric(fsApply(exp93_IgG_lgcl, nrow, use.exprs = TRUE))

m <- matrix(c(60000, 100000, 150000, 175000, 165000, 150000, 130000, 100000, 60000,
              30000, 15000, 30000, 50000, 110000, 120000, 120000, 120000, 75000), nrow = 9)
colnames(m) <- c("FSC.A", "SSC.A")
morph_Gate <- polygonGate(m)
exp95_IgG_lgcl <- Subset(exp95_IgG_lgcl, morph_Gate)

pdf(file = "£FSCDIR/FSCSSC.IgG.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_IgG_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = IgG, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_IgG_lgcl <- Subset(exp95_IgG_lgcl, IgG)
exp95_IgG_morph <- as.numeric(fsApply(exp95_IgG_lgcl, nrow, use.exprs = TRUE))
```

```
pdf(file = "£FSCDIR/FSCSSC.IgG.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_IgG_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = IgG, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_IgG_lgcl <- Subset(exp98_IgG_lgcl, IgG)
exp98_IgG_morph <- as.numeric(fsApply(exp98_IgG_lgcl, nrow, use.exprs = TRUE))

### NaCl ###

NaCl <- norm2Filter("FSC.A", "SSC.A", filterId = "NaCl", scale = 1.4)

pdf(file = "£FSCDIR/FSCSSC.NaCl.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_NaCl_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = NaCl, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_NaCl_lgcl <- Subset(exp93_NaCl_lgcl, NaCl)
exp93_NaCl_morph <- as.numeric(fsApply(exp93_NaCl_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.NaCl.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_NaCl_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = NaCl, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_NaCl_lgcl <- Subset(exp95_NaCl_lgcl, NaCl)
exp95_NaCl_morph <- as.numeric(fsApply(exp95_NaCl_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.NaCl.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_NaCl_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = NaCl, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_NaCl_lgcl <- Subset(exp98_NaCl_lgcl, NaCl)
exp98_NaCl_morph <- as.numeric(fsApply(exp98_NaCl_lgcl, nrow, use.exprs = TRUE))

### TGF ###

TGF <- norm2Filter("FSC.A", "SSC.A", filterId = "TGF", scale = 1.5)

pdf(file = "£FSCDIR/FSCSSC.TGF.exp93.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp93_TGF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = TGF, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_TGF_lgcl <- Subset(exp93_TGF_lgcl, TGF)
exp93_TGF_morph <- as.numeric(fsApply(exp93_TGF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.TGF.exp95.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp95_TGF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = TGF, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_TGF_lgcl <- Subset(exp95_TGF_lgcl, TGF)
exp95_TGF_morph <- as.numeric(fsApply(exp95_TGF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£FSCDIR/FSCSSC.TGF.exp98.pdf", width = 5, height = 5)
xyplot(\`SSC.A\`~\`FSC.A\`, exp98_TGF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = TGF, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_TGF_lgcl <- Subset(exp98_TGF_lgcl, TGF)
exp98_TGF_morph <- as.numeric(fsApply(exp98_TGF_lgcl, nrow, use.exprs = TRUE))

######################
## Live-Dead Gating ##
######################

setwd("£LIVEDIR")

Dump <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.8), filterId = "Dump")
exp98_Dump <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.55), filterId = "exp98_Dump")
### AB Teflon ###

pdf(file = "£LIVEDIR/Live.ABTeflon.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_ABT_lgcl[[12]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_ABT_lgcl <- Subset(exp93_ABT_lgcl, Dump)
exp93_ABT_live <- as.numeric(fsApply(exp93_ABT_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.ABTeflon.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_ABT_lgcl[[12]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_ABT_lgcl <- Subset(exp95_ABT_lgcl, Dump)
exp95_ABT_live <- as.numeric(fsApply(exp95_ABT_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.ABTeflon.exp98.pdf")
```

```
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_ABT_lgcl[[12]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_ABT_lgcl <- Subset(exp98_ABT_lgcl, exp98_Dump)
exp98_ABT_live <- as.numeric(fsApply(exp98_ABT_lgcl, nrow, use.exprs = TRUE))

### AB Plastic ###

pdf(file = "£LIVEDIR/Live.ABTcPlastic.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_AB_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_AB_lgcl <- Subset(exp93_AB_lgcl, Dump)
exp93_AB_live <- as.numeric(fsApply(exp93_AB_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.ABTcPlastic.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_AB_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_AB_lgcl <- Subset(exp95_AB_lgcl, Dump)
exp95_AB_live <- as.numeric(fsApply(exp95_AB_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.ABTcPlastic.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_AB_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_AB_lgcl <- Subset(exp98_AB_lgcl, exp98_Dump)
exp98_AB_live <- as.numeric(fsApply(exp98_AB_lgcl, nrow, use.exprs = TRUE))

### Collagen I ###

ColI <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.5), filterId = "ColI")
pdf(file = "£LIVEDIR/Live.CollagenI.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_ColI_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = ColI, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_ColI_lgcl <- Subset(exp93_ColI_lgcl, ColI)
exp93_ColI_live <- as.numeric(fsApply(exp93_ColI_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.CollagenI.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_ColI_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_ColI_lgcl <- Subset(exp95_ColI_lgcl, Dump)
exp95_ColI_live <- as.numeric(fsApply(exp95_ColI_lgcl, nrow, use.exprs = TRUE))

ColI <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.4), filterId = "ColI")
pdf(file = "£LIVEDIR/Live.CollagenI.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_ColI_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = ColI, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_ColI_lgcl <- Subset(exp98_ColI_lgcl, ColI)
exp98_ColI_live <- as.numeric(fsApply(exp98_ColI_lgcl, nrow, use.exprs = TRUE))

### Collagen IV ###

ColIV <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.35), filterId = "ColIV")
pdf(file = "£LIVEDIR/Live.CollagenIV.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_ColIV_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = ColIV, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_ColIV_lgcl <- Subset(exp93_ColIV_lgcl, ColIV)
exp93_ColIV_live <- as.numeric(fsApply(exp93_ColIV_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.CollagenIV.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_ColIV_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_ColIV_lgcl <- Subset(exp95_ColIV_lgcl, Dump)
exp95_ColIV_live <- as.numeric(fsApply(exp95_ColIV_lgcl, nrow, use.exprs = TRUE))

ColI <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.4), filterId = "ColI")
pdf(file = "£LIVEDIR/Live.CollagenIV.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_ColIV_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_ColIV_lgcl <- Subset(exp98_ColIV_lgcl, exp98_Dump)
exp98_ColIV_live <- as.numeric(fsApply(exp98_ColIV_lgcl, nrow, use.exprs = TRUE))

### Fibronectin ###

pdf(file = "£LIVEDIR/Live.Fibronectin.exp93.pdf")
```

```
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_Fib_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_Fib_lgcl <- Subset(exp93_Fib_lgcl, Dump)
exp93_Fib_live <- as.numeric(fsApply(exp93_Fib_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.Fibronectin.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_Fib_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_Fib_lgcl <- Subset(exp95_Fib_lgcl, Dump)
exp95_Fib_live <- as.numeric(fsApply(exp95_Fib_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.Fibronectin.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_Fib_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_Fib_lgcl <- Subset(exp98_Fib_lgcl, exp98_Dump)
exp98_Fib_live <- as.numeric(fsApply(exp98_Fib_lgcl, nrow, use.exprs = TRUE))

### Laminin ###

pdf(file = "£LIVEDIR/Live.Laminin.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_Lam_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_Lam_lgcl <- Subset(exp93_Lam_lgcl, Dump)
exp93_Lam_live <- as.numeric(fsApply(exp93_Lam_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.Laminin.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_Lam_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_Lam_lgcl <- Subset(exp95_Lam_lgcl, Dump)
exp95_Lam_live <- as.numeric(fsApply(exp95_Lam_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.Laminin.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_Lam_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_Lam_lgcl <- Subset(exp98_Lam_lgcl, exp98_Dump)
exp98_Lam_live <- as.numeric(fsApply(exp98_Lam_lgcl, nrow, use.exprs = TRUE))

### Non-Adherent ###

Non <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.5), filterId = "Non")
pdf(file = "£LIVEDIR/Live.NonAdherent.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_Non_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Non, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_Non_lgcl <- Subset(exp93_Non_lgcl, Non)
exp93_Non_live <- as.numeric(fsApply(exp93_Non_lgcl, nrow, use.exprs = TRUE))

Non <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.6), filterId = "Non")
pdf(file = "£LIVEDIR/Live.NonAdherent.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_Non_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Non, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_Non_lgcl <- Subset(exp95_Non_lgcl, Non)
exp95_Non_live <- as.numeric(fsApply(exp95_Non_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.NonAdherent.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_Non_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_Non_lgcl <- Subset(exp98_Non_lgcl, exp98_Dump)
exp98_Non_live <- as.numeric(fsApply(exp98_Non_lgcl, nrow, use.exprs = TRUE))

### iDC ###

pdf(file = "£LIVEDIR/Live.iDC.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_iDC_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_iDC_lgcl <- Subset(exp93_iDC_lgcl, Dump)
exp93_iDC_live <- as.numeric(fsApply(exp93_iDC_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.iDC.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_iDC_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_iDC_lgcl <- Subset(exp95_iDC_lgcl, Dump)
```

```
exp95_iDC_live <- as.numeric(fsApply(exp95_iDC_lgcl, nrow, use.exprs = TRUE))

iDC <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.8), filterId = "iDC")
pdf(file = "£LIVEDIR/Live.iDC.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_iDC_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = iDC, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_iDC_lgcl <- Subset(exp98_iDC_lgcl, iDC)
exp98_iDC_live <- as.numeric(fsApply(exp98_iDC_lgcl, nrow, use.exprs = TRUE))

### GMCSF ###

GMCSF <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.9), filterId = "GMCSF")
pdf(file = "£LIVEDIR/Live.GMCSF.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_GMCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = GMCSF, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_GMCSF_lgcl <- Subset(exp93_GMCSF_lgcl, GMCSF)
exp93_GMCSF_live <- as.numeric(fsApply(exp93_GMCSF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.GMCSF.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_GMCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_GMCSF_lgcl <- Subset(exp95_GMCSF_lgcl, Dump)
exp95_GMCSF_live <- as.numeric(fsApply(exp95_GMCSF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.GMCSF.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_GMCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = GMCSF, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_GMCSF_lgcl <- Subset(exp98_GMCSF_lgcl, GMCSF)
exp98_GMCSF_live <- as.numeric(fsApply(exp98_GMCSF_lgcl, nrow, use.exprs = TRUE))

### MCSF ###

pdf(file = "£LIVEDIR/Live.MCSF.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_MCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_MCSF_lgcl <- Subset(exp93_MCSF_lgcl, Dump)
exp93_MCSF_live <- as.numeric(fsApply(exp93_MCSF_lgcl, nrow, use.exprs = TRUE))

MCSF <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 2.5), filterId = "MCSF")
pdf(file = "£LIVEDIR/Live.MCSF.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_MCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = MCSF, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_MCSF_lgcl <- Subset(exp95_MCSF_lgcl, MCSF)
exp95_MCSF_live <- as.numeric(fsApply(exp95_MCSF_lgcl, nrow, use.exprs = TRUE))

MCSF <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 2), filterId = "MCSF")
pdf(file = "£LIVEDIR/Live.MCSF.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_MCSF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = MCSF, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_MCSF_lgcl <- Subset(exp98_MCSF_lgcl, MCSF)
exp98_MCSF_live <- as.numeric(fsApply(exp98_MCSF_lgcl, nrow, use.exprs = TRUE))

### Adiponectin ###

pdf(file = "£LIVEDIR/Live.Adiponectin.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_Adipo_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_Adipo_lgcl <- Subset(exp93_Adipo_lgcl, Dump)
exp93_Adipo_live <- as.numeric(fsApply(exp93_Adipo_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.Adiponectin.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_Adipo_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_Adipo_lgcl <- Subset(exp95_Adipo_lgcl, Dump)
exp95_Adipo_live <- as.numeric(fsApply(exp95_Adipo_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.Adiponectin.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_Adipo_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_Adipo_lgcl <- Subset(exp98_Adipo_lgcl, exp98_Dump)
exp98_Adipo_live <- as.numeric(fsApply(exp98_Adipo_lgcl, nrow, use.exprs = TRUE))
```

```
### Serotonin ###

Sero <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 2.3), filterId = "Sero")
pdf(file = "£LIVEDIR/Live.Serotonin.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_Sero_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Sero, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_Sero_lgcl <- Subset(exp93_Sero_lgcl, Sero)
exp93_Sero_live <- as.numeric(fsApply(exp93_Sero_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.Serotonin.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_Sero_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Sero, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_Sero_lgcl <- Subset(exp95_Sero_lgcl, Sero)
exp95_Sero_live <- as.numeric(fsApply(exp95_Sero_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.Serotonin.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_Sero_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Sero, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_Sero_lgcl <- Subset(exp98_Sero_lgcl, Sero)
exp98_Sero_live <- as.numeric(fsApply(exp98_Sero_lgcl, nrow, use.exprs = TRUE))

### Hermin ###

Hermin <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 2.3), filterId = "Hermin")
pdf(file = "£LIVEDIR/Live.Hermin.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_Hermin_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Hermin, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_Hermin_lgcl <- Subset(exp93_Hermin_lgcl, Hermin)
exp93_Hermin_live <- as.numeric(fsApply(exp93_Hermin_lgcl, nrow, use.exprs = TRUE))

Hermin <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 2.2), filterId = "Hermin")
pdf(file = "£LIVEDIR/Live.Hermin.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_Hermin_lgcl[[5]], smooth = FALSE, pos = c(0.9, 0.9), filter = Hermin, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_Hermin_lgcl <- Subset(exp95_Hermin_lgcl, Hermin)
exp95_Hermin_live <- as.numeric(fsApply(exp95_Hermin_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.Hermin.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_Hermin_lgcl[[5]], smooth = FALSE, pos = c(0.9, 0.9), filter = Hermin, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_Hermin_lgcl <- Subset(exp98_Hermin_lgcl, Hermin)
exp98_Hermin_live <- as.numeric(fsApply(exp98_Hermin_lgcl, nrow, use.exprs = TRUE))

### IgG ###

pdf(file = "£LIVEDIR/Live.IgG.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_IgG_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_IgG_lgcl <- Subset(exp93_IgG_lgcl, Dump)
exp93_IgG_live <- as.numeric(fsApply(exp93_IgG_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.IgG.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_IgG_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_IgG_lgcl <- Subset(exp95_IgG_lgcl, Dump)
exp95_IgG_live <- as.numeric(fsApply(exp95_IgG_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.IgG.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_IgG_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_IgG_lgcl <- Subset(exp98_IgG_lgcl, exp98_Dump)
exp98_IgG_live <- as.numeric(fsApply(exp98_IgG_lgcl, nrow, use.exprs = TRUE))

### NaCl ###

pdf(file = "£LIVEDIR/Live.NaCl.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_NaCl_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_NaCl_lgcl <- Subset(exp93_NaCl_lgcl, Dump)
exp93_NaCl_live <- as.numeric(fsApply(exp93_NaCl_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.NaCl.exp95.pdf")
```

```
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_NaCl_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_NaCl_lgcl <- Subset(exp95_NaCl_lgcl, Dump)
exp95_NaCl_live <- as.numeric(fsApply(exp95_NaCl_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.NaCl.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_NaCl_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_NaCl_lgcl <- Subset(exp98_NaCl_lgcl, exp98_Dump)
exp98_NaCl_live <- as.numeric(fsApply(exp98_NaCl_lgcl, nrow, use.exprs = TRUE))

### TGF ###

TGF <- rectangleGate("PerCP.Cy5.5.A" = c(-Inf, 1.65), filterId = "TGF")
pdf(file = "£LIVEDIR/Live.TGF.exp93.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp93_TGF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = TGF, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp93_TGF_lgcl <- Subset(exp93_TGF_lgcl, TGF)
exp93_TGF_live <- as.numeric(fsApply(exp93_TGF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.TGF.exp95.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp95_TGF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = TGF, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp95_TGF_lgcl <- Subset(exp95_TGF_lgcl, TGF)
exp95_TGF_live <- as.numeric(fsApply(exp95_TGF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£LIVEDIR/Live.TGF.exp98.pdf")
xyplot(\`PerCP.Cy5.5.A\`~\`FSC.A\`, exp98_TGF_lgcl[[2]], smooth = FALSE, pos = c(0.9, 0.9), filter = exp98_Dump, stats = TRUE, xlim =
↪  c(0,266144))
dev.off()
exp98_TGF_lgcl <- Subset(exp98_TGF_lgcl, exp98_Dump)
exp98_TGF_live <- as.numeric(fsApply(exp98_TGF_lgcl, nrow, use.exprs = TRUE))

####################
## Singlet Gating ##
####################

setwd("£SINGLEDIR")

sg <- matrix(c(50000, 150000, 220000, 200000, 100000, 50000,
               23000, 90000, 160000, 170000, 100000, 60000), nrow = 6)
colnames(sg) <- c("FSC.A", "FSC.H")
singlet1 <- polygonGate(sg)

sg <- matrix(c(50000, 150000, 220000, 200000, 100000, 50000,
               23000, 90000, 160000, 170000, 100000, 60000), nrow = 6)
colnames(sg) <- c("FSC.A", "FSC.H")
singlet2 <- polygonGate(sg)

sg <- matrix(c(40000, 150000, 220000, 200000, 100000, 40000,
               23000, 100000, 160000, 170000, 100000, 60000), nrow = 6)
colnames(sg) <- c("FSC.A", "FSC.H")
singlet3 <- polygonGate(sg)

### AB Teflon ###

pdf(file = "£SINGLEDIR/Single.ABTeflon.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp93_ABT_lgcl[[12]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_ABT_lgcl <- Subset(exp93_ABT_lgcl, singlet1)
exp93_ABT_single <- as.numeric(fsApply(exp93_ABT_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.ABTeflon.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp95_ABT_lgcl[[12]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_ABT_lgcl <- Subset(exp95_ABT_lgcl, singlet2)
exp95_ABT_single <- as.numeric(fsApply(exp95_ABT_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.ABTeflon.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp98_ABT_lgcl[[12]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_ABT_lgcl <- Subset(exp98_ABT_lgcl, singlet3)
exp98_ABT_single <- as.numeric(fsApply(exp98_ABT_lgcl, nrow, use.exprs = TRUE))

### AB Plastic ###
```

```
pdf(file = "£SINGLEDIR/Single.ABTcPlastic.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp93_AB_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp93_AB_lgcl <- Subset(exp93_AB_lgcl, singlet1)
exp93_AB_single <- as.numeric(fsApply(exp93_AB_lgcl, nrow, use.exprs = TRUE))

sg <- matrix(c(50000, 150000, 220000, 200000, 100000, 10000,
               23000, 100000, 160000, 170000, 100000, 23000), nrow = 6)
colnames(sg) <- c("FSC.A", "FSC.H")
singlet4 <- polygonGate(sg)

pdf(file = "£SINGLEDIR/Single.ABTcPlastic.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp95_AB_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet4, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp95_AB_lgcl <- Subset(exp95_AB_lgcl, singlet4)
exp95_AB_single <- as.numeric(fsApply(exp95_AB_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.ABTcPlastic.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp98_AB_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim = c(0,266144),
↪  ylim = c(0,266144))
dev.off()
exp98_AB_lgcl <- Subset(exp98_AB_lgcl, singlet3)
exp98_AB_single <- as.numeric(fsApply(exp98_AB_lgcl, nrow, use.exprs = TRUE))

### Collagen I ###

pdf(file = "£SINGLEDIR/Single.CollagenI.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp93_ColI_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_ColI_lgcl <- Subset(exp93_ColI_lgcl, singlet1)
exp93_ColI_single <- as.numeric(fsApply(exp93_ColI_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.CollagenI.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp95_ColI_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_ColI_lgcl <- Subset(exp95_ColI_lgcl, singlet2)
exp95_ColI_single <- as.numeric(fsApply(exp95_ColI_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.CollagenI.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp98_ColI_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_ColI_lgcl <- Subset(exp98_ColI_lgcl, singlet3)
exp98_ColI_single <- as.numeric(fsApply(exp98_ColI_lgcl, nrow, use.exprs = TRUE))

### Collagen IV ###

pdf(file = "£SINGLEDIR/Single.CollagenIV.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp93_ColIV_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_ColIV_lgcl <- Subset(exp93_ColIV_lgcl, singlet1)
exp93_ColIV_single <- as.numeric(fsApply(exp93_ColIV_lgcl, nrow, use.exprs = TRUE))

sg <- matrix(c(50000, 150000, 220000, 220000, 100000, 50000,
               20000, 80000, 150000, 170000, 100000, 50000), nrow = 6)
colnames(sg) <- c("FSC.A", "FSC.H")
singlet5 <- polygonGate(sg)

pdf(file = "£SINGLEDIR/Single.CollagenIV.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp95_ColIV_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet5, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_ColIV_lgcl <- Subset(exp95_ColIV_lgcl, singlet5)
exp95_ColIV_single <- as.numeric(fsApply(exp95_ColIV_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.CollagenIV.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp98_ColIV_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_ColIV_lgcl <- Subset(exp98_ColIV_lgcl, singlet3)
exp98_ColIV_single <- as.numeric(fsApply(exp98_ColIV_lgcl, nrow, use.exprs = TRUE))

### Fibronectin ###

pdf(file = "£SINGLEDIR/Single.Fibronectin.exp93.pdf", width = 5, height = 5)
```

```
xyplot(\`FSC.H\``~\`FSC.A\`, exp93_Fib_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_Fib_lgcl <- Subset(exp93_Fib_lgcl, singlet1)
exp93_Fib_single <- as.numeric(fsApply(exp93_Fib_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Fibronectin.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp95_Fib_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_Fib_lgcl <- Subset(exp95_Fib_lgcl, singlet2)
exp95_Fib_single <- as.numeric(fsApply(exp95_Fib_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Fibronectin.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp98_Fib_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_Fib_lgcl <- Subset(exp98_Fib_lgcl, singlet3)
exp98_Fib_single <- as.numeric(fsApply(exp98_Fib_lgcl, nrow, use.exprs = TRUE))

### Laminin ###

pdf(file = "£SINGLEDIR/Single.Laminin.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp93_Lam_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_Lam_lgcl <- Subset(exp93_Lam_lgcl, singlet1)
exp93_Lam_single <- as.numeric(fsApply(exp93_Lam_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Laminin.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp95_Lam_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet4, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_Lam_lgcl <- Subset(exp95_Lam_lgcl, singlet4)
exp95_Lam_single <- as.numeric(fsApply(exp95_Lam_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Laminin.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp98_Lam_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_Lam_lgcl <- Subset(exp98_Lam_lgcl, singlet3)
exp98_Lam_single <- as.numeric(fsApply(exp98_Lam_lgcl, nrow, use.exprs = TRUE))

### Non-Adherent ###

pdf(file = "£SINGLEDIR/Single.NonAdherent.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp93_Non_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_Non_lgcl <- Subset(exp93_Non_lgcl, singlet1)
exp93_Non_single <- as.numeric(fsApply(exp93_Non_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.NonAdherent.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp95_Non_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet4, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_Non_lgcl <- Subset(exp95_Non_lgcl, singlet4)
exp95_Non_single <- as.numeric(fsApply(exp95_Non_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.NonAdherent.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp98_Non_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_Non_lgcl <- Subset(exp98_Non_lgcl, singlet3)
exp98_Non_single <- as.numeric(fsApply(exp98_Non_lgcl, nrow, use.exprs = TRUE))

### iDC ###

pdf(file = "£SINGLEDIR/Single.iDC.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp93_iDC_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_iDC_lgcl <- Subset(exp93_iDC_lgcl, singlet1)
exp93_iDC_single <- as.numeric(fsApply(exp93_iDC_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.iDC.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp95_iDC_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_iDC_lgcl <- Subset(exp95_iDC_lgcl, singlet2)
exp95_iDC_single <- as.numeric(fsApply(exp95_iDC_lgcl, nrow, use.exprs = TRUE))
```

```
pdf(file = "£SINGLEDIR/Single.iDC.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp98_iDC_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_iDC_lgcl <- Subset(exp98_iDC_lgcl, singlet3)
exp98_iDC_single <- as.numeric(fsApply(exp98_iDC_lgcl, nrow, use.exprs = TRUE))

### GMCSF ###

pdf(file = "£SINGLEDIR/Single.GMCSF.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp93_GMCSF_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_GMCSF_lgcl <- Subset(exp93_GMCSF_lgcl, singlet1)
exp93_GMCSF_single <- as.numeric(fsApply(exp93_GMCSF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.GMCSF.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp95_GMCSF_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet4, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_GMCSF_lgcl <- Subset(exp95_GMCSF_lgcl, singlet4)
exp95_GMCSF_single <- as.numeric(fsApply(exp95_GMCSF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.GMCSF.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp98_GMCSF_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_GMCSF_lgcl <- Subset(exp98_GMCSF_lgcl, singlet3)
exp98_GMCSF_single <- as.numeric(fsApply(exp98_GMCSF_lgcl, nrow, use.exprs = TRUE))

### MCSF ###

pdf(file = "£SINGLEDIR/Single.MCSF.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp93_MCSF_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_MCSF_lgcl <- Subset(exp93_MCSF_lgcl, singlet1)
exp93_MCSF_single <- as.numeric(fsApply(exp93_MCSF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.MCSF.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp95_MCSF_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_MCSF_lgcl <- Subset(exp95_MCSF_lgcl, singlet2)
exp95_MCSF_single <- as.numeric(fsApply(exp95_MCSF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.MCSF.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp98_MCSF_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_MCSF_lgcl <- Subset(exp98_MCSF_lgcl, singlet3)
exp98_MCSF_single <- as.numeric(fsApply(exp98_MCSF_lgcl, nrow, use.exprs = TRUE))

### Adiponectin ###

pdf(file = "£SINGLEDIR/Single.Adiponectin.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp93_Adipo_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_Adipo_lgcl <- Subset(exp93_Adipo_lgcl, singlet1)
exp93_Adipo_single <- as.numeric(fsApply(exp93_Adipo_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Adiponectin.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp95_Adipo_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_Adipo_lgcl <- Subset(exp95_Adipo_lgcl, singlet2)
exp95_Adipo_single <- as.numeric(fsApply(exp95_Adipo_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Adiponectin.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp98_Adipo_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_Adipo_lgcl <- Subset(exp98_Adipo_lgcl, singlet3)
exp98_Adipo_single <- as.numeric(fsApply(exp98_Adipo_lgcl, nrow, use.exprs = TRUE))

### Serotonin ###

pdf(file = "£SINGLEDIR/Single.Serotonin.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\`~\`FSC.A\`, exp93_Sero_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
```

```
exp93_Sero_lgcl <- Subset(exp93_Sero_lgcl, singlet1)
exp93_Sero_single <- as.numeric(fsApply(exp93_Sero_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Serotonin.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp95_Sero_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_Sero_lgcl <- Subset(exp95_Sero_lgcl, singlet2)
exp95_Sero_single <- as.numeric(fsApply(exp95_Sero_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Serotonin.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp98_Sero_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_Sero_lgcl <- Subset(exp98_Sero_lgcl, singlet3)
exp98_Sero_single <- as.numeric(fsApply(exp98_Sero_lgcl, nrow, use.exprs = TRUE))

### Hermin ###

pdf(file = "£SINGLEDIR/Single.Hermin.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp93_Hermin_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_Hermin_lgcl <- Subset(exp93_Hermin_lgcl, singlet1)
exp93_Hermin_single <- as.numeric(fsApply(exp93_Hermin_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Hermin.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp95_Hermin_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_Hermin_lgcl <- Subset(exp95_Hermin_lgcl, singlet2)
exp95_Hermin_single <- as.numeric(fsApply(exp95_Hermin_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.Hermin.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp98_Hermin_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_Hermin_lgcl <- Subset(exp98_Hermin_lgcl, singlet3)
exp98_Hermin_single <- as.numeric(fsApply(exp98_Hermin_lgcl, nrow, use.exprs = TRUE))

### IgG ###

pdf(file = "£SINGLEDIR/Single.IgG.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp93_IgG_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_IgG_lgcl <- Subset(exp93_IgG_lgcl, singlet1)
exp93_IgG_single <- as.numeric(fsApply(exp93_IgG_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.IgG.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp95_IgG_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_IgG_lgcl <- Subset(exp95_IgG_lgcl, singlet2)
exp95_IgG_single <- as.numeric(fsApply(exp95_IgG_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.IgG.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp98_IgG_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_IgG_lgcl <- Subset(exp98_IgG_lgcl, singlet3)
exp98_IgG_single <- as.numeric(fsApply(exp98_IgG_lgcl, nrow, use.exprs = TRUE))

### NaCl ###

pdf(file = "£SINGLEDIR/Single.NaCl.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp93_NaCl_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_NaCl_lgcl <- Subset(exp93_NaCl_lgcl, singlet1)
exp93_NaCl_single <- as.numeric(fsApply(exp93_NaCl_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.NaCl.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp95_NaCl_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_NaCl_lgcl <- Subset(exp95_NaCl_lgcl, singlet2)
exp95_NaCl_single <- as.numeric(fsApply(exp95_NaCl_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.NaCl.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp98_NaCl_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
```

```
dev.off()
exp98_NaCl_lgcl <- Subset(exp98_NaCl_lgcl, singlet3)
exp98_NaCl_single <- as.numeric(fsApply(exp98_NaCl_lgcl, nrow, use.exprs = TRUE))

### TGF ###

pdf(file = "£SINGLEDIR/Single.TGF.exp93.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp93_TGF_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet1, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp93_TGF_lgcl <- Subset(exp93_TGF_lgcl, singlet1)
exp93_TGF_single <- as.numeric(fsApply(exp93_TGF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.TGF.exp95.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp95_TGF_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet2, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp95_TGF_lgcl <- Subset(exp95_TGF_lgcl, singlet2)
exp95_TGF_single <- as.numeric(fsApply(exp95_TGF_lgcl, nrow, use.exprs = TRUE))

pdf(file = "£SINGLEDIR/Single.TGF.exp98.pdf", width = 5, height = 5)
xyplot(\`FSC.H\``~\`FSC.A\`, exp98_TGF_lgcl[[2]], smooth = FALSE, pos = c(0.1, 0.9), filter = singlet3, stats = TRUE, xlim =
↪  c(0,266144), ylim = c(0,266144))
dev.off()
exp98_TGF_lgcl <- Subset(exp98_TGF_lgcl, singlet3)
exp98_TGF_single <- as.numeric(fsApply(exp98_TGF_lgcl, nrow, use.exprs = TRUE))

##############################################################
### Saving corrected Data for each Sample in a Dataframe ###
##############################################################

setwd("£OUTDIR")

### AB Teflon ###

Conds <- c("exp93_ABT_lgcl", "exp95_ABT_lgcl", "exp98_ABT_lgcl")

for(j in Conds){
  for(x in seq(1:12)){
    m <- paste(j,x,sep="_")
    d <- get(j)
    d <- as.data.frame(exprs(d[[x]]))
    assign(m, d)
  }
}

### AB Plastic ###

Conds <- c("exp93_AB_lgcl", "exp95_AB_lgcl", "exp98_AB_lgcl", "exp93_ColI_lgcl", "exp95_ColI_lgcl", "exp98_ColI_lgcl",
↪  "exp93_ColIV_lgcl", "exp95_ColIV_lgcl", "exp98_ColIV_lgcl", "exp93_Fib_lgcl", "exp95_Fib_lgcl", "exp98_Fib_lgcl", "exp93_Lam_lgcl",
↪  "exp95_Lam_lgcl", "exp98_Lam_lgcl", "exp93_Non_lgcl", "exp95_Non_lgcl", "exp98_Non_lgcl", "exp93_iDC_lgcl", "exp95_iDC_lgcl",
↪  "exp98_iDC_lgcl", "exp93_GMCSF_lgcl", "exp95_GMCSF_lgcl", "exp98_GMCSF_lgcl", "exp93_MCSF_lgcl", "exp95_MCSF_lgcl",
↪  "exp98_MCSF_lgcl", "exp93_Adipo_lgcl", "exp95_Adipo_lgcl", "exp98_Adipo_lgcl", "exp93_Sero_lgcl", "exp95_Sero_lgcl",
↪  "exp98_Sero_lgcl", "exp93_IgG_lgcl", "exp95_IgG_lgcl", "exp98_IgG_lgcl", "exp93_NaCl_lgcl", "exp95_NaCl_lgcl", "exp98_NaCl_lgcl",
↪  "exp93_TGF_lgcl", "exp95_TGF_lgcl", "exp98_TGF_lgcl", "exp93_Hermin_lgcl")

#Unstained Cells
for(j in Conds){
  m <- paste(j,"unstained",sep="_")
  d <- get(j)
  d <- as.data.frame(exprs(d[[1]]))
  assign(m, d)
}

#Panel Cells
for(i in Conds){
  m <- paste(i,"panel",sep="_")
  d <- get(i)
  d <- as.data.frame(exprs(d[[2]]))
  assign(m, d)
}

### Hermin ###

Conds <- c("exp95_Hermin_lgcl", "exp98_Hermin_lgcl")

for(j in Conds){
  for(x in seq(1:5)){
    m <- paste(j,x,sep="_")
    d <- get(j)
    d <- as.data.frame(exprs(d[[x]]))
    assign(m, d)
  }
}


##################################################################
### Calculating Median Fluorescence Intensity for all Samples ###
##################################################################

### AB Teflon ###

exp93_ABT_lgcl_df <- data.frame(cbind(median(exp93_ABT_lgcl_12\£FITC.A) - median(exp93_ABT_lgcl_5\£FITC.A),
↪  median(exp93_ABT_lgcl_12\£BV711.A) - median(exp93_ABT_lgcl_10\£BV711.A), median(exp93_ABT_lgcl_12\£PE.A) -
↪  median(exp93_ABT_lgcl_7\£PE.A), median(exp93_ABT_lgcl_12\£PE.Cy7.A) - median(exp93_ABT_lgcl_11\£PE.Cy7.A),
↪  median(exp93_ABT_lgcl_12\£APC.A) - median(exp93_ABT_lgcl_4\£APC.A), median(exp93_ABT_lgcl_12\£APC.H7.A) -
↪  median(exp93_ABT_lgcl_8\£APC.H7.A), median(exp93_ABT_lgcl_12\£BV421.A) - median(exp93_ABT_lgcl_3\£BV421.A),
↪  median(exp93_ABT_lgcl_12\£BV510.A) - median(exp93_ABT_lgcl_6\£BV510.A), median(exp93_ABT_lgcl_12\£BV650.A) -
↪  median(exp93_ABT_lgcl_9\£BV650.A)))
colnames(exp93_ABT_lgcl_df) <- c("CD15", "CD14", "CD16", "HLADR", "CD11b", "CD71", "CD115", "CD64", "CD163")
rownames(exp93_ABT_lgcl_df) <- c("exp93_AB_Teflon_df")

exp95_ABT_lgcl_df <- data.frame(cbind(median(exp95_ABT_lgcl_12\£FITC.A) - median(exp95_ABT_lgcl_5\£FITC.A),
↪  median(exp95_ABT_lgcl_12\£BV711.A) - median(exp95_ABT_lgcl_10\£BV711.A), median(exp95_ABT_lgcl_12\£PE.A) -
↪  median(exp95_ABT_lgcl_7\£PE.A), median(exp95_ABT_lgcl_12\£PE.Cy7.A) - median(exp95_ABT_lgcl_11\£PE.Cy7.A),
↪  median(exp95_ABT_lgcl_12\£APC.A) - median(exp95_ABT_lgcl_4\£APC.A), median(exp95_ABT_lgcl_12\£APC.H7.A) -
↪  median(exp95_ABT_lgcl_8\£APC.H7.A), median(exp95_ABT_lgcl_12\£BV421.A) - median(exp95_ABT_lgcl_3\£BV421.A),
↪  median(exp95_ABT_lgcl_12\£BV510.A) - median(exp95_ABT_lgcl_6\£BV510.A), median(exp95_ABT_lgcl_12\£BV650.A) -
↪  median(exp95_ABT_lgcl_9\£BV650.A)))
colnames(exp95_ABT_lgcl_df) <- c("CD15", "CD14", "CD16", "HLADR", "CD11b", "CD71", "CD115", "CD64", "CD163")
rownames(exp95_ABT_lgcl_df) <- c("exp95_AB_Teflon_df")

exp98_ABT_lgcl_df <- data.frame(cbind(median(exp98_ABT_lgcl_12\£FITC.A) - median(exp98_ABT_lgcl_5\£FITC.A),
↪  median(exp98_ABT_lgcl_12\£BV711.A) - median(exp98_ABT_lgcl_10\£BV711.A), median(exp98_ABT_lgcl_12\£PE.A) -
↪  median(exp98_ABT_lgcl_7\£PE.A), median(exp98_ABT_lgcl_12\£PE.Cy7.A) - median(exp98_ABT_lgcl_11\£PE.Cy7.A),
↪  median(exp98_ABT_lgcl_12\£APC.A) - median(exp98_ABT_lgcl_4\£APC.A), median(exp98_ABT_lgcl_12\£APC.H7.A) -
↪  median(exp98_ABT_lgcl_8\£APC.H7.A), median(exp98_ABT_lgcl_12\£BV421.A) - median(exp98_ABT_lgcl_3\£BV421.A),
↪  median(exp98_ABT_lgcl_12\£BV510.A) - median(exp98_ABT_lgcl_6\£BV510.A), median(exp98_ABT_lgcl_12\£BV650.A) -
↪  median(exp98_ABT_lgcl_9\£BV650.A)))
colnames(exp98_ABT_lgcl_df) <- c("CD15", "CD14", "CD16", "HLADR", "CD11b", "CD71", "CD115", "CD64", "CD163")
rownames(exp98_ABT_lgcl_df) <- c("exp98_AB_Teflon_df")

### AB Plastic ###

exp93 <- c("exp93_AB_lgcl","exp93_ColI_lgcl", "exp93_ColIV_lgcl","exp93_Fib_lgcl", "exp93_Lam_lgcl","exp93_Non_lgcl","exp93_iDC_lgcl",
↪  "exp93_GMCSF_lgcl", "exp93_MCSF_lgcl", "exp93_Adipo_lgcl","exp93_Sero_lgcl", "exp93_IgG_lgcl","exp93_NaCl_lgcl","exp93_TGF_lgcl",
↪  "exp93_Hermin_lgcl")
exp95 <- c("exp95_AB_lgcl", "exp95_ColI_lgcl", "exp95_ColIV_lgcl", "exp95_Fib_lgcl","exp95_Lam_lgcl",  "exp95_Non_lgcl",
↪  "exp95_iDC_lgcl","exp95_GMCSF_lgcl", "exp95_MCSF_lgcl","exp95_Adipo_lgcl", "exp95_Sero_lgcl","exp95_IgG_lgcl","exp95_NaCl_lgcl",
↪  "exp95_TGF_lgcl")
exp98 <- c( "exp98_AB_lgcl","exp98_ColI_lgcl",  "exp98_ColIV_lgcl",  "exp98_Fib_lgcl",  "exp98_Lam_lgcl",  "exp98_Non_lgcl",
↪  "exp98_iDC_lgcl", "exp98_GMCSF_lgcl", "exp98_MCSF_lgcl",   "exp98_Adipo_lgcl", "exp98_Sero_lgcl", "exp98_IgG_lgcl",
↪  "exp98_NaCl_lgcl", "exp98_TGF_lgcl")

for(i in exp93){
  n <- paste(i,"panel",sep="_")
  m <- paste(i,"df",sep="_")
  f <- get(n)
  d <- data.frame(cbind(median(f\£FITC.A) - median(exp93_ABT_lgcl_5\£FITC.A), median(f\£BV711.A) - median(exp93_ABT_lgcl_10\£BV711.A),
↪  median(f\£PE.A) - median(exp93_ABT_lgcl_7\£PE.A), median(f\£PE.Cy7.A) - median(exp93_ABT_lgcl_11\£PE.Cy7.A), median(f\£APC.A) -
↪  median(exp93_ABT_lgcl_4\£APC.A), median(f\£APC.H7.A) - median(exp93_ABT_lgcl_8\£APC.H7.A), median(f\£BV421.A) -
↪  median(exp93_ABT_lgcl_3\£BV421.A), median(f\£BV510.A) - median(exp93_ABT_lgcl_6\£BV510.A), median(f\£BV650.A) -
↪  median(exp93_ABT_lgcl_9\£BV650.A)))
  colnames(d) <- c("CD15", "CD14", "CD16", "HLADR", "CD11b", "CD71", "CD115", "CD64", "CD163")
  rownames(d) <- m
  assign(m, d)
}

for(i in exp95){
  n <- paste(i,"panel",sep="_")
  m <- paste(i,"df",sep="_")
  f <- get(n)
  d <- data.frame(cbind(median(f\£FITC.A) - median(exp95_ABT_lgcl_5\£FITC.A), median(f\£BV711.A) - median(exp95_ABT_lgcl_10\£BV711.A),
↪  median(f\£PE.A) - median(exp95_ABT_lgcl_7\£PE.A), median(f\£PE.Cy7.A) - median(exp95_ABT_lgcl_11\£PE.Cy7.A), median(f\£APC.A) -
↪  median(exp95_ABT_lgcl_4\£APC.A), median(f\£APC.H7.A) - median(exp95_ABT_lgcl_8\£APC.H7.A), median(f\£BV421.A) -
↪  median(exp95_ABT_lgcl_3\£BV421.A), median(f\£BV510.A) - median(exp95_ABT_lgcl_6\£BV510.A), median(f\£BV650.A) -
↪  median(exp95_ABT_lgcl_9\£BV650.A)))
  colnames(d) <- c("CD15", "CD14", "CD16", "HLADR", "CD11b", "CD71", "CD115", "CD64", "CD163")
  rownames(d) <- m
  assign(m, d)
}

for(i in exp98){
  n <- paste(i,"panel",sep="_")
  m <- paste(i,"df",sep="_")
  f <- get(n)
  d <- data.frame(cbind(median(f\£FITC.A) - median(exp98_ABT_lgcl_5\£FITC.A), median(f\£BV711.A) - median(exp98_ABT_lgcl_10\£BV711.A),
↪  median(f\£PE.A) - median(exp98_ABT_lgcl_7\£PE.A), median(f\£PE.Cy7.A) - median(exp98_ABT_lgcl_11\£PE.Cy7.A), median(f\£APC.A) -
↪  median(exp93_ABT_lgcl_4\£APC.A), median(f\£APC.H7.A) - median(exp98_ABT_lgcl_8\£APC.H7.A), median(f\£BV421.A) -
↪  median(exp98_ABT_lgcl_3\£BV421.A), median(f\£BV510.A) - median(exp98_ABT_lgcl_6\£BV510.A), median(f\£BV650.A) -
↪  median(exp98_ABT_lgcl_9\£BV650.A)))
  colnames(d) <- c("CD15", "CD14", "CD16", "HLADR", "CD11b", "CD71", "CD115", "CD64", "CD163")
  rownames(d) <- m
  assign(m, d)
}

### Hermin ###

exp95_Hermin_lgcl_df <- data.frame(cbind(median(exp95_Hermin_lgcl_5\£FITC.A) - median(exp95_Hermin_lgcl_2\£FITC.A),
↪  median(exp95_Hermin_lgcl_5\£BV711.A) - median(exp95_ABT_lgcl_10\£BV711.A), median(exp95_Hermin_lgcl_5\£PE.A) -
↪  median(exp95_Hermin_lgcl_3\£PE.A), median(exp95_Hermin_lgcl_5\£PE.Cy7.A) - median(exp95_ABT_lgcl_11\£PE.Cy7.A),
↪  median(exp95_Hermin_lgcl_5\£APC.A) - median(exp95_ABT_lgcl_4\£APC.A), median(exp95_Hermin_lgcl_5\£APC.H7.A) -
↪  median(exp95_ABT_lgcl_8\£APC.H7.A), median(exp95_Hermin_lgcl_5\£BV421.A) - median(exp95_ABT_lgcl_3\£BV421.A),
↪  median(exp95_Hermin_lgcl_5\£BV510.A) - median(exp95_Hermin_lgcl_4\£BV510.A), median(exp95_Hermin_lgcl_5\£BV650.A) -
↪  median(exp95_ABT_lgcl_9\£BV650.A)))
colnames(exp95_Hermin_lgcl_df) <- c("CD15", "CD14", "CD16", "HLADR", "CD11b", "CD71", "CD115", "CD64", "CD163")
rownames(exp95_Hermin_lgcl_df) <- c("exp95_Hermin_df")

exp98_Hermin_lgcl_df <- data.frame(cbind(median(exp98_Hermin_lgcl_5\£FITC.A) - median(exp98_Hermin_lgcl_2\£FITC.A),
↪  median(exp98_Hermin_lgcl_5\£BV711.A) - median(exp98_ABT_lgcl_10\£BV711.A), median(exp98_Hermin_lgcl_5\£PE.A) -
↪  median(exp98_Hermin_lgcl_3\£PE.A), median(exp98_Hermin_lgcl_5\£PE.Cy7.A) - median(exp98_ABT_lgcl_11\£PE.Cy7.A),
↪  median(exp98_Hermin_lgcl_5\£APC.A) - median(exp98_ABT_lgcl_4\£APC.A), median(exp98_Hermin_lgcl_5\£APC.H7.A) -
↪  median(exp98_ABT_lgcl_8\£APC.H7.A), median(exp98_Hermin_lgcl_5\£BV421.A) - median(exp98_ABT_lgcl_3\£BV421.A),
↪  median(exp98_Hermin_lgcl_5\£BV510.A) - median(exp98_Hermin_lgcl_4\£BV510.A), median(exp98_Hermin_lgcl_5\£BV650.A) -
↪  median(exp98_ABT_lgcl_9\£BV650.A)))
colnames(exp98_Hermin_lgcl_df) <- c("CD15", "CD14", "CD16", "HLADR", "CD11b", "CD71", "CD115", "CD64", "CD163")
rownames(exp98_Hermin_lgcl_df) <- c("exp98_Hermin_df")

ls()

MFI <- rbind(exp93_AB_lgcl_df, exp95_AB_lgcl_df, exp98_AB_lgcl_df, exp93_ABT_lgcl_df, exp95_ABT_lgcl_df, exp98_ABT_lgcl_df,
↪  exp93_Hermin_lgcl_df, exp95_Hermin_lgcl_df, exp98_Hermin_lgcl_df, exp93_ColI_lgcl_df, exp95_ColI_lgcl_df, exp98_ColI_lgcl_df,
↪  exp93_ColIV_lgcl_df, exp95_ColIV_lgcl_df, exp98_ColIV_lgcl_df, exp93_Fib_lgcl_df, exp95_Fib_lgcl_df, exp98_Fib_lgcl_df,
↪  exp93_Lam_lgcl_df, exp95_Lam_lgcl_df, exp98_Lam_lgcl_df, exp93_Non_lgcl_df, exp95_Non_lgcl_df, exp98_Non_lgcl_df,
↪  exp93_iDC_lgcl_df, exp95_iDC_lgcl_df, exp98_iDC_lgcl_df, exp93_GMCSF_lgcl_df, exp95_GMCSF_lgcl_df, exp98_GMCSF_lgcl_df,
↪  exp93_MCSF_lgcl_df, exp95_MCSF_lgcl_df, exp98_MCSF_lgcl_df, exp93_Adipo_lgcl_df, exp95_Adipo_lgcl_df, exp98_Adipo_lgcl_df,
↪  exp93_Sero_lgcl_df, exp95_Sero_lgcl_df, exp98_Sero_lgcl_df, exp93_IgG_lgcl_df, exp95_IgG_lgcl_df ,exp98_IgG_lgcl_df,
↪  exp93_NaCl_lgcl_df, exp95_NaCl_lgcl_df, exp98_NaCl_lgcl_df, exp93_TGF_lgcl_df, exp95_TGF_lgcl_df, exp98_TGF_lgcl_df)


write.table(MFI, file = "£OUTDIR/MFI_M1M2M3.txt", sep = "\t", col.names = NA)

EOF
chmod 750 ~/Documents/Doktorarbeit/FACS.R
R < ~/Documents/Doktorarbeit/FACS.R  --no-save
rm ~/Documents/Doktorarbeit/FACS.R

#################################################################################
### This R Script containes plots obtained with the MFI values calculated above ###
#################################################################################

MARKERS="CD15 CD14 CD16 CD11b CD64 CD71 CD115 CD163 HLADR"
for i in £{MARKERS[@]}
do
    cat > ~/Documents/Doktorarbeit/Plots_FACS.$i.R << EOF
library(ggplot2)
library(ggpubr)
library(RColorBrewer)
library(reshape2)

data <- read.delim(file = "£OUTDIR/MFI_M1M2M3.txt", sep = "\t", row.names = "X")
data\£Type <- c(rep("AB_Plastic", 3), rep("AB_Teflon", 3), rep("Hermin", 3), rep("Collagen_I", 3), rep("Collagen_IV", 3),
↪  rep("Fibronectin", 3), rep("Laminin", 3), rep("Non_Adherent", 3), rep("iDC", 3), rep("GMCSF", 3), rep("MCSF", 3),
↪  rep("Adiponectin", 3), rep("Serotonin", 3), rep("IgG", 3), rep("NaCl", 3), rep("TGF", 3))

colors_group <- c("£colABTcPlastic", "£colLam", "£colFib", "£colNONAD", "£colCol1", "£colCol4", "£colABTeflon", "£colAdipo",
↪  "£colSero", "£colTGF", "£colHermin", "£colNaCl", "£colIgG", "£colMCSF", "£colGMCSF", "£coliDC")

type <- c("AB_Plastic", "Laminin", "Fibronectin", "Non_Adherent", "Collagen_I", "Collagen_IV", "AB_Teflon", "Adiponectin", "Serotonin",
↪  "TGF", "Hermin", "NaCl", "IgG", "MCSF", "GMCSF", "iDC")

data\£Type <- factor(data\£Type, levels = type)

my_comparisons <- list(c("AB_Plastic", "Collagen_I"), c("AB_Plastic", "Collagen_IV"), c("AB_Teflon", "TGF"),c("AB_Teflon",
↪  "Serotonin"), c("AB_Teflon", "Hermin"))
```

```
p1 <- ggplot(data = data, aes(x = Type, y = £i)) + theme_bw() +
geom_point(aes(colour = Type), size = 2, show.legend = FALSE) +
scale_colour_manual(values = colors_group) +
ylim(-0.5,3)+
labs(x = "", y = "MFI") +
stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = "crossbar", width = 0.75, size = .1)+
scale_x_discrete(labels = c("AB/ MCSF", "Laminin", "Fibronectin", "Non_Adherent", "Collagen_I", "Collagen_IV", "AB-Teflon",
↪  "Adiponectin", "Serotonin", "TGF", "Heme", "NaCl", "IgG", "MCSF/ FCS", "GMCSF", "iDC"))+
stat_compare_means(comparisons = my_comparisons, method = "t.test", paired = TRUE, aes(label = ..p.signif..),label.y =
↪  c(2.6,2.9,2.3,2.6,2.9))+
theme(panel.grid.minor = element_blank(), panel.grid.major = element_blank(), axis.text.x = element_text(family = "ArialMT", colour =
↪  "black", size = 14, angle = 45, hjust = 1),
axis.text.y = element_text(family = "ArialMT", colour = "black", size = 14),
axis.title = element_text(family = "ArialMT", colour = "black", size = 16))

pdf(file = "£OUTDIR/Dotplot_£i.pdf", width = 6, height = 3.5)
plot(p1)
dev.off()


EOF
chmod 750 ~/Documents/Doktorarbeit/Plots_FACS.$i.R
R < ~/Documents/Doktorarbeit/Plots_FACS.$i.R  --no-save
rm ~/Documents/Doktorarbeit/Plots_FACS.$i.R
done
```

The following scripts were written for further analysis of all obtained data sets. The following scripts contain the RNA-seq analysis.

```
#!/usr/bin/env bash
#File: M1M3M6_AllSamples_diffExpGenes.sh by Claudia Kiesewetter on 01.10.2018
#Script for basic RNA-seq analysis using R and EdgeR for macrophages treated with different conditions.

#setting up environment variables

DIR_PKG="/misc/software/ngs"

PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin

export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH

ANALYSISDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6"
DATADIR="/misc/data/processedData/mapping/RNA/GRCh38/RNAseq/MOMACDC/"
IGVDIR="/misc/data/analysis/project_MAC_CK/IGV_Sessions"
BIGWIGDIR="/misc/data/processedData/bigWig/RNA/GRCh38/RNAseq/MOMACDC"
CHROMSIZES="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes/GRCh38.PRI_p10.chrom.sizes"


mkdir -p $ANALYSISDIR/tmp
mkdir -p $ANALYSISDIR/stat_files

STATDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6/stat_files"

### Setting colors for the different cell types ###

colCol1="deeppink"
colCol4="violetred3"
colABTcPlastic="slateblue3"
colLam="orchid1"
colMono="forestgreen"
colAdipo="khaki2"
colSero="lightsalmon2"
colTGF="darkgoldenrod3"
colHermin="brown4"
colNaCl="gold2"
colIgG="orange1"
colFib="plum3"
colMCSF="cornflowerblue"
coliDC="dodgerblue3"
colGMCSF="cadetblue3"
colNONAD="hotpink"
colABTeflon="wheat3"
```

```
### Creating a Gene Count Table from STAR output ###

#pasting all files together in the specified order
cd $DATADIR
echo "Start creating gene count table."

paste /misc/software/ngs/genome/sequence/GRCh38.PRI_p10/STAR_transcriptIDshort.txt RNAseq_CK_4_M1_Monocytes.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_21_M3_Monocytes.trimmed.ReadsPerGene.txt RNA-seq_CK_83_M6_Monocytes.ReadsPerGene.txt
↪  RNAseq_CK_11_M1_AB_Teflon.trimmed.ReadsPerGene.txt RNAseq_CK_22_M3_AB_Teflon.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_5_M1_Adiponectin.trimmed.ReadsPerGene.txt RNAseq_CK_25_M3_Adiponectin.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_6_M1_Serotonin.trimmed.ReadsPerGene.txt RNAseq_CK_26_M3_Serotonin.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_7_M1_TGF.trimmed.ReadsPerGene.txt RNAseq_CK_24_M3_TGF.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_8_M1_Hermin.trimmed.ReadsPerGene.txt RNAseq_CK_27_M3_Hermin.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_9_M1_NaCl.trimmed.ReadsPerGene.txt RNAseq_CK_23_M3_NaCl.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_10_M1_IgG.trimmed.ReadsPerGene.txt RNAseq_CK_28_M3_IgG.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_12_M1_Collagen_1.trimmed.ReadsPerGene.txt RNAseq_CK_33_M3_Collagen_1.trimmed.ReadsPerGene.txt
↪  RNA-seq_CK_85_M6_Collagen_1.ReadsPerGene.txt RNAseq_CK_13_M1_Collagen_4.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_34_M3_Collagen_4.trimmed.ReadsPerGene.txt RNA-seq_CK_86_M6_Collagen_4.ReadsPerGene.txt
↪  RNAseq_CK_14_M1_Fibronectin.trimmed.ReadsPerGene.txt RNAseq_CK_35_M3_Fibronectin.trimmed.ReadsPerGene.txt
↪  RNA-seq_CK_87_M6_Fibronectin.ReadsPerGene.txt RNAseq_CK_15_M1_Laminin.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_36_M3_Laminin.trimmed.ReadsPerGene.txt RNA-seq_CK_88_M6_Laminin.ReadsPerGene.txt
↪  RNAseq_CK_16_M1_MCSF.trimmed.ReadsPerGene.txt RNAseq_CK_30_M3_MCSF.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_17_M1_AB_TcPlastic.trimmed.ReadsPerGene.txt RNAseq_CK_29_M3_AB_TcPlastic.trimmed.ReadsPerGene.txt
↪  RNA-seq_CK_84_M6_AB_TcPlastic.ReadsPerGene.txt RNAseq_CK_18_M1_iDC.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_32_M3_iDC.trimmed.ReadsPerGene.txt RNAseq_CK_19_M1_GMCSF.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_31_M3_GMCSF.trimmed.ReadsPerGene.txt RNAseq_CK_20_M1_Rollerflaschen.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_37_M3_Rollerflaschen.trimmed.ReadsPerGene.txt > /loctmp/GeneCount_M1M3M6_AllSamples.txt

#Selecting the correct column of the gene count file per Sample
#M1 und M3: Truseq RNAseq kit --> Spalte 3
#M6: Scriptseq Kit --> Spalte 2
#Separated by samples to easily see if column selection is correct:

#cut -f 1,
#5,9,12,
#17,21,25,29,33,37,41,45,49,53,57,61,65,69,
#73,77,80,
#85,89,92,
#97,101,104,
#109,113,116,
#121,125,
#129,133,136,
#141,145,
#149,153,
#157,161 > /loctmp/GeneCount_M1M3M6_AllSamples.cut.txt

#Select columns:

cut -f
↪  1,5,9,12,17,21,25,29,33,37,41,45,49,53,57,61,65,69,73,77,80,85,89,92,97,101,104,109,113,116,121,125,129,133,136,141,145,149,153,157,161
↪  /loctmp/GeneCount_M1M3M6_AllSamples.txt > /loctmp/GeneCount_M1M3M6_AllSamples.cut.txt

tail -n +5 /loctmp/GeneCount_M1M3M6_AllSamples.cut.txt > /loctmp/GeneCount_M1M3M6_AllSamples.tail.txt

#Change header
echo
↪  £'Gene\tM1_Mono\tM3_Mono\tM6_Mono\tM1_AB_Teflon\tM3_AB_Teflon\tM1_Adiponectin\tM3_Adiponectin\tM1_Serotonin\tM3_Serotonin\tM1_TGF\tM3_TGF\tM1_Hermin\tM3_Herm
↪  | cat - /loctmp/GeneCount_M1M3M6_AllSamples.tail.txt > $ANALYSISDIR/GeneCount_M1M3M6_AllSamples_red.txt

echo "Finished Gene Count Table."

cd £{ANALYSISDIR}
_Date=$(date +%s)

## Create overlapping BigWig Files ##

#Monocytes

myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_4_M1_Monocytes.Unique.for.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_21_M3_Monocytes.Unique.for.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Mono.rev.bigwig
myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_4_M1_Monocytes.Unique.rev.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_21_M3_Monocytes.Unique.rev.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Mono.for.bigwig

#AB TcPlastic

myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_17_M1_AB_TcPlastic.Unique.for.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_29_M3_AB_TcPlastic.Unique.for.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_AB_rev.bigwig
myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_17_M1_AB_TcPlastic.Unique.rev.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_29_M3_AB_TcPlastic.Unique.rev.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_AB.for.bigwig

#Collagen I
```

```
myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_33_M3_Collagen_1.Unique.for.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_12_M1_Collagen_1.Unique.for.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Col1_rev.bigwig
myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_33_M3_Collagen_1.Unique.rev.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_12_M1_Collagen_1.Unique.rev.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Col1.for.bigwig

#Collagen IV

myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_34_M3_Collagen_4.Unique.for.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_13_M1_Collagen_4.Unique.for.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Col4_rev.bigwig
myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_34_M3_Collagen_4.Unique.rev.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_13_M1_Collagen_4.Unique.rev.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Col4.for.bigwig

#Fibronectin

myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_14_M1_Fibronectin.Unique.for.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_35_M3_Fibronectin.Unique.for.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Fibronectin_rev.bigwig
myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_14_M1_Fibronectin.Unique.rev.bigwig
↪  £{BIGWIGDIR}/RNAseq_CK_35_M3_Fibronectin.Unique.rev.bigwig -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Fibronectin.for.bigwig

#Laminin

myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_15_M1_Laminin.Unique.for.bigwig £{BIGWIGDIR}/RNAseq_CK_36_M3_Laminin.Unique.for.bigwig
↪  -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Laminin_rev.bigwig
myAverageBigWig.pl -bw £{BIGWIGDIR}/RNAseq_CK_15_M1_Laminin.Unique.rev.bigwig £{BIGWIGDIR}/RNAseq_CK_36_M3_Laminin.Unique.rev.bigwig
↪  -chr £{CHROMSIZES} -o £{IGVDIR}/RNAseq_Laminin.for.bigwig

### Starting analysis in R ###

cat > "/loctmp/R.edgeR.M1M3M6.£{_Date}.R" << EOF
library(edgeR)
library(sqldf)
library(extrafont)

data <- read.delim(file = "£{ANALYSISDIR}/GeneCount_M1M3M6_AllSamples_red.txt", sep = "\t", row.names = "Gene")

group <- factor(c(rep("Mono",3), rep("AB_Teflon", 2), rep("Adiponektin", 2), rep("Serotonin", 2), rep("TGF",2), rep("Hermin", 2),
↪  rep("NaCl", 2), rep("IgG", 2), rep("Collagen_I", 3), rep("Collagen_IV", 3), rep("Fibronectin", 3), rep("Laminin", 3), rep("MCSF",
↪  2), rep("AB_TcPlastic", 3), rep("iDC", 2), rep("GMCSF", 2), rep("Rollerflaschen", 2)))

#Creating the DGEList object for EdgeR
d <- DGEList(counts = data, group = group)
keep <- rowSums(cpm(d) > 1) >= 2
summary(keep)
d <- d[keep, keep.lib.sizes=FALSE]
d <- calcNormFactors(d) #normalisierung!
d\£samples

logcpm <- cpm(d, prior.count = 2, normalized.lib.sizes = TRUE, log = TRUE)

### First MDS Plot ###

group <- factor(c(rep("Mono",3), rep("AB_Teflon", 2), rep("Adiponektin", 2), rep("Serotonin", 2), rep("TGF",2), rep("Hermin", 2),
↪  rep("NaCl", 2), rep("IgG", 2), rep("Collagen_I",3), rep("Collagen_IV",3), rep("Fibronectin", 3), rep("Laminin",3), rep("MCSF", 2),
↪  rep("AB_TcPlastic", 3), rep("iDC", 2), rep("GMCSF", 2), rep("Rollerflaschen", 2)))

colors_group <- c(rep("£colMono", 3), rep("colABTeflon", 2), rep("colAdipo", 2), rep("colSero", 2), rep("colTGF", 2), rep("colHermin",
↪  2), rep("colNaCl", 2), rep("colIgG", 2), rep("colCol1", 3), rep("colCol4", 3), rep("colFib", 3), rep("colLam", 3), rep("colMCSF",
↪  2), rep("colABTcPlastic", 3), rep("coliDC", 2), rep("colGMCSF", 2), rep("colNONAD", 2))


Batch <- factor(c(0,4,7,2,4,0,4,0,4,0,4,0,0,0,4,2,0,2,6,7,2,6,7,2,6,7,3,6,7,3,0,3,0,7,3,0,3,0,3,6))
colors_batch <- rep(c("chartreuse3", "dodgerblue4", "chocolate", "cyan3", "plum2", "darkgoldenrod3", "magenta2", "firebrick3"))

pdf(file = "£{ANALYSISDIR}/tmp/MDSPlot_AllSamples.pdf")
plotMDS(logcpm, col = colors_group, cex = 1, main = "Coloured by Group")
plotMDS(logcpm, col = colors_batch[Batch], cex = 1, main = "Coloured by Library Preparation Batches")
dev.off()

### Remove Batch Effect ###

donor <- factor(c(1,2,4,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,4,1,2,4,1,2,4,1,2,4,1,2,1,2,4,1,2,1,2,1,2))

#Library preparation Batch:
#batch1, batch5 not included, as they produce error

Batch <- factor(c(0,4,7,2,4,0,4,0,4,0,4,0,0,0,4,2,0,2,6,7,2,6,7,2,6,7,3,6,7,3,0,3,0,7,3,0,3,0,3,6))
#batch 7 not estimable like this

Batch <- factor(c(0,4,0,2,4,0,4,0,4,0,4,0,0,0,4,2,0,2,6,7,2,6,7,2,6,7,3,6,7,3,0,3,0,7,3,0,3,0,3,6))


design1 <- model.matrix(~0 + group)
```

```
rownames(design1) <- colnames(d)
logcpm_rmBE <- removeBatchEffect(logcpm, design = design1, batch = donor, batch2 = Batch)

write.table(logcpm_rmBE, file = "£{ANALYSISDIR}/AllSamples_logcpm_rmBE.txt", sep = "\t", col.names = NA)

### Second MDS Plot ###

pdf(file = "£{ANALYSISDIR}/tmp/MDSPlot_AllSamples_rmBE.pdf")
plotMDS(logcpm_rmBE, col = colors_group, cex = 1, main = "Coloured by Group")
dev.off()

### Saving data for dotplots of selected genes ###

splnames <- unlist(strsplit(as.character(rownames(logcpm_rmBE)), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
rown <- data.frame(row.matrix)
data_box <- cbind(rown, logcpm_rmBE)
write.table(data_box, file = "M1M3M6_logcpm_rmBE_data_dotplots.txt", sep = "\t", col.names = NA, quote = FALSE)

##### Start Analysis for differential Gene expression #####

#Creating the design matrix

group <- factor(c(rep("Mono",3), rep("AB_Teflon", 2), rep("Adiponektin", 2), rep("Serotonin", 2), rep("TGF",2), rep("Hermin", 2),
↪  rep("NaCl", 2), rep("IgG", 2), rep("Collagen_I", 3), rep("Collagen_IV", 3), rep("Fibronectin", 3), rep("Laminin", 3), rep("MCSF",
↪  2), rep("AB_TcPlastic", 3), rep("iDC", 2), rep("GMCSF", 2), rep("Rollerflaschen", 2)))

#Donor batch:
donor <- factor(c(1,2,4,1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,4,1,2,4,1,2,4,1,2,4,1,2,1,2,4,1,2,1,2,1,2))
#Library preparation Batch:

Batch <- factor(c(0,4,7,2,4,0,4,0,4,0,4,0,0,0,4,2,0,2,6,7,2,6,7,2,6,7,3,6,7,3,0,3,0,7,3,0,3,0,3,6))
#batch 7 not estimable like this

Batch <- factor(c(0,4,0,2,4,0,4,0,4,0,4,0,0,0,4,2,0,2,6,7,2,6,7,2,6,7,3,6,7,3,0,3,0,7,3,0,3,0,3,6))

#resulting design
design <- model.matrix(˜0 + group + donor + Batch)
rownames(design) <- colnames(d)
design

#Cytokine Addition grouping
#plusMCSF: AB-TcPlastic, Laminin, Fibronektin, Rollerflaschen
#woMCSF: Adiponektin, NaCl, Serotonin, TGF, Hermin, IgG, AB-Teflon

group2 <- factor(c(rep("Mono", 3), rep("woMCSF", 14), rep("Collagen", 6), rep("plusMCSF", 6), rep("MCSF", 2), rep("plusMCSF", 3),
↪  rep("iDC", 2), rep("GMCSF", 2), rep("plusMCSF", 2)))

#resulting design
design2 <- model.matrix(˜0 + group2 + donor + Batch)
rownames(design2) <- colnames(d)
design2

### Calculate Dispersions ###
#only in vitro Mac, design1
d <- estimateDisp(d, design = design)
d\£common.dispersion

#design2; Cytokine addition groups
d2 <- estimateDisp(d, design = design2)
d2\£common.dispersion

#calculate generalized linar model
fit <- glmQLFit(d, design = design, robust = TRUE)
fit2 <- glmQLFit(d2, design = design2, robust = TRUE)

my.contrasts <- makeContrasts(ColIvsColIV = groupCollagen_I - groupCollagen_IV,
                              ColIvsAB = groupCollagen_I - groupAB_TcPlastic,
                              ColIVvsAB = groupCollagen_IV - groupAB_TcPlastic,
                              LamvsAB = groupLaminin - groupAB_TcPlastic,
                              FibvsAB = groupFibronectin - groupAB_TcPlastic,
                              FibvsColI = groupFibronectin - groupCollagen_I,
                              LamvsColI = groupLaminin - groupCollagen_I,
                              FibvsLam = groupFibronectin - groupLaminin,
                              iDCvsAB = groupiDC - groupAB_TcPlastic,
                              iDCvsG = groupiDC - groupGMCSF,
                              RollvsAB = groupRollerflaschen - groupAB_TcPlastic,
                              MvsAB = groupMCSF - groupAB_TcPlastic,
                              GvsAB = groupGMCSF - groupAB_TcPlastic,
                              MvsG = groupMCSF - groupGMCSF,
                              MvsiDC = groupMCSF - groupiDC,
                              ABvsAB = groupAB_Teflon - groupAB_TcPlastic,
                              HvsAB = groupHermin - groupAB_Teflon,
                              AdvsAB = groupAdiponektin - groupAB_Teflon,
                              IvsAB = groupIgG - groupAB_Teflon,
                              SvsAB = groupSerotonin - groupAB_Teflon,
                              NvsAB = groupNaCl - groupAB_Teflon,
                              TvsAB = groupTGF - groupAB_Teflon, levels = design)

my.contrasts_cytokines <- makeContrasts(ColvsCTRL = group2Collagen - group2plusMCSF,
                                        ColvsiDC = group2Collagen - group2iDC,
                                        ColvsMCSF = group2Collagen - group2MCSF,
                                        MonovsAll = group2Mono - ((group2Collagen + group2GMCSF + group2iDC + group2MCSF +
↪   group2plusMCSF + group2woMCSF)/7),
                                        woMCSFvsplusMCSF = group2woMCSF - group2plusMCSF,
                                        ColvswoMCSF = group2Collagen - group2woMCSF,
                                        iDCvsG = group2iDC - group2GMCSF,
                                        levels = design2)

#### Test for Differentially Expressed genes using defined Comparisons ####

lrt_ColvsCTRL <- glmQLFTest(fit2, contrast = my.contrasts_cytokines[, "ColvsCTRL"])
stat_ColvsCTRL <- topTags(lrt_ColvsCTRL, n = Inf)
write.table(stat_ColvsCTRL, file = "£{STATDIR}/AllSamples_stat_ColvsCTRL.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.ColvsCTRL <- decideTestsDGE(lrt_ColvsCTRL))

lrt_ColIvsAB <- glmQLFTest(fit, contrast = my.contrasts[, "ColIvsAB"])
stat_ColIvsAB <- topTags(lrt_ColIvsAB, n = Inf)
write.table(stat_ColIvsAB, file = "£{STATDIR}/AllSamples_stat_ColIvsAB.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.ColIvsAB <- decideTestsDGE(lrt_ColIvsAB))

lrt_ColIVvsAB <- glmQLFTest(fit, contrast = my.contrasts[, "ColIVvsAB"])
stat_ColIVvsAB <- topTags(lrt_ColIVvsAB, n = Inf)
write.table(stat_ColIVvsAB, file = "£{STATDIR}/AllSamples_stat_ColIVvsAB.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.ColIVvsAB <- decideTestsDGE(lrt_ColIVvsAB))

lrt_LamvsAB <- glmQLFTest(fit, contrast = my.contrasts[, "LamvsAB"])
stat_LamvsAB <- topTags(lrt_LamvsAB, n = Inf)
write.table(stat_LamvsAB, file = "£{STATDIR}/AllSamples_stat_LamvsAB.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.LamvsAB <- decideTestsDGE(lrt_LamvsAB))
#0

lrt_FibvsAB <- glmQLFTest(fit, contrast = my.contrasts[, "FibvsAB"])
stat_FibvsAB <- topTags(lrt_FibvsAB, n = Inf)
write.table(stat_FibvsAB, file = "£{STATDIR}/AllSamples_stat_FibvsAB.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.FibvsAB <- decideTestsDGE(lrt_FibvsAB))

lrt_FibvsColI <- glmQLFTest(fit, contrast = my.contrasts[, "FibvsColI"])
stat_FibvsColI <- topTags(lrt_FibvsColI, n = Inf)
write.table(stat_FibvsColI, file = "£{STATDIR}/AllSamples_stat_FibvsColI.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.FibvsColI <- decideTestsDGE(lrt_FibvsColI))

lrt_LamvsColI <- glmQLFTest(fit, contrast = my.contrasts[, "LamvsColI"])
stat_LamvsColI <- topTags(lrt_LamvsColI, n = Inf)
write.table(stat_LamvsColI, file = "£{STATDIR}/AllSamples_stat_LamvsColI.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.LamvsColI <- decideTestsDGE(lrt_LamvsColI))

lrt_FibvsLam <- glmQLFTest(fit, contrast = my.contrasts[, "FibvsLam"])
stat_FibvsLam <- topTags(lrt_FibvsLam, n = Inf)
write.table(stat_FibvsLam, file = "£{STATDIR}/AllSamples_stat_FibvsLam.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.FibvsLam <- decideTestsDGE(lrt_FibvsLam))

lrt_anova_all <- glmQLFTest(fit, contrast = my.contrasts)
stat_anova_all <- topTags(lrt_anova_all, n = Inf)
write.table(stat_anova_all, file = "£{STATDIR}/AllSamples_stat_anova.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.anovaall <- decideTestsDGE(lrt_anova_all))

lrt_anova_all2 <- glmQLFTest(fit2, contrast = my.contrasts_cytokines)
stat_anova_all2 <- topTags(lrt_anova_all2, n = Inf)
write.table(stat_anova_all2, file = "£{STATDIR}/AllSamples_stat_anova2.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.anovaall2 <- decideTestsDGE(lrt_anova_all2))

#### Add gene expression data ####

x1 <- read.delim(file = "£{ANALYSISDIR}/AllSamples_logcpm_rmBE.txt", sep = "\t")
colnames(x1) <- c("ID", colnames(logcpm_rmBE))

x2 <- read.delim("£{STATDIR}/AllSamples_stat_ColIvsAB.txt", sep = "\t")
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{STATDIR}/AllSamples_ColIvsAB_diffExp_rmBE.txt",sep = "\t", col.names = NA)

x2 <- read.delim("£{STATDIR}/AllSamples_stat_ColIVvsAB.txt", sep = "\t")
```

```
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{STATDIR}/AllSamples_ColIVvsAB_diffExp_rmBE.txt",sep = "\t", col.names = NA)

x2 <- read.delim("£{STATDIR}/AllSamples_stat_FibvsAB.txt", sep = "\t")
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{STATDIR}/AllSamples_FibvsAB_diffExp_rmBE.txt",sep = "\t", col.names = NA)

x2 <- read.delim("£{STATDIR}/AllSamples_stat_FibvsColI.txt", sep = "\t")
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{STATDIR}/AllSamples_FibvsColI_diffExp_rmBE.txt",sep = "\t", col.names = NA)

x2 <- read.delim("£{STATDIR}/AllSamples_stat_LamvsColI.txt", sep = "\t")
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{STATDIR}/AllSamples_LamvsColI_diffExp_rmBE.txt",sep = "\t", col.names = NA)

x2 <- read.delim("£{STATDIR}/AllSamples_stat_FibvsLam.txt", sep = "\t")
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{STATDIR}/AllSamples_FibvsLam_diffExp_rmBE.txt",sep = "\t", col.names = NA)

x2 <- read.delim("£{STATDIR}/AllSamples_stat_LamvsAB.txt", sep = "\t")
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{STATDIR}/AllSamples_LamvsAB_diffExp_rmBE.txt",sep = "\t", col.names = NA)

x2 <- read.delim("£{STATDIR}/AllSamples_stat_ColvsCTRL.txt", sep = "\t")
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{STATDIR}/AllSamples_ColvsCTRL_diffExp_rmBE.txt",sep = "\t", col.names = NA)

x2 <- read.delim("£{STATDIR}/AllSamples_stat_anova2.txt", sep = "\t")
colnames(x2) <- c("ID", colnames(stat_anova_all2))
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{STATDIR}/AllSamples_anova2_diffExp_rmBE.txt",sep = "\t", col.names = NA)

EOF
chmod 750 "/loctmp/R.edgeR.M1M3M6.£{_Date}.R"
R < /loctmp/R.edgeR.M1M3M6.£{_Date}.R --no-save
rm /loctmp/R.edgeR.M1M3M6.£{_Date}.R
```

```
#!/usr/bin/env bash
#File: HC_DiffExpAnalysis.sh by Claudia Kiesewetter on October 06, 2018
#This script was used to compare RNAseq data from hofbauer cells with collagen treated macrophages

#setting up environment variables

DIR_PKG="/misc/software/ngs"

PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin

export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH

ANALYSISDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/HofbauerCells"
HEATDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/HofbauerCells/Heatmaps"
DATADIR="/misc/data/processedData/mapping/RNA/GRCh38/RNAseq/MOMACDC"
COLDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6/Heatmaps"
TMPDIR="/loctmp"
DATADIRBMM="/misc/data/processedData/mapping/collaborations/Erlangen/Heiko/RNA/GRCh38/RNAseq/BMM"

mkdir -p $ANALYSISDIR/tmp

### Setting colors for the different cell types ###

colHC="darkorchid4"
colCol1="deeppink"
colCol4="violetred3"
colBMM="darkgoldenrod3"
colAB="slateblue3"
colLam="orchid1"
colMono="forestgreen"

### Creating a gene count table from STAR output ###
```

```
#pasting all files together in the specified order
cd $DATADIR
paste /misc/software/ngs/genome/sequence/GRCh38.PRI_p10/STAR_transcriptIDshort.txt RNA-seq_CK_HC_6.trimmed.ReadsPerGene.txt
↪  RNA-seq_CK_HC_7.trimmed.ReadsPerGene.txt RNA-seq_CK_HC_8.trimmed.ReadsPerGene.txt £{DATADIRBMM}/RNAseq_ER_CG_142.ReadsPerGene.txt
↪  £{DATADIRBMM}/RNAseq_ER_CG_143.ReadsPerGene.txt £{DATADIRBMM}/RNAseq_ER_CG_144.ReadsPerGene.txt
↪  £{DATADIRBMM}/RNAseq_ER_CG_145.ReadsPerGene.txt £{DATADIRBMM}/RNAseq_ER_CG_146.ReadsPerGene.txt
↪  £{DATADIRBMM}/RNAseq_ER_EM1A_CD14.trimmed.ReadsPerGene.txt £{DATADIRBMM}/RNAseq_GS_Spender4_CD14.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_4_M1_Monocytes.trimmed.ReadsPerGene.txt RNAseq_CK_21_M3_Monocytes.trimmed.ReadsPerGene.txt
↪  RNA-seq_CK_83_M6_Monocytes.ReadsPerGene.txt RNAseq_CK_12_M1_Collagen_1.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_33_M3_Collagen_1.trimmed.ReadsPerGene.txt RNA-seq_CK_85_M6_Collagen_1.ReadsPerGene.txt
↪  RNAseq_CK_13_M1_Collagen_4.trimmed.ReadsPerGene.txt RNAseq_CK_34_M3_Collagen_4.trimmed.ReadsPerGene.txt
↪  RNA-seq_CK_86_M6_Collagen_4.ReadsPerGene.txt RNAseq_CK_15_M1_Laminin.trimmed.ReadsPerGene.txt
↪  RNAseq_CK_36_M3_Laminin.trimmed.ReadsPerGene.txt RNA-seq_CK_88_M6_Laminin.ReadsPerGene.txt
↪  RNAseq_CK_17_M1_AB_TcPlastic.trimmed.ReadsPerGene.txt RNAseq_CK_29_M3_AB_TcPlastic.trimmed.ReadsPerGene.txt
↪  RNA-seq_CK_84_M6_AB_TcPlastic.ReadsPerGene.txt > £{TMPDIR}/GeneCount_HCBMMMonoColLamAB.txt

cut -f 1,4,8,12,16,20,24,28,32,36,40,45,49,52,57,61,64,69,73,76,81,85,88,93,97,100 £{TMPDIR}/GeneCount_HCBMMMonoColLamAB.txt >
↪  £{TMPDIR}/GeneCount_HCBMMMonoColLamAB.cut.txt

tail -n +5 £{TMPDIR}/GeneCount_HCBMMMonoColLamAB.cut.txt > £{TMPDIR}/GeneCount_HCBMMMonoColLamAB.tail.txt

echo
↪  £'ID\tHC_6\tHC_7\tHC_8\tBMM_142\tBMM_143\tBMM_144\tBMM_145\tBMM_146\tBMM_EM1A\tBMM_GS\tM1_Mono\tM3_Mono\tM6_Mono\tM1_Collagen_I\tM3_Collagen_I\tM6_Collagen_I\tM1
↪  | cat - £{TMPDIR}/GeneCount_HCBMMMonoColLamAB.tail.txt > £{ANALYSISDIR}/GeneCount_HCBMMMonoColLamAB_red.txt

mkdir -p $ANALYSISDIR/tmp
cd $ANALYSISDIR
_Date=$(date +%s)

### Starting R script for differnetial gene expression analysis ###

cat > "/loctmp/R.edgeR.HC.£{_Date}.R" << EOF
library(edgeR)
library(ggplot2)
library(ggrepel)
library(gplots)
library(RColorBrewer)
library(sqldf)
library(ggpubr)
library(Rtsne)

data <- read.delim(file = "£{ANALYSISDIR}/GeneCount_HCBMMMonoColLamAB_red.txt", sep = "\t", header = TRUE, row.names = "ID")

group <- factor(c(rep("HC", 3), rep("BMM", 7), rep("Mono", 3), rep("Collagen", 6), rep("Laminin", 3), rep("AB", 3)))
d <- DGEList(counts = data, group = group)
keep <- rowSums(cpm(d) > 1) >= 2
summary(keep)
d <- d[keep, keep.lib.sizes=FALSE]
d <- calcNormFactors(d) #normalisierung!
d\£samples

logcpm <- cpm(d, prior.count = 2, normalized.lib.sizes = TRUE, log = TRUE)

### MDS Plot 1 ###
colors_group <- c(rep("£colHC", 3), rep("£colBMM", 7), rep("£colMono", 3), rep("£colCol1", 6), rep("£colLam", 3), rep("£colAB", 3))

png(file = "£ANALYSISDIR/tmp/MDS_HCBMMColLamABMono.png", width = 750, height = 750)
plotMDS(logcpm, col = colors_group, cex = 1, main = "Coloured by Cell Type")
dev.off()

### Remove Batch Effect from BMM, Mono, Collagen and fibronectin ####

donor <- factor(c(0,0,0,0,5,5,5,5,0,0,1,2,0,1,2,3,1,2,3,1,2,3,1,2,3))

design1 <- model.matrix(~0 + group)
rownames(design1) <- colnames(d)
logcpm_rmBE <- removeBatchEffect(logcpm, design = design1, batch = donor)

write.table(logcpm_rmBE, file = "£ANALYSISDIR/HCBMMMonoColLamAB_logcpm_rmBE.txt", sep = "\t", col.names = NA)

### MDS plot 2 ###

png(file = "£ANALYSISDIR/tmp/MDS_HCBMMColLamABMono_rmBE.png", width = 600, height = 600)
plotMDS(logcpm_rmBE, col = colors_group, cex = 1, main = "Coloured by Group")
dev.off()

### Pearson Correlation Heatmap ###

clustercol <- colorRampPalette(c("dodgerblue4", "white", "firebrick3"))(299)

colors_group <- c(rep("£colHC", 3), rep("£colBMM", 7), rep("£colMono", 3), rep("£colCol1", 6), rep("£colLam", 3), rep("£colAB", 3))
```

```
z <- cor(logcpm_rmBE, use="all.obs", method="pearson")
hz <- hclust(dist(t(z), method = "manhattan"), method = "ward.D2")

pdf(file = "£HEATDIR/Pearson_Correlation_Heatmap_HCMonoColLamABBMM_rmBE.pdf", width = 8, height = 7)
par(cex.main = 1.5, cex.lab = 2, cex.sub = 2.5, cex.axis = 1.75, lend = 1, family = "ArialMT")
heatmap.2(z, Colv = as.dendrogram(hz), Rowv = as.dendrogram(hz), col = clustercol, na.rm = TRUE, scale = "none", margins = c(2,10), key = TRUE, density.info = "none", trace = "none", key.title = NA, key.xlab = "R", cexRow = 2, RowSideColors = colors_group, ColSideColors = colors_group, colRow = colors_group, labCol = FALSE, labRow = group)
dev.off()

### Saving data for dotplots of selected genes ###

splnames <- unlist(strsplit(as.character(rownames(logcpm_rmBE)), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
rown <- data.frame(row.matrix)
data_box <- cbind(rown, logcpm_rmBE)
write.table(data_box, file = "£{ANALYSISDIR}/HCBMMMonoColLamAB_logcpm_rmBE_data_dotplots.txt", sep = "\t", col.names = NA, quote = FALSE)

### Starting analysis in EdgeR for differnetially expressed genes ###

group <- factor(c(rep("HC", 3), rep("BMM", 7), rep("Mono", 3), rep("Collagen", 6), rep("Laminin", 3), rep("AB", 3)))
donor <- factor(c(0,0,0,0,5,5,5,5,0,0,1,2,0,1,2,3,1,2,3,1,2,3,1,2,3))

#Design matrix with batches
design <- model.matrix(~0 + group + donor)
rownames(design) <- colnames(d)
design

#calculate dispersions needed for glm fitting
d <- estimateDisp(d, design = design)
d\£common.dispersion
plotBCV(d)
#fit glm model to data
fit <- glmQLFit(d, design = design, robust = TRUE)
plotQLDisp(fit)

#specify contrasts between groups
my.contrasts <- makeContrasts(HCvsCol = groupHC - groupCollagen,
                              HCvsBMM = groupHC - groupBMM,
                              HCvsMono = groupHC - groupMono,
                              HCvsAB = groupHC - groupAB,
                              ColvsBMM = groupCollagen - groupBMM,
                              ColvsAB = groupCollagen - groupAB,
                              ColvsMono = groupCollagen - groupMono,
                              BMMvsMono = groupBMM - groupMono,
                              BMMvsAB = groupBMM - groupAB,
                              ABvsMono = groupAB - groupMono, levels = design)

my.contrast2 <- makeContrasts(HCvsCol = groupHC - groupCollagen,
                              HCvsBMM = groupHC - groupBMM,
                              ColvsBMM = groupCollagen - groupBMM, levels = design)


### Anova like analysis ###

lrt_anova <- glmQLFTest(fit, contrast = my.contrasts)
stat_anova <- topTags(lrt_anova, n = Inf)
write.table(stat_anova, file = "£{ANALYSISDIR}/HCBMMColLamABMono_stat_anova.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.anova <- decideTestsDGE(lrt_anova))

lrt_anova2 <- glmQLFTest(fit, contrast = my.contrast2)
stat_anova2 <- topTags(lrt_anova2, n = Inf)
write.table(stat_anova2, file = "£{ANALYSISDIR}/HCBMMColLamABMono_stat_anova2.txt", sep = "\t", row.names = TRUE, col.names = NA)
summary(dt.anova2 <- decideTestsDGE(lrt_anova2))

#merge information about differentially expressed genes with expression data
x1 <- read.delim(file = "£ANALYSISDIR/HCBMMMonoColLamAB_logcpm_rmBE.txt", sep = "\t")
colnames(x1) <- c("ID", colnames(logcpm_rmBE))

x2 <- read.delim(file = "£{ANALYSISDIR}/HCBMMColLamABMono_stat_anova.txt", sep = "\t")
colnames(x2) <- c("ID", colnames(stat_anova))
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{ANALYSISDIR}/HCBMMColLamABMono_anova_diffExp_rmBE.txt",sep = "\t", col.names = NA)

x2 <- read.delim(file = "£{ANALYSISDIR}/HCBMMColLamABMono_stat_anova2.txt", sep = "\t")
colnames(x2) <- c("ID", colnames(stat_anova2))
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{ANALYSISDIR}/HCBMMColLamABMono_anova2_diffExp_rmBE.txt",sep = "\t", col.names = NA)

EOF
```

```
chmod 750 "/loctmp/R.edgeR.HC.£{_Date}.R"
R < /loctmp/R.edgeR.HC.£{_Date}.R --no-save
rm /loctmp/R.edgeR.HC.£{_Date}.R

#####################################
#### Dotplots with several genes ####
#####################################

cat > "/loctmp/Dotplots_HC_CK.R" << EOF

library(ggplot2)
library(sqldf)
library(reshape2)

  data <- read.delim(file = "£{ANALYSISDIR}/HCBMMMonoColLamAB_logcpm_rmBE_data_dotplots.txt", sep = "\t", row.names = "X")
#HC specific
HCList=c("NR4A1", "WNT5A", "SPRY2", "EGR2", "CSF1", "NFKB1", "PPARG", "MAP3K8", "AMOT")
HC <- data.frame(HCList)
colnames(HC) <- c("HC")
### create rpkm values ###

data\£RPKM_HC1 <- ((2^data\£HC_6) / data\£TranscriptLength) * 1000
data\£RPKM_HC2 <- ((2^data\£HC_7) / data\£TranscriptLength) * 1000
data\£RPKM_HC3 <- ((2^data\£HC_8) / data\£TranscriptLength) * 1000

data\£RPKM_BMM1 <- ((2^data\£BMM_143) / data\£TranscriptLength) * 1000
data\£RPKM_BMM2 <- ((2^data\£BMM_144) / data\£TranscriptLength) * 1000
data\£RPKM_BMM3 <- ((2^data\£BMM_145) / data\£TranscriptLength) * 1000
data\£RPKM_BMM4 <- ((2^data\£BMM_146) / data\£TranscriptLength) * 1000
data\£RPKM_BMM5 <- ((2^data\£BMM_EM1A) / data\£TranscriptLength) * 1000
data\£RPKM_BMM6 <- ((2^data\£BMM_GS) / data\£TranscriptLength) * 1000

data\£RPKM_M1Col1 <- ((2^data\£M1_Collagen_I) / data\£TranscriptLength) * 1000
data\£RPKM_M3Col1 <- ((2^data\£M3_Collagen_I) / data\£TranscriptLength) * 1000
data\£RPKM_M6Col1 <- ((2^data\£M6_Collagen_I) / data\£TranscriptLength) * 1000
data\£RPKM_M1Col4 <- ((2^data\£M1_Collagen_IV) / data\£TranscriptLength) * 1000
data\£RPKM_M3Col4 <- ((2^data\£M3_Collagen_IV)/ data\£TranscriptLength) * 1000
data\£RPKM_M6Col4 <- ((2^data\£M6_Collagen_IV) / data\£TranscriptLength) * 1000

data\£RPKM_M1AB <- ((2^data\£M1_AB_TcPlastic) / data\£TranscriptLength) * 1000
data\£RPKM_M3AB <- ((2^data\£M3_AB_TcPlastic) / data\£TranscriptLength) * 1000
data\£RPKM_M6AB <- ((2^data\£M6_AB_TcPlastic) / data\£TranscriptLength) * 1000
data\£RPKM_M1Lam <- ((2^data\£M1_Laminin) / data\£TranscriptLength) * 1000
data\£RPKM_M3Lam <- ((2^data\£M3_Laminin) / data\£TranscriptLength) * 1000
data\£RPKM_M6Lam <- ((2^data\£M6_Laminin) / data\£TranscriptLength) * 1000

data2 <- sqldf("SELECT GeneSymbol, RPKM_M1Col1, RPKM_M3Col1, RPKM_M6Col1, RPKM_M1Col4, RPKM_M3Col4, RPKM_M6Col4, RPKM_M1AB, RPKM_M3AB,
↪→  RPKM_M6AB, RPKM_M1Lam, RPKM_M3Lam, RPKM_M6Lam, RPKM_HC1, RPKM_HC2, RPKM_HC3, RPKM_BMM1, RPKM_BMM2, RPKM_BMM3, RPKM_BMM4, RPKM_BMM5,
↪→  RPKM_BMM6 FROM data INNER JOIN HC ON GeneSymbol = HC")

md <- melt(data2)
md\£Type <- c(rep("Collagen I", 27), rep("Collagen IV", 27), rep("AB", 27), rep("Laminin", 27), rep("HC", 27), rep("BMM", 54))
md\£Type <- factor(md\£Type, levels = c("AB", "Laminin", "Collagen I", "Collagen IV", "BMM", "HC") )

md\£GeneSymbol <- factor(md\£GeneSymbol, levels = c("NR4A1", "WNT5A", "SPRY2", "EGR2", "CSF1", "NFKB1", "PPARG", "MAP3K8", "AMOT"))

p1 <- ggplot(md) + theme_bw() +
        geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£colAB", "£colLam", "£colCol1", "£colCol4", "£colBMM", "£colHC"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪→  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪→  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR/HC_expression.pdf", width = 10, height = 4)
p1
dev.off()

HCDown=c("LYZ", "STAT6", "RUNX2", "ELK4", "CASP8", "MMP25", "HDAC1", "HDAC3", "MBD2")
HCDown <- data.frame(HCDown)
colnames(HCDown) <- c("HC")

data2 <- sqldf("SELECT GeneSymbol, RPKM_M1Col1, RPKM_M3Col1, RPKM_M6Col1, RPKM_M1Col4, RPKM_M3Col4, RPKM_M6Col4, RPKM_M1AB, RPKM_M3AB,
↪→  RPKM_M6AB, RPKM_M1Lam, RPKM_M3Lam, RPKM_M6Lam, RPKM_HC1, RPKM_HC2, RPKM_HC3, RPKM_BMM1, RPKM_BMM2, RPKM_BMM3, RPKM_BMM4, RPKM_BMM5,
↪→  RPKM_BMM6 FROM data INNER JOIN HCDown ON GeneSymbol = HC")

md <- melt(data2)
md\£Type <- c(rep("Collagen I", 27), rep("Collagen IV", 27), rep("AB", 27), rep("Laminin", 27), rep("HC", 27), rep("BMM", 54))
md\£Type <- factor(md\£Type, levels = c("AB", "Laminin", "Collagen I", "Collagen IV", "BMM", "HC") )

md\£GeneSymbol <- factor(md\£GeneSymbol, levels = c("LYZ", "STAT6", "RUNX2", "ELK4", "CASP8", "MMP25", "HDAC1", "HDAC3", "MBD2"))
```

```
md

p1 <- ggplot(md) + theme_bw() +
        geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£colAB", "£colLam", "£colCol1", "£colCol4", "£colBMM", "£colHC"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR/HCDown_expression.pdf", width = 10, height = 4)
p1
dev.off()

HCCol=c("CD163", "CD28", "CD209", "C1QA", "C1QB", "C1QC", "CSF1R", "MAF", "NFATC2")
HCCol <- data.frame(HCCol)
colnames(HCCol) <- c("HCCol")

data2 <- sqldf("SELECT GeneSymbol, RPKM_M1Col1, RPKM_M3Col1, RPKM_M6Col1, RPKM_M1Col4, RPKM_M3Col4, RPKM_M6Col4, RPKM_M1AB, RPKM_M3AB,
↪  RPKM_M6AB, RPKM_M1Lam, RPKM_M3Lam, RPKM_M6Lam, RPKM_HC1, RPKM_HC2, RPKM_HC3, RPKM_BMM1, RPKM_BMM2, RPKM_BMM3, RPKM_BMM4, RPKM_BMM5,
↪  RPKM_BMM6 FROM data INNER JOIN HCCol ON GeneSymbol = HCCol")

md <- melt(data2)
md\£Type <- c(rep("Collagen I", 27), rep("Collagen IV", 27), rep("AB", 27), rep("Laminin", 27), rep("HC", 27), rep("BMM", 54))
md\£Type <- factor(md\£Type, levels = c("AB", "Laminin", "Collagen I", "Collagen IV", "BMM", "HC") )

md\£GeneSymbol <- factor(md\£GeneSymbol, levels = c("CD163", "CD28", "CD209", "C1QA", "C1QB", "C1QC", "CSF1R", "MAF", "NFATC2"))

md

p1 <- ggplot(md) + theme_bw() +
        geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£colAB", "£colLam", "£colCol1", "£colCol4", "£colBMM", "£colHC"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR/HCCol_expression.pdf", width = 10, height = 4)
p1
dev.off()

EOF
chmod 750 "/loctmp/Dotplots_HC_CK.R"
R < /loctmp/Dotplots_HC_CK.R --no-save
rm /loctmp/Dotplots_HC_CK.R
```

```
#!/bin/bash
#File: EGR2KD_DiffExpAnalysis
#by Claudia Kiesewetter started on 28.09.2018
#This script was written to perform differential gene expression analysis for RNAseq Data on EGR2 Knock Down#Samples

#setting up environment variables

DIR_PKG="/misc/software/ngs"
PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin
export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH

ANALYSISDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/EGR2KD"
DATADIR="/misc/data/processedData/mapping/RNA/GRCh38/RNAseq/MOMACDC/"
COLDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6/Heatmaps"

mkdir -p $ANALYSISDIR/tmp

### Setting colors for the different cell types ###

colMono="forestgreen"
colCol1="deeppink"
colCol4="violetred3"
colAB="slateblue3"
colLam="orchid1"
colFib="plum3"
colCol4Lam="hotpink"
colCTRL="dodgerblue3"
colMock="royalblue2"
col1132="gold2"
col2665="goldenrod3"
```

```
### Creating a gene count table from STAR output ###

#pasting all files together in the specified order
cd $DATADIR
paste /misc/software/ngs/genome/sequence/GRCh38.PRI_p10/STAR_transcriptIDshort.txt RNAseq_CK_89_exp129_Mono.ReadsPerGene.txt
↪  RNAseq_CK_94_exp130_Mono.ReadsPerGene.txt RNAseq_CK_99_exp132_Mono.ReadsPerGene.txt RNAseq_CK_90_exp129_Mock.ReadsPerGene.txt
↪  RNAseq_CK_95_exp130_Mock.ReadsPerGene.txt RNAseq_CK_100_exp132_Mock.ReadsPerGene.txt RNAseq_CK_91_exp129_CTRL.ReadsPerGene.txt
↪  RNAseq_CK_96_exp130_CTRL.ReadsPerGene.txt RNAseq_CK_101_exp132_CTRL.ReadsPerGene.txt RNAseq_CK_92_exp129_1132.ReadsPerGene.txt
↪  RNAseq_CK_97_exp130_1132.ReadsPerGene.txt RNAseq_CK_102_exp132_1132.ReadsPerGene.txt RNAseq_CK_93_exp129_2665.ReadsPerGene.txt
↪  RNAseq_CK_98_exp130_2665.ReadsPerGene.txt RNAseq_CK_103_exp132_2665.ReadsPerGene.txt > /loctmp/GeneCount_EGR2KD.txt

#selecting correct column
#Scriptseq: fourth column of star output is selected (lowest No_Feature Count)
cut -f 1,4,8,12,16,20,24,28,32,36,40,44,48,52,56,60 /loctmp/GeneCount_EGR2KD.txt > /loctmp/GeneCount_EGR2KD_cut.txt
tail -n +5 /loctmp/GeneCount_EGR2KD_cut.txt > /loctmp/GeneCount_EGR2KD_tail.txt
#change header
echo
↪  £'ID\texp129_Mono\texp130_Mono\texp132_Mono\texp129_Mock\texp130_Mock\texp132_Mock\texp129_CTRL\texp130_CTRL\texp132_CTRL\texp129_1132\texp130_1132\texp132_1132\
↪  | cat - /loctmp/GeneCount_EGR2KD_tail.txt > $ANALYSISDIR/GeneCount_EGR2KD_red.txt

### Starting basic Analysis in R ###

cd £{ANALYSISDIR}
_DATE=$(date +%s)
cat > "/loctmp/R.edgeR.P.£{_DATE}.R" <<EOF
library(ggplot2)
library(edgeR)
library(RColorBrewer)
library(ggrepel)
library(gplots)
library(sqldf)
library(Rtsne)
library(ggpubr)
library(extrafont)

#Read in Data Table:
data <- read.delim(file = "£{ANALYSISDIR}/GeneCount_EGR2KD_red.txt", sep = "\t", row.names = "ID")

#Creating EdgeR DGEList object
group <- c(rep("Mono", 3), rep("Mock", 3), rep("CTRL", 3), rep("siR1132", 3), rep("siR2665", 3))
d <- DGEList(counts = data, group = group)
keep <- rowSums(cpm(d) > 1) >= 5 #filtering of genes
summary(keep)
d <- d[keep, keep.lib.sizes=FALSE]
d <- calcNormFactors(d) #normalisierung!
d£samples
#Log2 of orgininal expression value
logcpm <- cpm(d, prior.count = 2, normalized.lib.sizes = TRUE, log = TRUE)

### First MDS Plots ###
group <- factor(c(rep("Mono", 3), rep("Mock", 3), rep("CTRL", 3), rep("siR1132", 3), rep("siR2665", 3)))
colors = c(rep("£colCTRL", 1), rep("£colMock", 1), rep("£colMono", 1), rep("£col1132", 1), rep("£col2665", 1))

prep_batch <- factor(c(1,1,2,1,1,2,1,1,2,1,2,2,1,2,2))
colors_batch <- c("goldenrod", "dodgerblue4")

donor_batch <- factor(c(1,2,3,1,2,3,1,2,3,1,2,3,1,2,3))
colors_donor = c("dodgerblue4", "cyan3", "darkorange1")

png(file="£{ANALYSISDIR}/tmp/MDSPlot_EGR2KD_AllSamples_colouredCondition.png", width = 800, height = 800)
plotMDS(logcpm, col = colors[group], cex = 1.5)
dev.off()
png(file="£{ANALYSISDIR}/tmp/MDSPlot_EGR2KD_AllSamples_colouredBatch1.png", width = 800, height = 800)
plotMDS(logcpm, col = colors_batch[prep_batch], cex = 0.85, main = "Coloured by Preparation Batches")
dev.off()
png(file="£{ANALYSISDIR}/tmp/MDSPlot_EGR2KD_AllSamples_colouredBatch2.png", width = 800, height = 800)
plotMDS(logcpm, col = colors_donor[donor_batch], cex = 0.85, main = "Coloured by Donor")
dev.off()

### Remove Batch Effect from Data ###

prep_batch <- factor(c(0,1,2,1,1,2,1,1,2,1,2,3,1,2,3))
donor_batch <- factor(c(1,2,3,1,2,3,1,2,3,1,2,3,1,2,3))

design1 <- model.matrix(~0 + group)
rownames(design1) <- colnames(d)
logcpm_rmBE <- removeBatchEffect(logcpm, design = design1, batch = donor_batch, batch2 = prep_batch)

### Second MDS Plots ###
png(file="£{ANALYSISDIR}/tmp/MDSPlot_EGR2KD_AllSamples_rmBE.png", width = 800, height = 800)
plotMDS(logcpm_rmBE, col = colors[group], cex = 1.5)
```

```
dev.off()
write.table(logcpm_rmBE, file = "£ANALYSISDIR/EGR2KD_logcpm_rmBE.txt", sep = "\t", col.names = NA)

### Saving data for dotplots of selected genes ###

splnames <- unlist(strsplit(as.character(rownames(logcpm_rmBE)), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
rown <- data.frame(row.matrix)
data_box <- cbind(rown, logcpm_rmBE)
write.table(data_box, file = "£{ANALYSISDIR}/Logcpm_rmBE_data_dotplots.txt", sep = "\t", col.names = NA, quote = FALSE)

### Differentially Expressed Genes using GLM Method ###

## Creating Designs for later comparison of specific groups ##

prep_batch <- factor(c(0,1,2,1,1,2,1,1,2,1,2,3,1,2,3))
donor_batch <- factor(c(1,2,3,1,2,3,1,2,3,1,2,3,1,2,3))
group2 <- c(rep("Mono", 3), rep("CTRL", 6), rep("siRNA", 6))

design2 <- model.matrix(~0 + group2 + donor_batch + prep_batch)
rownames(design2) <- colnames(d)
design2

#Calculate Dispersions for Designs

d2 <- estimateDisp(d, design = design2, robust = TRUE)
d2\£common.dispersion
fit2 <- glmQLFit(d2, design = design2)
plotQLDisp(fit2)

my.contrasts2 <- makeContrasts(siRNAvsCTRL = group2siRNA - group2CTRL, levels = design2)

## Calculating Differentially Expressed Genes ###

lrt_siRNAvsCTRL <- glmQLFTest(fit2, contrast = my.contrasts2[, "siRNAvsCTRL"])
stat_siRNAvsCTRL <- topTags(lrt_siRNAvsCTRL, n = Inf)
write.table(stat_siRNAvsCTRL, file = "£ANALYSISDIR/EGR2KD_AllSamples_stat_siRNAvsCTRL.txt", sep = "\t", row.names = TRUE, col.names =
↪  NA)
summary(dt.siRNAvsCTRL <- decideTestsDGE(lrt_siRNAvsCTRL))

## Merging statistics with expresison values for all Genes. Here the LogcPM after remove Batch Effect values are used ###

x1 <- read.delim("EGR2KD_logcpm_rmBE.txt", sep = "\t")
colnames(x1) <- c("ID", colnames(logcpm_rmBE))

x2 <- read.delim("EGR2KD_AllSamples_stat_siRNAvsCTRL.txt", sep = "\t")
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "EGR2KD_AllSamples_siRNAvsCTRL_diffExp_rmBE.txt", sep = "\t", col.names = NA)

### Data preprocessing for the Heatmap ###
x <- read.delim("EGR2KD_AllSamples_siRNAvsCTRL_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep = "\t")
subdata <- subset(x, abs(logFC) > 1.5 & FDR < 0.05 & logCPM > 1, select = c(exp129_Mock, exp130_Mock, exp132_Mock, exp129_CTRL,
↪  exp130_CTRL, exp132_CTRL, exp129_1132, exp130_1132, exp132_1132, exp129_2665, exp130_2665, exp132_2665))
data.scaled <- (scale(t(subdata)))
data <- data.matrix(t(data.scaled))
#Clustering
hc <- hclust(dist(t(data), method = "manhattan"), method = "ward.D2")
hr <- hclust(dist(data, method = "manhattan"), method = "ward.D2")
clustercol <- colorRampPalette(c("dodgerblue4", "white", "firebrick3"))(299)
col_breaks = c(seq(-2, -0.75, length = 100), seq(-0.749,0.749, length = 100), seq(0.75,2,length=100))
#mark clusters
mycl <- cutree(hr, h=max(hr\£height/1.5))
clustercols <- rainbow(length(unique(mycl)))
myClusterSideBar <- clustercols[mycl]
#create data table
splnames <- unlist(strsplit(as.character(rownames(t(data.scaled))), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
subdata.clustered <- cbind(t(data.scaled), row.matrix, clusterID=mycl)
clustered.data <- apply(subdata.clustered[hr\£order,],2,rev)
write.table(clustered.data, file = "EGR2KD_AllSamples_Table_ManhattanWardZscores_stat_siRNAvsCTRL.txt", sep = "\t", col.names = NA)

clustercols <- c("seagreen", "goldenrod3")
myClusterSideBar <- clustercols[mycl]

colors <- c(rep("£colCTRL", 3), rep("£colMock", 3), rep("£col1132", 3), rep("£col2665", 3))

pdf(file = "£ANALYSISDIR/EGR2KD_Heatmap_siRNAvsCTRL.pdf", width = 11, height = 9)
par(cex.main = 3, cex.lab = 2.5, cex.sub = 2, cex.axis = 2, family = "ArialMT")
heatmap.2(data, Rowv = as.dendrogram(hr), Colv = as.dendrogram(hc), col = clustercol,
labRow = FALSE, labCol = FALSE,
na.rm = TRUE, scale = "none", margins = c(2,10),
cexCol=1, key = TRUE, density.info = "none", trace = "none", key.title = NA, key.xlab = "Z score", dendrogram ="column",
RowSideColors = myClusterSideBar, ColSideColors = colors)
dev.off()

splnames <- unlist(strsplit(as.character(rownames(x)), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
x1 <- cbind(x, row.matrix)

#Add information about Collagen regulated Genes:

ColvsRest <- read.delim(file = "£COLDIR/Table_ManhattanWardZscores_AllSamples_pairwiseComparisons.logFC2.FDR0.05.txt", sep = "\t")

Col_Up <- sqldf("SELECT GeneSymbol FROM ColvsRest WHERE clusterID = '1'")
colnames(Col_Up) <- c("GeneName")
nrow(Col_Up)
EGR2_Up_Col_Up <- sqldf("SELECT GeneSymbol, FDR, logFC FROM x1 INNER JOIN Col_Up ON GeneSymbol = GeneName")
nrow(EGR2_Up_Col_Up)

Col_Down <- sqldf("SELECT GeneSymbol FROM ColvsRest WHERE clusterID = '2'")
colnames(Col_Down) <- c("GeneName")
nrow(Col_Down)
EGR2_Down_Col_Down <- sqldf("SELECT GeneSymbol, FDR, logFC FROM x1 INNER JOIN Col_Down ON GeneSymbol = GeneName")

x1\£Significance <- ifelse(x\£FDR < 0.05, "FDR < 0.05", "Not Significant")
x2 <- sqldf("SELECT GeneSymbol, FDR, logFC FROM x1 WHERE logFC > 1.5 AND FDR < 0.05")
x3 <- sqldf("SELECT GeneSymbol, FDR, logFC FROM x1 WHERE logFC < -1.5 AND FDR < 0.05")
x4 <- sqldf("SELECT GeneSymbol, FDR, logFC FROM EGR2_Up_Col_Up WHERE logFC > 1.5 AND FDR < 0.05")
x5 <- sqldf("SELECT GeneSymbol, FDR, logFC FROM EGR2_Down_Col_Down WHERE logFC < -1.5 AND FDR < 0.05")

p2 <- ggplot(x1, aes(x = logFC, y = -log10(FDR))) + theme_classic(base_size = 16) +
geom_point(data = x1, col = "gray70", alpha = 0.5) +
geom_point(data = x2, col = "firebrick2", alpha = 0.2) +
geom_point(data = x3, col = "dodgerblue1", alpha = 0.2) +
geom_point(data = x4, col = "firebrick4", alpha = 0.7) +
geom_point(data = x5, col = "dodgerblue4", alpha = 0.7) +
labs(x = "Fold Change (log2)", y = "FDR (-log10)")

pdf(file = "£{ANALYSISDIR}/VolcanoPlot_EGR2KD_withColvsRest_addedPoints.pdf", width = 9, height = 9)
par(family = "ArialMT")
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.edgeR.P.£{_DATE}.R"
R < /loctmp/R.edgeR.P.£{_DATE}.R  --no-save
rm /loctmp/R.edgeR.P.£{_DATE}.R
```

```
#!/usr/bin/env bash
#File: Dec2KD_DiffExpAnalysis.sh by Claudia Kiesewetter October 6, 2018
#Script for the analysis of differential genes by Dec2 knock down, after mapping

#setting up environment variables

DIR_PKG="/misc/software/ngs"

PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin

export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH

ANALYSISDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/DEC2KD"
DATADIR="/misc/data/processedData/mapping/RNA/GRCh38/RNAseq/MOMACDC/"
COLDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6/Heatmaps"
TMPDIR="/loctmp"

mkdir -p $ANALYSISDIR/tmp

colMono="forestgreen"
colCTRL="dodgerblue3"
colMock="royalblue2"
col684="gold2"
col1793="goldenrod3"

### Creating a gene count table from STAR output ###
```

```
#pasting all files together in the specified order
cd $DATADIR
paste /misc/software/ngs/genome/sequence/GRCh38.PRI_p10/STAR_transcriptIDshort.txt RNAseq_CK_exp115_Mono.ReadsPerGene.txt
↪  RNAseq_CK_exp116_Mono.ReadsPerGene.txt RNAseq_CK_exp118_Mono.ReadsPerGene.txt RNAseq_CK_104_exp113_noT.ReadsPerGene.txt
↪  RNAseq_CK_exp115_noT.ReadsPerGene.txt RNAseq_CK_exp116_noT.ReadsPerGene.txt RNAseq_CK_exp118_noT.ReadsPerGene.txt
↪  RNAseq_CK_105_exp113_CTRL.ReadsPerGene.txt RNAseq_CK_exp115_CTRL.ReadsPerGene.txt RNAseq_CK_exp116_CTRL.ReadsPerGene.txt
↪  RNAseq_CK_exp118_CTRL.ReadsPerGene.txt RNAseq_CK_106_exp113_684.ReadsPerGene.txt RNAseq_CK_exp115_684.ReadsPerGene.txt
↪  RNAseq_CK_exp116_684.ReadsPerGene.txt RNAseq_CK_exp118_684.ReadsPerGene.txt RNAseq_CK_107_exp113_1793.ReadsPerGene.txt
↪  RNAseq_CK_exp115_1793.ReadsPerGene.txt RNAseq_CK_exp116_1793.ReadsPerGene.txt RNAseq_CK_exp118_1793.ReadsPerGene.txt >
↪  $ANALYSISDIR/GeneCount_Dec2KD.txt

cut -f 1,4,8,12,16,20,24,28,32,36,40,44,48,52,56,60,64,68,72,76 $ANALYSISDIR/GeneCount_Dec2KD.txt > $TMPDIR/GeneCount_Dec2KD_cut.txt

tail -n +5 $TMPDIR/GeneCount_Dec2KD_cut.txt > $TMPDIR/GeneCount_Dec2KD_tail.txt

echo
↪  £'ID\texp115_Mono\texp116_Mono\texp118_Mono\texp113_Mock\texp115_Mock\texp116_Mock\texp118_Mock\texp113_CTRL\texp115_CTRL\texp116_CTRL\texp118_CTRL\texp113_
↪  cat - $TMPDIR/GeneCount_Dec2KD_tail.txt > $ANALYSISDIR/GeneCount_DEC2KD_red.txt

### Starting basic Analysis in R ###

cd £{ANALYSISDIR}
_DATE=$(date +%s)
cat > "/loctmp/R.edgeRDec2.£{_DATE}.R" <<EOF
library(ggplot2)
library(edgeR)
library(RColorBrewer)
library(ggrepel)
library(gplots)
library(sqldf)
library(Rtsne)
library(ggpubr)
library(extrafont)
library(VennDiagram)

data <- read.delim(file = "£{ANALYSISDIR}/GeneCount_DEC2KD_red.txt", sep = "\t", row.names = "ID")

group <- c(rep("Mono", 3), rep("Mock", 4), rep("CTRL", 4), rep("siR684", 4), rep("siR1793", 4))
d <- DGEList(counts = data, group = group)
keep <- rowSums(cpm(d) > 1) >= 2
summary(keep)
d <- d[keep, keep.lib.sizes=FALSE]
d <- calcNormFactors(d) #normalisierung!
d£samples

logcpm <- cpm(d, prior.count = 2, normalized.lib.sizes = TRUE, log = TRUE)

group <- factor(c(rep("Mono", 3), rep("Mock", 4), rep("CTRL", 4), rep("siR684", 4), rep("siR1793", 4)))
colors = c(rep("forestgreen", 1), rep("royalblue2", 1), rep("dodgerblue4", 1), rep("indianred1", 1), rep("indianred3", 1))

prep_batch <- factor(c(1,1,2,1,1,2,3,1,1,2,3,1,2,2,3,1,2,2,3))
colors_batch <- c("goldenrod", "dodgerblue4", "chartreuse3")

donor_batch <- factor(c(1,2,3,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4))
colors_donor = c("dodgerblue4", "cyan3", "darkorange1", "chartreuse2")

png(file = "£{ANALYSISDIR}/tmp/MDSPlot_Dec2KD_AllSamples.png", width = 800, height = 800)
plotMDS(logcpm, col = colors[group], cex = 1.5)
dev.off()

png(file = "£{ANALYSISDIR}/tmp/MDSPlot_Dec2KD_AllSamples_prepBatch.png", width = 800, height = 800)
plotMDS(logcpm, col = colors_batch[prep_batch], cex = 0.85, main = "Coloured by Preparation Batches")
dev.off()

png(file = "£{ANALYSISDIR}/tmp/MDSPlot_Dec2KD_AllSamples_donorBatch.png", width = 800, height = 800)
plotMDS(logcpm, col = colors[group], cex = 1.5)
plotMDS(logcpm, col = colors_donor[donor_batch], cex = 0.85, main = "Coloured by Donor")
dev.off()

### Remove Batch Effect ###

prep_batch <- factor(c(1,1,2,1,1,2,3,1,1,2,3,1,2,2,3,1,2,2,3))
donor_batch <- factor(c(1,2,0,1,2,0,0,1,2,0,0,1,2,0,0,1,2,0,0))

design1 <- model.matrix(~0 + group)
rownames(design1) <- colnames(d)
logcpm_rmBE <- removeBatchEffect(logcpm, design = design1, batch = prep_batch, batch2 = donor_batch)

png(file = "£{ANALYSISDIR}/tmp/MDSPlot_Dec2KD_AllSamples_rmBE.png", width = 800, height = 800)
plotMDS(logcpm_rmBE, col = colors[group], cex = 1.5, main = "Coloured by Group")
dev.off()
```

```
write.table(logcpm_rmBE, file = "£{ANALYSISDIR}/Dec2KD_AllSamples_logcpm_rmBE.txt", sep = "\t", col.names = NA)

### Saving data for dotplots of selected genes ###

splnames <- unlist(strsplit(as.character(rownames(logcpm_rmBE)), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
rown <- data.frame(row.matrix)
data_box <- cbind(rown, logcpm_rmBE)
write.table(data_box, file = "£{ANALYSISDIR}/Logcpm_rmBE_data_dotplots.txt", sep = "\t", col.names = NA, quote = FALSE)

prep_batch <- factor(c(1,1,2,1,1,2,3,1,1,2,3,1,2,2,3,1,2,2,3))

donor_batch <- factor(c(1,2,0,1,2,0,0,1,2,0,0,1,2,0,0,1,2,0,0))

group <- factor(c(rep("Mono", 3), rep("Mock", 4), rep("CTRL", 4), rep("siR684", 4), rep("siR1793", 4)))
group2 <- factor(c(rep("Mono", 3), rep("CTRL", 8), rep("siRNA", 8)))

design <- model.matrix(~0 + group + donor_batch + prep_batch)
design2 <- model.matrix(~0 + group2 + prep_batch + donor_batch)

rownames(design) <- colnames(d)
rownames(design2) <- colnames(d)

design
design2

#Calculate Dispersions for Designs

d <- estimateDisp(d, design = design, robust = TRUE)
d\£common.dispersion
fit <- glmQLFit(d, design = design)
plotQLDisp(fit)

d2 <- estimateDisp(d, design = design2, robust = TRUE)
d2\£common.dispersion
fit2 <- glmFit(d2, design = design2)

my.contrasts <- makeContrasts(siR1793vsCTRL = groupsiR1793 - groupCTRL,
siR684vsCTRL = groupsiR684 - groupCTRL,
siR1793vssiR684 = groupsiR1793 - groupsiR684,
siR1793vsnoT = groupsiR1793 - groupMock,
siR684vsnoT = groupsiR684 - groupMock,
noTvsCTRL = groupMock - groupCTRL, levels = design)

my.contrasts2 <- makeContrasts(siRNAvsCTRL = group2siRNA - group2CTRL, levels = design2)

## Calculating Differentially Expressed Genes ###

lrt_siR1793vsCTRL <- glmQLFTest(fit, contrast = my.contrasts[, "siR1793vsCTRL"])
stat_siR1793vsCTRL <- topTags(lrt_siR1793vsCTRL, n = Inf)
write.table(stat_siR1793vsCTRL, file = "£{ANALYSISDIR}/Dec2KD_AllSamples_stat_siR1793vsCTRL.txt", sep = "\t", row.names = TRUE,
↪  col.names = NA)
summary(dt.siR1793vsCTRL <- decideTestsDGE(lrt_siR1793vsCTRL))

lrt_siR684vsCTRL <- glmQLFTest(fit, contrast = my.contrasts[, "siR684vsCTRL"])
stat_siR684vsCTRL <- topTags(lrt_siR684vsCTRL, n = Inf)
write.table(stat_siR684vsCTRL, file = "£{ANALYSISDIR}/Dec2KD_AllSamples_stat_siR684vsCTRL.txt", sep = "\t", row.names = TRUE, col.names
↪  = NA)
summary(dt.siR684vsCTRL <- decideTestsDGE(lrt_siR684vsCTRL))

lrt_siRNAvsCTRL <- glmLRT(fit2, contrast = my.contrasts2[, "siRNAvsCTRL"])
stat_siRNAvsCTRL <- topTags(lrt_siRNAvsCTRL, n = Inf)
write.table(stat_siRNAvsCTRL, file = "£{ANALYSISDIR}/Dec2KD_AllSamples_stat_siRNAvsCTRL.txt", sep = "\t", row.names = TRUE, col.names =
↪  NA)
summary(dt.siRNAvsCTRL <- decideTestsDGE(lrt_siRNAvsCTRL))

#Merging statistics with expresison values for all Genes. Here the LogcPM after remove Batch Effect values are used ###

x1 <- read.delim(file = "£{ANALYSISDIR}/Dec2KD_AllSamples_logcpm_rmBE.txt", sep = "\t")
colnames(x1) <- c("ID", colnames(logcpm_rmBE))

x2 <- read.delim(file = "£{ANALYSISDIR}/Dec2KD_AllSamples_stat_siRNAvsCTRL.txt", sep = "\t")
colnames(x2) <- c("ID", "logFC", "logCPM", "F", "PValue", "FDR")
x <- merge(x1, x2, by.x = "ID", by.y = "ID")
write.table(x, file = "£{ANALYSISDIR}/Dec2KD_AllSamples_siRNAvsCTRL_diffExp_rmBE.txt", sep = "\t", col.names = NA)

### Heatmap siRNA vs CTRL ###

x <- read.delim(file = "£{ANALYSISDIR}/Dec2KD_AllSamples_siRNAvsCTRL_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep
↪  = "\t")
```

```
subdata <- subset(x, abs(logFC) > 1 & FDR < 0.05, select = c(exp113_Mock, exp115_Mock, exp116_Mock, exp118_Mock, exp113_CTRL,
↪  exp115_CTRL, exp116_CTRL, exp118_CTRL, exp113_684, exp115_684, exp116_684, exp118_684, exp113_1793, exp115_1793, exp116_1793,
↪  exp118_1793))
data.scaled <- (scale(t(subdata)))
data <- data.matrix(t(data.scaled))
#Clustering
hc <- hclust(dist(t(data), method = "manhattan"), method = "ward.D2")
hr <- hclust(dist(data, method = "manhattan"), method = "ward.D2")
clustercol <- colorRampPalette(c("dodgerblue4", "white", "firebrick3"))(299)
col_breaks = c(seq(-2, -0.75, length = 100), seq(-0.749,0.749, length = 100), seq(0.75,2,length=100))
#mark clusters
mycl <- cutree(hr, h=max(hr\£height/1.5))
clustercols <- rainbow(length(unique(mycl)))
myClusterSideBar <- clustercols[mycl]
#create data table
splnames <- unlist(strsplit(as.character(rownames(t(data.scaled))), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
subdata.clustered <- cbind(t(data.scaled), row.matrix, clusterID=mycl)
clustered.data <- apply(subdata.clustered[hr\£order,],2,rev)
write.table(clustered.data, file = "£{ANALYSISDIR}/Dec2KD_AllSamples_Table_ManhattanWardZscores_stat_siRNAvsCTRL.txt", sep = "\t",
↪  col.names = NA)
clustered <- data.frame(clustered.data)

clustercols <- c("lightseagreen", "gold3")
myClusterSideBar <- clustercols[mycl]

colors <- c(rep("£colCTRL", 4), rep("£colMock", 4), rep("£col684", 4), rep("£col1793", 4))

pdf("Dec2KD_AllSamples_Heatmap_siRNAvsCTRL_logFC1.pdf", width = 12, height = 10)
par(cex.main = 3, cex.lab = 2.5, cex.sub = 2, cex.axis = 2, family = "ArialMT")
heatmap.2(data, Rowv = as.dendrogram(hr), Colv = as.dendrogram(hc), col = clustercol,
labRow = FALSE, labCol = FALSE,
na.rm = TRUE, scale = "none", margins = c(2,14),
cexCol=1, key = TRUE, density.info = "none", trace = "none", key.title = NA, key.xlab = "Z score", dendrogram ="column",
RowSideColors = myClusterSideBar, ColSideColors = colors)
dev.off()

EOF
chmod 750 "/loctmp/R.edgeRDec2.£{_DATE}.R"
R < /loctmp/R.edgeRDec2.£{_DATE}.R  --no-save
rm /loctmp/R.edgeRDec2.£{_DATE}.R
```

Several Scripts were used for t-SNE plotting in R, which basically follow the script below, just with different samples and colours:

```
#!/usr/bin/env bash
#File: tSNE_M1M3M6.sh created on 02.10.2018 by Claudia Kiesewetter
#This script runs the t-SNE algorithm for different macrophage populatiobs, while playing with the different parameters possible.

#Setting the environment variables

PATH_R=/misc/software/package/RBioC/3.4.3/bin

export PATH=£{PATH_R}:£{PATH}
export PATH

AnalysisDir="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6"

mkdir -p "$AnalysisDir/tsne"

### Setting colors for the different cell types ###

colCol1="deeppink"
colCol4="violetred3"
colABTcPlastic="slateblue3"
colLam="orchid1"
colMono="forestgreen"
colAdipo="khaki2"
colSero="lightsalmon2"
colTGF="darkgoldenrod3"
colHermin="brown4"
colNaCl="gold2"
colIgG="orange1"
colFib="plum3"
colMCSF="cornflowerblue"
coliDC="dodgerblue3"
```

```
colGMCSF="cadetblue3"
colNONAD="hotpink"
colABTeflon="wheat3"

### Checking number of arguments ###

if [[ $# -lt 2 ]] || [[ $# -gt 2 ]]
then
  echo "Correct Usage:"
  echo "bash tSNE_M1M3M6.sh <perlexity> <theta>"
  exit
fi

cd £{AnalysisDir}
_DATE=$(date +%s)

cat > "/loctmp/R.tSNE.M1M3M6.P$1.T$2.£{_DATE}.R" <<EOF
library(ggplot2)
library(Rtsne)
library(extrafont)

#specifying groups of samples
celltype <- factor(c(rep("Mono",3), rep("AB_Teflon", 2), rep("Adiponectin", 2), rep("Serotonin", 2), rep("TGF",2), rep("Hermin", 2),
↪  rep("NaCl", 2), rep("IgG", 2), rep("Collagen_I",3), rep("Collagen_IV",3), rep("Fibronectin", 3), rep("Laminin",3), rep("MCSF", 2),
↪  rep("AB_TcPlastic", 3), rep("iDC", 2), rep("GMCSF", 2), rep("Non-Adherent", 2)))
celltype

#specify colors used
colors = c("£colMono", "£colABTcPlastic", "£colLam", "£colFib", "£colNONAD", "£colCol1", "£colCol4", "£colABTeflon", "£colAdipo",
↪  "£colSero", "£colTGF", "£colHermin", "£colNaCl", "£colIgG", "£colMCSF", "£colGMCSF", "£coliDC")

colors_group <- c(rep("£colMono", 3), rep("£colABTeflon", 2), rep("£colAdipo", 2), rep("£colSero", 2), rep("£colTGF", 2),
↪  rep("£colHermin", 2), rep("£colNaCl", 2), rep("£colIgG", 2), rep("£colCol1", 3), rep("£colCol4", 3), rep("£colFib", 3),
↪  rep("£colLam", 3), rep("£colMCSF", 2), rep("£colABTcPlastic", 3), rep("£coliDC", 2), rep("£colGMCSF", 2), rep("£colNONAD", 2))

#save data in a numeric matrix
logcpm_rmBE <- read.delim(file = "£{AnalysisDir}/AllSamples_logcpm_rmBE.txt", sep = "\t", row.names= "X")
mydata <- data.matrix(t(logcpm_rmBE))

#Run tSNE algorithm with user specified parameters
rtsne_out <- Rtsne(mydata, check_duplicates = FALSE, pca = TRUE, perplexity= £1, theta=£2, dims=2, max_iter = 3000)

#create data table for plotting with ggplot
embedding <- as.data.frame(rtsne_out\£Y)
embedding\£Class <- as.factor(celltype)
embedding\£Color <- as.factor(colors_group)
embedding\£Class <- factor(embedding\£Class, levels = c("Mono", "AB_TcPlastic", "Laminin", "Fibronectin", "Non-Adherent", "Collagen_I",
↪  "Collagen_IV", "AB_Teflon", "Adiponectin", "Serotonin", "TGF", "Hermin", "NaCl", "IgG", "MCSF", "GMCSF", "iDC"))
embedding\£Class
embedding
#save data table in tsne folder
write.table(embedding, file = "£{AnalysisDir}/tsne/tSNE.P£1.T£2.£{_DATE}.txt", sep = "\t")

#starting ggplot:
p <- ggplot(embedding, aes(x=V1, y=V2)) +
geom_point(size=5, aes(color = embedding\£Class)) +
scale_colour_manual(values = colors, name = "Cell Type", labels = c("Monocytes", "AB-TcPlastic", "Laminin", "Fibronectin",
↪  "Non-adherent", "Collagen I", "Collagen IV", "AB-Teflon", "Adiponectin", "Serotonin", "TGF", "Hermin", "NaCl", "IgG", "M-CSF",
↪  "GM-CSF", "iDC"))  +
xlab("t-SNE 1") + ylab("t-SNE 2") +
theme_bw(base_size=20) +
theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), text = element_text(family = "ArialMT"), axis.text =
↪  element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT", face = "bold", size = 22), legend.title =
↪  element_text(family = "ArialMT", size = 22), legend.text = element_text(family = "ArialMT", size = 20), legend.key.size = unit(1,
↪  "cm"))

pdf(file = "£{AnalysisDir}/tsne/tSNE_M1M3M6.P£1.T£2.£{_DATE}.pdf", width = 12, height = 10)
plot(p, labels=TRUE)
dev.off()

EOF
chmod 750 "/loctmp/R.tSNE.M1M3M6.P$1.T$2.£{_DATE}.R"
R < /loctmp/R.tSNE.M1M3M6.P$1.T$2.£{_DATE}.R  --no-save
rm /loctmp/R.tSNE.M1M3M6.P$1.T$2.£{_DATE}.R

#+
#scale_colour_manual(values = colors, name = "Cell Type", limits = c("Mono", "AB_TcPlastic", "Laminin", "Fibronectin", "Non-Adherent",
↪  "Collagen_I", "Collagen_IV", "AB_Teflon", "Adiponectin", "Serotonin", "TGF", "Hermin", "NaCl", "IgG", "MCSF", "GMCSF", "iDC")) +
#xlab("t-SNE 1") + ylab("t-SNE 2") +
#theme_classic(base_size=20) +
#theme(text = element_text(family = "ArialMT"), axis.text = element_text(family = "ArialMT"), axis.title = element_text(family =
↪  "ArialMT", face = "bold", size = 22), legend.title = element_text(family = "ArialMT", size = 22), legend.text = element_text(family
↪  = "ArialMT", size = 20), legend.key.size = unit(1, "cm"))
```

Heatmaps were created with following scripts for different analyses:

```
#!/usr/bin/env bash
#File: tSNE_M1M3M6.sh created on 02.10.2018 by Claudia Kiesewetter
#This script runs the t-SNE algorithm for different macrophage populatiobs, while playing with the different parameters possible.

#Setting the environment variables

PATH_R=/misc/software/package/RBioC/3.4.3/bin

export PATH=£{PATH_R}:£{PATH}
export PATH

AnalysisDir="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6"

mkdir -p "$AnalysisDir/tsne"

### Setting colors for the different cell types ###

colCol1="deeppink"
colCol4="violetred3"
colABTcPlastic="slateblue3"
colLam="orchid1"
colMono="forestgreen"
colAdipo="khaki2"
colSero="lightsalmon2"
colTGF="darkgoldenrod3"
colHermin="brown4"
colNaCl="gold2"
colIgG="orange1"
colFib="plum3"
colMCSF="cornflowerblue"
coliDC="dodgerblue3"
colGMCSF="cadetblue3"
colNONAD="hotpink"
colABTeflon="wheat3"

### Checking number of arguments ###

if [[ $# -lt 2 ]] || [[ $# -gt 2 ]]
then
  echo "Correct Usage:"
  echo "bash tSNE_M1M3M6.sh <perlexity> <theta>"
  exit
fi

cd £{AnalysisDir}
_DATE=$(date +%s)

cat > "/loctmp/R.tSNE.M1M3M6.P$1.T$2.£{_DATE}.R" <<EOF
library(ggplot2)
library(Rtsne)
library(extrafont)

#specifying groups of samples
celltype <- factor(c(rep("Mono",3), rep("AB_Teflon", 2), rep("Adiponectin", 2), rep("Serotonin", 2), rep("TGF",2), rep("Hermin", 2),
↪  rep("NaCl", 2), rep("IgG", 2), rep("Collagen_I",3), rep("Collagen_IV",3), rep("Fibronectin", 3), rep("Laminin",3), rep("MCSF", 2),
↪  rep("AB_TcPlastic", 3), rep("iDC", 2), rep("GMCSF", 2), rep("Non-Adherent", 2)))
celltype

#specify colors used
colors = c("£colMono", "£colABTcPlastic", "£colLam", "£colFib", "£colNONAD", "£colCol1", "£colCol4", "£colABTeflon", "£colAdipo",
↪  "£colSero", "£colTGF", "£colHermin", "£colNaCl", "£colIgG", "£colMCSF", "£colGMCSF", "£coliDC")

colors_group <- c(rep("£colMono", 3), rep("£colABTeflon", 2), rep("£colAdipo", 2), rep("£colSero", 2), rep("£colTGF", 2),
↪  rep("£colHermin", 2), rep("£colNaCl", 2), rep("£colIgG", 2), rep("£colCol1", 3), rep("£colCol4", 3), rep("£colFib", 3),
↪  rep("£colLam", 3), rep("£colMCSF", 2), rep("£colABTcPlastic", 3), rep("£coliDC", 2), rep("£colGMCSF", 2), rep("£colNONAD", 2))

#save data in a numeric matrix
logcpm_rmBE <- read.delim(file = "£{AnalysisDir}/AllSamples_logcpm_rmBE.txt", sep = "\t", row.names= "X")
mydata <- data.matrix(t(logcpm_rmBE))

#Run tSNE algorithm with user specified parameters
rtsne_out <- Rtsne(mydata, check_duplicates = FALSE, pca = TRUE, perplexity= £1, theta=£2, dims=2, max_iter = 3000)
```

```
#create data table for plotting with ggplot
embedding <- as.data.frame(rtsne_out\£Y)
embedding\£Class <- as.factor(celltype)
embedding\£Color <- as.factor(colors_group)
embedding\£Class <- factor(embedding\£Class, levels = c("Mono", "AB_TcPlastic", "Laminin", "Fibronectin", "Non-Adherent", "Collagen_I",
↪  "Collagen_IV", "AB_Teflon", "Adiponectin", "Serotonin", "TGF", "Hermin", "NaCl", "IgG", "MCSF", "GMCSF", "iDC"))
embedding\£Class
embedding
#save data table in tsne folder
write.table(embedding, file = "£{AnalysisDir}/tsne/tSNE.P£1.T£2.£{_DATE}.txt", sep = "\t")

#starting ggplot:
p <- ggplot(embedding, aes(x=V1, y=V2)) +
geom_point(size=5, aes(color = embedding\£Class)) +
scale_colour_manual(values = colors, name = "Cell Type", labels = c("Monocytes", "AB-TcPlastic", "Laminin", "Fibronectin",
↪  "Non-adherent", "Collagen I", "Collagen IV", "AB-Teflon", "Adiponectin", "Serotonin", "TGF", "Hermin", "NaCl", "IgG", "M-CSF",
↪  "GM-CSF", "iDC"))  +
xlab("t-SNE 1") + ylab("t-SNE 2") +
theme_bw(base_size=20) +
theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), text = element_text(family = "ArialMT"), axis.text =
↪  element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT", face = "bold", size = 22), legend.title =
↪  element_text(family = "ArialMT", size = 22), legend.text = element_text(family = "ArialMT", size = 20), legend.key.size = unit(1,
↪  "cm"))

pdf(file = "£{AnalysisDir}/tsne/tSNE_M1M3M6.P£1.T£2.£{_DATE}.pdf", width = 12, height = 10)
plot(p, labels=TRUE)
dev.off()

EOF
chmod 750 "/loctmp/R.tSNE.M1M3M6.P$1.T$2.£{_DATE}.R"
R < /loctmp/R.tSNE.M1M3M6.P$1.T$2.£{_DATE}.R  --no-save
rm /loctmp/R.tSNE.M1M3M6.P$1.T$2.£{_DATE}.R

#+
#scale_colour_manual(values = colors, name = "Cell Type", limits = c("Mono", "AB_TcPlastic", "Laminin", "Fibronectin", "Non-Adherent",
↪  "Collagen_I", "Collagen_IV", "AB_Teflon", "Adiponectin", "Serotonin", "TGF", "Hermin", "NaCl", "IgG", "MCSF", "GMCSF", "iDC")) +
#xlab("t-SNE 1") + ylab("t-SNE 2") +
#theme_classic(base_size=20) +
#theme(text = element_text(family = "ArialMT"), axis.text = element_text(family = "ArialMT"), axis.title = element_text(family =
↪  "ArialMT", face = "bold", size = 22), legend.title = element_text(family = "ArialMT", size = 22), legend.text = element_text(family
↪  = "ArialMT", size = 20), legend.key.size = unit(1, "cm"))
```

```
#!/usr/bin/env bash
#File: Heatmaps_RNAseq_M1M3M6.sh by Claudia Kiesewetter on 02.10.2018
#Statistics for differentially expressed Genes were calculated beforehand with script M1M3M6M1M3M6_AllSamples_diffExpGenes.sh

### Setting environment variables ###

PATH_R=/misc/software/package/RBioC/3.4.3/bin

export PATH=£{PATH_R}:£{PATH}
export PATH

ANALYSISDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6"
STATDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6/stat_files"

mkdir -p $ANALYSISDIR/Heatmaps

HEATDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6/Heatmaps"

### Setting colors for the different cell types ###

colCol1="deeppink"
colCol4="violetred3"
colABTcPlastic="slateblue3"
colLam="orchid1"
colMono="forestgreen"
colAdipo="khaki2"
colSero="lightsalmon2"
colTGF="darkgoldenrod3"
colHermin="brown4"
colNaCl="gold2"
colIgG="orange1"
colFib="plum3"
colMCSF="cornflowerblue"
coliDC="dodgerblue3"
colGMCSF="cadetblue3"
colNONAD="hotpink"
colABTeflon="wheat3"
```

```
### Checking number of arguments ###

if [[ $# -lt 1 ]] || [[ $# -gt 2 ]]
then
  echo "Correct Usage:"
  echo "bash Heatmaps_RNAseq_M1M3M6.sh <logFC> <FDR cutoff>"
  echo "Prerequisite: the script M1M3M6_AllSamples_diffExpGenes.sh should be run beforehand in the same directory:"
  exit
fi

cd $ANALYSISDIR
_Date=$(date +%s)

### Starting R ###

cat > "/loctmp/R.Heatmaps.M1M3M6.£{_Date}.R" << EOF
library(ggplot2)
library(RColorBrewer)
library(ggpubr)
library(sqldf)
library(gplots)

#### Creating Heatmaps for diff exp Genes ####

### Merging Pairwise comparisons to one table, in order to create one heatmap with
### Collagen, Laminin, Fibronectin, AB

x <- read.delim("£{STATDIR}/AllSamples_ColIvsAB_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep = "\t")
subdata <- subset(x, abs(logFC) > £1 & logCPM > 1 & FDR < £2, select = c(M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV,
↪  M3_Collagen_IV, M6_Collagen_IV, M1_Laminin, M3_Laminin, M6_Laminin, M1_AB_TcPlastic, M3_AB_TcPlastic, M6_AB_TcPlastic,
↪  M1_Fibronectin, M3_Fibronectin, M6_Fibronectin))

x2 <- read.delim("£{STATDIR}/AllSamples_ColIVvsAB_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep = "\t")
subdata2 <- subset(x2, abs(logFC) > £1 & logCPM > 1 & FDR < £2, select = c(M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV,
↪  M3_Collagen_IV, M6_Collagen_IV, M1_Laminin, M3_Laminin, M6_Laminin, M1_AB_TcPlastic, M3_AB_TcPlastic, M6_AB_TcPlastic,
↪  M1_Fibronectin, M3_Fibronectin, M6_Fibronectin))

x3 <- read.delim("£{STATDIR}/AllSamples_FibvsAB_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep = "\t")
subdata3 <- subset(x3, abs(logFC) > £1 & logCPM > 1 & FDR < £2, select = c(M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV,
↪  M3_Collagen_IV, M6_Collagen_IV, M1_Laminin, M3_Laminin, M6_Laminin, M1_AB_TcPlastic, M3_AB_TcPlastic, M6_AB_TcPlastic,
↪  M1_Fibronectin, M3_Fibronectin, M6_Fibronectin))

x4 <- read.delim("£{STATDIR}/AllSamples_FibvsColI_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep = "\t")
subdata4 <- subset(x4, abs(logFC) > £1 & logCPM > 1 & FDR < £2, select = c(M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV,
↪  M3_Collagen_IV, M6_Collagen_IV, M1_Laminin, M3_Laminin, M6_Laminin, M1_AB_TcPlastic, M3_AB_TcPlastic, M6_AB_TcPlastic,
↪  M1_Fibronectin, M3_Fibronectin, M6_Fibronectin))

x5 <- read.delim("£{STATDIR}/AllSamples_LamvsColI_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep = "\t")
subdata5 <- subset(x5, abs(logFC) > £1 & logCPM > 1 & FDR < £2, select = c(M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV,
↪  M3_Collagen_IV, M6_Collagen_IV, M1_Laminin, M3_Laminin, M6_Laminin, M1_AB_TcPlastic, M3_AB_TcPlastic, M6_AB_TcPlastic,
↪  M1_Fibronectin, M3_Fibronectin, M6_Fibronectin))

x6 <- read.delim(file = "£{STATDIR}/AllSamples_LamvsAB_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep = "\t")
subdata6 <- subset(x5, abs(logFC) > £1 & logCPM > 1 & FDR < £2, select = c(M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV,
↪  M3_Collagen_IV, M6_Collagen_IV, M1_Laminin, M3_Laminin, M6_Laminin, M1_AB_TcPlastic, M3_AB_TcPlastic, M6_AB_TcPlastic,
↪  M1_Fibronectin, M3_Fibronectin, M6_Fibronectin))

x7 <- read.delim(file = "£{STATDIR}/AllSamples_FibvsLam_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep = "\t")
subdata7 <- subset(x5, abs(logFC) > £1 & logCPM > 1 & FDR < £2, select = c(M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV,
↪  M3_Collagen_IV, M6_Collagen_IV, M1_Laminin, M3_Laminin, M6_Laminin, M1_AB_TcPlastic, M3_AB_TcPlastic, M6_AB_TcPlastic,
↪  M1_Fibronectin, M3_Fibronectin, M6_Fibronectin))

comb_data <- rbind(subdata, subdata2, subdata3, subdata4, subdata5, subdata6, subdata7)
data <- unique(comb_data)
nrow(data)

data.scaled <- (scale(t(data)))
data <- data.matrix(t(data.scaled))
#Clustering
hc <- hclust(dist(t(data), method = "manhattan"), method = "ward.D2")
hr <- hclust(dist(data, method = "manhattan"), method = "ward.D2")
clustercol <- colorRampPalette(c("dodgerblue4", "white", "firebrick3"))(299)
col_breaks = c(seq(-2, -0.75, length = 100), seq(-0.74,0.74, length = 100), seq(0.75,2,length=100))
#mark clusters
mycl <- cutree(hr, h=max(hr\£height/1.5))
clustercols <- rainbow(length(unique(mycl)))
myClusterSideBar <- clustercols[mycl]
#create data table
splnames <- unlist(strsplit(as.character(rownames(t(data.scaled))), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
```

```
subdata.clustered <- cbind(t(data.scaled), row.matrix, clusterID=mycl)
clustered.data <- apply(subdata.clustered[hr\£order,],2,rev)
write.table(clustered.data, file = "£{HEATDIR}/Table_ManhattanWardZscores_AllSamples_pairwiseComparisons.logFC£1.FDR£2.txt", sep =
↪  "\t", col.names = NA)

colsidecolors <- c(rep("£colCol1", 3), rep("£colCol4", 3), rep("£colLam", 3), rep("£colABTcPlastic", 3), rep("£colFib", 3))
clustercols = c("mediumseagreen", "gold3")
myClusterSideBar <- clustercols[mycl]

#FC > 2, FDR< 0.05
#EGR2: 103, CHIT1: 265; BHLHE41: 455; MAF: 971; ETS2: 1277; C1QA: 1300

#Heatmap
pdf(file = "£{HEATDIR}/Heatmap_ColvsRest.logFC£1.FDR£2.pdf", width = 11, height = 9)
par(cex.main = 2, cex.lab = 2, cex.sub = 1.5, cex.axis = 1.5, family = "ArialMT")
heatmap.2(data, Rowv = as.dendrogram(hr), Colv = as.dendrogram(hc), col = clustercol, labRow = FALSE, na.rm = TRUE, scale = "none",
↪  margins = c(2,13), key = TRUE, density.info = "none", trace = "none", key.title = NA, key.xlab = "Z score", RowSideColors =
↪  myClusterSideBar, keysize = 1.3, ColSideColors = colsidecolors, dendrogram = "column", labCol = FALSE, breaks = col_breaks)
dev.off()

# aking mean of every condition to reduce amount of columns in the heatmap #

x <- read.delim("£{STATDIR}/AllSamples_anova2_diffExp_rmBE.txt", row.names = "ID", header = TRUE, dec = ".", sep = "\t")

colI <- rowMeans(sqldf("SELECT M1_Collagen_I, M3_Collagen_I, M6_Collagen_I FROM x"))
colIV <- rowMeans(sqldf("SELECT M1_Collagen_IV, M3_Collagen_IV, M6_Collagen_IV FROM x"))
lam <- rowMeans(sqldf("SELECT M1_Laminin, M3_Laminin, M6_Laminin FROM x"))
fib <- rowMeans(sqldf("SELECT M1_Fibronectin, M3_Fibronectin, M6_Fibronectin FROM x"))
ab_p <- rowMeans(sqldf("SELECT M1_AB_TcPlastic, M3_AB_TcPlastic, M6_AB_TcPlastic FROM x"))
ab <- rowMeans(sqldf("SELECT M1_AB_Teflon, M3_AB_Teflon FROM x"))
tgf <- rowMeans(sqldf("SELECT M1_TGF, M3_TGF FROM x"))
nacl <- rowMeans(sqldf("SELECT M1_NaCl, M3_NaCl FROM x"))
sero <- rowMeans(sqldf("SELECT M1_Serotonin, M3_Serotonin FROM x"))
adipo <- rowMeans(sqldf("SELECT M1_Adiponectin, M3_Adiponectin FROM x"))
heme <- rowMeans(sqldf("SELECT M1_Hermin, M3_Hermin FROM x"))
igg <- rowMeans(sqldf("SELECT M1_IgG, M3_IgG FROM x"))
mcsf <- rowMeans(sqldf("SELECT M1_MCSF, M3_MCSF FROM x"))
gmcsf <- rowMeans(sqldf("SELECT M1_GMCSF, M3_GMCSF FROM x"))
idc <- rowMeans(sqldf("SELECT M1_iDC, M3_iDC FROM x"))
roller <- rowMeans(sqldf("SELECT M1_Rollerflaschen, M3_Rollerflaschen FROM x"))

x2 <- cbind(x, colI, colIV, lam, fib, ab_p, ab, tgf, nacl, sero, adipo, heme, igg, mcsf, gmcsf, idc, roller)
colnames(x2) <- c(colnames(x), "Collagen_I", "Collagen_IV", "Laminin", "Fibronectin", "AB_TcPlastic", "AB_Teflon", "TGF", "NaCl",
↪  "Serotonin", "Adiponectin", "Heme", "IgG", "MCSF", "GMCSF", "iDC", "Non_Adherent")

subdata <- subset(x2, abs(logFC.ColvsCTRL) > £1 & abs(logFC.ColvswoMCSF) > £1 & logCPM > 1 & FDR < £2, select = c(Collagen_I,
↪  Collagen_IV, Laminin, Fibronectin, AB_TcPlastic, AB_Teflon, TGF, NaCl, Serotonin, Adiponectin, Heme, IgG, MCSF, GMCSF, iDC,
↪  Non_Adherent))
data.scaled <- (scale(t(subdata)))
data <- data.matrix(t(data.scaled))

#Clustering
hc <- hclust(dist(t(data), method = "manhattan"), method = "ward.D2")
hr <- hclust(dist(data, method = "manhattan"), method = "ward.D2")
clustercol <- colorRampPalette(c("dodgerblue4", "white", "firebrick3"))(299)
col_breaks = c(seq(-2, -0.75, length = 100), seq(-0.74,0.74, length = 100), seq(0.75,2,length=100))
#mark clusters
mycl <- cutree(hr, h=max(hr\£height/1.5))
clustercols <- rainbow(length(unique(mycl)))
myClusterSideBar <- clustercols[mycl]
#create data table
splnames <- unlist(strsplit(as.character(rownames(t(data.scaled))), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
subdata.clustered <- cbind(t(data.scaled), row.matrix, clusterID=mycl)
clustered.data <- apply(subdata.clustered[hr\£order,],2,rev)
write.table(clustered.data, file = "£{HEATDIR}/Table_ManhattanWardZscores_anova2.logFC£1.FDR£2.txt", sep = "\t", col.names = NA)

#pdf(file = "£{HEATDIR}/Heatmap_Anova2_meanAllSamples.FC£1.FDR£2.pdf", width = 9, height = 8)
#par(cex.main = 2, cex.lab = 2, cex.sub = 1.5, cex.axis = 1.5, family = "ArialMT")
#heatmap.2(data, Rowv = as.dendrogram(hr), Colv = as.dendrogram(hc), col = clustercol,
          #labRow = FALSE,
          #na.rm = TRUE, scale = "none", margins = c(12,1),
          #key = TRUE, density.info = "none", trace = "none", key.title = NA, key.xlab = "Z score",
          #RowSideColors = myClusterSideBar,
          #srtCol = 45, cexRow = 2, cexCol = 2, keysize = 1.5, dendrogram = "both", breaks = #col_breaks)
#dev.off()

#### Heatmap with Genes with highes variance ####

x <- read.delim(file = "£ANALYSISDIR/AllSamples_logcpm_rmBE.txt", sep = "\t", row.names = "X")
head(x)
```

```
colI <- rowMeans(sqldf("SELECT M1_Collagen_I, M3_Collagen_I, M6_Collagen_I FROM x"))
colIV <- rowMeans(sqldf("SELECT M1_Collagen_IV, M3_Collagen_IV, M6_Collagen_IV FROM x"))
lam <- rowMeans(sqldf("SELECT M1_Laminin, M3_Laminin, M6_Laminin FROM x"))
fib <- rowMeans(sqldf("SELECT M1_Fibronectin, M3_Fibronectin, M6_Fibronectin FROM x"))
ab_p <- rowMeans(sqldf("SELECT M1_AB_TcPlastic, M3_AB_TcPlastic, M6_AB_TcPlastic FROM x"))
ab <- rowMeans(sqldf("SELECT M1_AB_Teflon, M3_AB_Teflon FROM x"))
tgf <- rowMeans(sqldf("SELECT M1_TGF, M3_TGF FROM x"))
nacl <- rowMeans(sqldf("SELECT M1_NaCl, M3_NaCl FROM x"))
sero <- rowMeans(sqldf("SELECT M1_Serotonin, M3_Serotonin FROM x"))
adipo <- rowMeans(sqldf("SELECT M1_Adiponectin, M3_Adiponectin FROM x"))
heme <- rowMeans(sqldf("SELECT M1_Hermin, M3_Hermin FROM x"))
igg <- rowMeans(sqldf("SELECT M1_IgG, M3_IgG FROM x"))
mcsf <- rowMeans(sqldf("SELECT M1_MCSF, M3_MCSF FROM x"))
gmcsf <- rowMeans(sqldf("SELECT M1_GMCSF, M3_GMCSF FROM x"))
idc <- rowMeans(sqldf("SELECT M1_iDC, M3_iDC FROM x"))
roller <- rowMeans(sqldf("SELECT M1_Rollerflaschen, M3_Rollerflaschen FROM x"))

x2 <- cbind(colI, colIV, lam, fib, ab_p, ab, tgf, nacl, sero, adipo, heme, igg, mcsf, gmcsf, idc, roller)
colnames(x2) <- c("Collagen_I", "Collagen_IV", "Laminin", "Fibronectin", "AB_TcPlastic", "AB_Teflon", "TGF", "NaCl", "Serotonin",
↪  "Adiponectin", "Heme", "IgG", "MCSF", "GMCSF", "iDC", "Non_Adherent")
rownames(x2) <- rownames(x)

#Calculate row Variance

rowVar <- rowVars(x2)

#sort by variation and select top genes

idx<- order(-rowVar)[1:750]
x3 <- x2[idx,]

data.scaled <- (scale(t(x3)))
data <- data.matrix(t(data.scaled))

#Clustering
hc <- hclust(dist(t(data), method = "manhattan"), method = "complete")
hr <- hclust(dist(data, method = "manhattan"), method = "complete")
clustercol <- colorRampPalette(c("dodgerblue4", "white", "firebrick3"))(299)

#Gene Labelling
splnames <- unlist(strsplit(as.character(rownames(x3)), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
data_names <- cbind(x3, row.matrix)

colors <- c("£colCol1", "£colCol4", "£colLam", "£colFib", "£colABTcPlastic", "£colABTeflon", "£colTGF", "£colNaCl", "£colSero",
↪  "£colAdipo", "£colHermin", "£colIgG", "£colMCSF", "£colGMCSF", "£coliDC", "£colNONAD")
column_colors <- c(rep("£colABTcPlastic", 5), rep("£colABTeflon", 7), rep("£colABTcPlastic", 4))

pdf(file = "£HEATDIR/Heatmap_top750_genes_AllSamples.pdf", width = 7, height = 7)
par(family = "ArialMT", cex.main = 2.5, cex.lab = 2, cex.sub = 2, cex.axis = 1.5, lend = 1)
heatmap.2(data, Rowv = as.dendrogram(hr), Colv = as.dendrogram(hc), col = clustercol, labRow = FALSE, na.rm = TRUE, scale = "none",
↪  margins = c(12,1), key = TRUE, density.info = "none", trace = "none", key.title = NA, key.xlab = "Z score", srtCol = 45, cexRow =
↪  2, cexCol = 2, keysize = 1.5, dendrogram = "column", ColSideColors = column_colors, colCol = colors)
dev.off()

EOF
chmod 750 "/loctmp/R.Heatmaps.M1M3M6.£{_Date}.R"
R < /loctmp/R.Heatmaps.M1M3M6.£{_Date}.R --no-save
rm /loctmp/R.Heatmaps.M1M3M6.£{_Date}.R
```

```
#!usr/bin/env bash
#File: Heatmaps_HC.sh by Claudia K on October 08 2018
#This script follows the script HC_DiffExpAnalysis.sh, and was written to plot Heatmaps for the anova like analysis done in the
↪  previous script.

#setting up environment variables

DIR_PKG="/misc/software/ngs"

PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin

export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH

ANALYSISDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/HofbauerCells"
TMPDIR="/loctmp"
```

```
mkdir -p $ANALYSISDIR/Heatmaps
HEATDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/HofbauerCells/Heatmaps"

### Setting colors for the different cell types ###

colHC="darkorchid4"
colCol1="deeppink"
colCol4="violetred3"
colBMM="darkgoldenrod3"
colAB="slateblue3"
colLam="orchid1"
colMono="forestgreen"

cat > "/loctmp/R.Heatmaps.HC.£{_Date}.R" << EOF

## Choosing FDR < 0.05 abs(logFC.HCvsCol) > 0 & abs(logFC.HCvsBMM) > 1 & abs(logFC.ColvsBMM) > 1 ##

#Data preprocessing
x <- read.delim(file = "£{ANALYSISDIR}/HCBMMColLamABMono_anova2_diffExp_rmBE.txt", row.names = "ID", sep = "\t")
subdata <- subset(x, logCPM > 1 & FDR < 0.05 & abs(logFC.HCvsCol) > 0 & abs(logFC.HCvsBMM) > 1 & abs(logFC.ColvsBMM) > 1, select =
↪  c(M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV, M3_Collagen_IV, M6_Collagen_IV, HC_6, HC_7, HC_8, BMM_EM1A, BMM_GS,
↪  BMM_142, BMM_143, BMM_144, BMM_145, BMM_146))
data.scaled <- (scale(t(subdata)))
data <- data.matrix(t(data.scaled))
#Clustering
hc <- hclust(dist(t(data), method = "manhattan"), method = "ward.D2")
hr <- hclust(dist(data, method = "manhattan"), method = "ward.D2")
clustercol <- colorRampPalette(c("navy", "white", "firebrick3"))(299)
col_breaks = c(seq(-2.5, -0.75, length = 100), seq(-0.74,0.74, length = 100), seq(0.75,2.5,length=100))
#mark clusters
mycl <- cutree(hr, h=max(hr\£height/4))
clustercols <- brewer.pal(4, "Set2")
myClusterSideBar <- clustercols[mycl]
#create data table
splnames <- unlist(strsplit(as.character(rownames(t(data.scaled))), "[£]"))
row.matrix <- matrix(splnames, ncol = 4, byrow = TRUE)
colnames(row.matrix) <- c("EnsembleID", "GeneSymbol", "TranscriptLength", "GeneType")
subdata.clustered <- cbind(t(data.scaled), row.matrix, clusterID=mycl)
clustered.data <- apply(subdata.clustered[hr\£order,],2,rev)
write.table(clustered.data, file = "£{HEATDIR}/Table_ManhattanWardZscores_anova_fdr.0.05.FCHCvsCol.0.FCHCvsBMM.1.FCColvsBMM.1.txt", sep
↪  = "\t", col.names = NA)

colors_group <- c(rep("£colCol1", 6), rep("£colHC",3), rep("£colBMM",7))

pdf(file = "£{HEATDIR}/Heatmap_anova_fdr.0.05.FCHCvsCol.0.FCHCvsBMM.1.FCColvsBMM.1.pdf", width = 10, height = 8)
par(cex.main = 1.5, cex.lab = 2, cex.sub = 2.5, cex.axis = 1.75, lend = 1, family = "ArialMT")
heatmap.2(data, Rowv = as.dendrogram(hr), Colv = as.dendrogram(hc), col = clustercol,
          labRow = FALSE, labCol = FALSE,
          na.rm = TRUE, scale = "none", margins = c(1,5),
          key = TRUE, density.info = "none", trace = "none", key.title = NA, key.xlab = "Z score",
          RowSideColors = myClusterSideBar, ColSideColors = colors_group,
          srtCol = 45, cexRow = 2, cexCol = 2, keysize = 1.5, dendrogram = "column", breaks = col_breaks)
legend("topright", legend = c("Hofbauer", "Collagen", "BMM"),
        col = c("£colHC", "£colCol1", "£colBMM"),
        lty = 1, lwd = 10, bty = "n", ncol = 1, cex = 1.25)
dev.off()

## Saving gene symbols from different clusters in text files ##

data <- read.delim(file = "£{HEATDIR}/Table_ManhattanWardZscores_anova_fdr.0.05.FCHCvsCol.0.FCHCvsBMM.1.FCColvsBMM.1.txt", sep = "\t")
cluster1 <- sqldf("SELECT M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV, M3_Collagen_IV, M6_Collagen_IV, HC_6, HC_7,
↪  HC_8, BMM_EM1A, BMM_GS, BMM_142, BMM_143, BMM_144, BMM_145, BMM_146, GeneSymbol FROM data WHERE ClusterID = '1'")
cluster2 <- sqldf("SELECT M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV, M3_Collagen_IV, M6_Collagen_IV, HC_6, HC_7,
↪  HC_8, BMM_EM1A, BMM_GS, BMM_142, BMM_143, BMM_144, BMM_145, BMM_146, GeneSymbol FROM data WHERE ClusterID = '2'")
cluster3 <- sqldf("SELECT M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV, M3_Collagen_IV, M6_Collagen_IV, HC_6, HC_7,
↪  HC_8, BMM_EM1A, BMM_GS, BMM_142, BMM_143, BMM_144, BMM_145, BMM_146, GeneSymbol FROM data WHERE ClusterID = '3'")
cluster4 <- sqldf("SELECT M1_Collagen_I, M3_Collagen_I, M6_Collagen_I, M1_Collagen_IV, M3_Collagen_IV, M6_Collagen_IV, HC_6, HC_7,
↪  HC_8, BMM_EM1A, BMM_GS, BMM_142, BMM_143, BMM_144, BMM_145, BMM_146, GeneSymbol FROM data WHERE ClusterID = '4'")

write.table(cluster1, file = "£{HEATDIR}/Heatmap_anova_fdr_0.05.FCHCvsCol.0.FCHCvsBMM.1.FCColvsBMM.1_Cluster1_GeneSymbols.txt", sep =
↪  "\t", quote = FALSE, row.names = FALSE)
write.table(cluster2, file = "£{HEATDIR}/Heatmap_anova_fdr_0.05.FCHCvsCol.0.FCHCvsBMM.1.FCColvsBMM.1_Cluster2_GeneSymbols.txt", sep =
↪  "\t", quote = FALSE, row.names = FALSE)
write.table(cluster3, file = "£{HEATDIR}/Heatmap_anova_fdr_0.05.FCHCvsCol.0.FCHCvsBMM.1.FCColvsBMM.1_Cluster3_GeneSymbols.txt", sep =
↪  "\t", quote = FALSE, row.names = FALSE)
write.table(cluster4, file = "£{HEATDIR}/Heatmap_anova_fdr_0.05.FCHCvsCol.0.FCHCvsBMM.1.FCColvsBMM.1_Cluster4_GeneSymbols.txt", sep =
↪  "\t", quote = FALSE, row.names = FALSE)

EOF
chmod 750 "/loctmp/R.Heatmaps.HC.£{_Date}.R"
R < /loctmp/R.Heatmaps.HC.£{_Date}.R --no-save
```

```
rm /loctmp/R.Heatmaps.HC.£{_Date}.R
```

This script contains the code for dot plots of gene expression data:

```
#!/usr/bin/env bash
#File: Dotplot for more than one gene

#setting up environment variables

DIR_PKG="/misc/software/ngs"
PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin
export PATH=£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH

ANALYSISDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6"
ANALYSISDIR2="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/EGR2KD"

### Setting colors for the different cell types ###

colCol1="deeppink"
colCol4="violetred3"
colABTcPlastic="slateblue3"
colLam="orchid1"
colMono="forestgreen"
colAdipo="khaki2"
colSero="lightsalmon2"
colTGF="darkgoldenrod3"
colHermin="brown4"
colNaCl="gold2"
colIgG="orange1"
colFib="plum3"
colMCSF="cornflowerblue"
coliDC="dodgerblue3"
colGMCSF="cadetblue3"
colNONAD="hotpink"
colABTeflon="wheat3"
colCTRL="dodgerblue3"
colMock="royalblue2"
col1132="gold2"
col2665="goldenrod3"

### Starting analysis in R ###

cat > "/loctmp/R.ETS.M1M3M6.£{_Date}.R" << EOF
library(ggplot2)
library(sqldf)
library(ggpubr)
library(extrafont)
library(RColorBrewer)
library(reshape2)

data <- read.delim(file = "£{ANALYSISDIR}/M1M3M6_logcpm_rmBE_data_dotplots.txt", sep = "\t", row.names = "X")

### create rpkm values ###

data\£RPKM_M1Col1 <- ((2^data\£M1_Collagen_I) / data\£TranscriptLength) * 1000
data\£RPKM_M3Col1 <- ((2^data\£M3_Collagen_I) / data\£TranscriptLength) * 1000
data\£RPKM_M6Col1 <- ((2^data\£M6_Collagen_I) / data\£TranscriptLength) * 1000
data\£RPKM_M1Col4 <- ((2^data\£M1_Collagen_IV) / data\£TranscriptLength) * 1000
data\£RPKM_M3Col4 <- ((2^data\£M3_Collagen_IV)/ data\£TranscriptLength) * 1000
data\£RPKM_M6Col4 <- ((2^data\£M6_Collagen_IV) / data\£TranscriptLength) * 1000
data\£RPKM_M1AB <- ((2^data\£M1_AB_TcPlastic) / data\£TranscriptLength) * 1000
data\£RPKM_M3AB <- ((2^data\£M3_AB_TcPlastic) / data\£TranscriptLength) * 1000
data\£RPKM_M6AB <- ((2^data\£M6_AB_TcPlastic) / data\£TranscriptLength) * 1000
data\£RPKM_M1Lam <- ((2^data\£M1_Laminin) / data\£TranscriptLength) * 1000
data\£RPKM_M3Lam <- ((2^data\£M3_Laminin) / data\£TranscriptLength) * 1000
data\£RPKM_M6Lam <- ((2^data\£M6_Laminin) / data\£TranscriptLength) * 1000
data\£RPKM_M1Fib <- ((2^data\£M1_Fibronectin) / data\£TranscriptLength) * 1000
data\£RPKM_M3Fib <- ((2^data\£M3_Fibronectin) / data\£TranscriptLength) * 1000
data\£RPKM_M6Fib <- ((2^data\£M6_Fibronectin) / data\£TranscriptLength) * 1000
data\£RPKM_M1Mono <- ((2^data\£M1_Mono) / data\£TranscriptLength) * 1000
data\£RPKM_M3Mono <- ((2^data\£M3_Mono) / data\£TranscriptLength) * 1000
data\£RPKM_M6Mono <- ((2^data\£M6_Mono) / data\£TranscriptLength) * 1000

### Plot for RUNX transcription factors ###
```

```
RUNX=c("RUNX1", "RUNX2", "RUNX3", "CBFB")
runx <- data.frame(RUNX)
colnames(runx) <- c("runx")
runx

data2 <- sqldf("SELECT GeneSymbol, RPKM_M1Col1, RPKM_M3Col1, RPKM_M6Col1, RPKM_M1Col4, RPKM_M3Col4, RPKM_M6Col4, RPKM_M1AB, RPKM_M3AB,
↪  RPKM_M6AB, RPKM_M1Lam, RPKM_M3Lam, RPKM_M6Lam, RPKM_M1Fib, RPKM_M3Fib, RPKM_M6Fib FROM data INNER JOIN runx ON GeneSymbol = runx")

data3 <- melt(data2)
data3
data3\£Type <- c(rep("Collagen I", 12), rep("Collagen IV", 12), rep("AB", 12), rep("Laminin", 12), rep("Fibronectin", 12))


p1 <- ggplot(data3) + theme_bw() +
        geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£colABTcPlastic", "£colCol1", "£colCol4", "£colFib", "£colLam"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR/RUNX_expression.pdf", width = 6, height = 4, useDingbats=FALSE)
p1
dev.off()

## Plot genes regulated by Collagen

col <- data.frame(c("EGR2", "BHLHE41", "CHIT1", "CHI3L1", "ITGB3", "MMP7", "CSF1", "FN1"))
colnames(col) <- c("col")
col

data3 <- sqldf("SELECT GeneSymbol, RPKM_M1Col1, RPKM_M3Col1, RPKM_M6Col1, RPKM_M1Col4, RPKM_M3Col4, RPKM_M6Col4, RPKM_M1AB, RPKM_M3AB,
↪  RPKM_M6AB, RPKM_M1Lam, RPKM_M3Lam, RPKM_M6Lam, RPKM_M1Fib, RPKM_M3Fib, RPKM_M6Fib, RPKM_M1Mono, RPKM_M3Mono, RPKM_M6Mono FROM data
↪  INNER JOIN col ON GeneSymbol = col")

md <- melt(data3)
md
md\£Type <- c(rep("Collagen I", 24), rep("Collagen IV", 24), rep("AB", 24), rep("Laminin", 24), rep("Fibronectin", 24), rep("Mono",
↪  24))
md\£GeneSymbol <- factor(md\£GeneSymbol, levels = c("EGR2", "BHLHE41", "CHIT1", "CHI3L1", "ITGB3", "MMP7", "CSF1", "FN1"))

p1 <- ggplot(md) + theme_bw() +
        geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£colABTcPlastic", "£colCol1", "£colCol4", "£colFib", "£colLam", "£colMono"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR/Col_expression_down.pdf", width = 9, height = 4, useDingbats=FALSE)
p1
dev.off()

## Plot genes up-regulated by Collagen

col <- data.frame(c("MAF", "ETS2", "IRF8", "IRF4", "C1QA", "CD163", "CSF1R", "NFATC2"))
colnames(col) <- c("col")
col

data3 <- sqldf("SELECT GeneSymbol, RPKM_M1Col1, RPKM_M3Col1, RPKM_M6Col1, RPKM_M1Col4, RPKM_M3Col4, RPKM_M6Col4, RPKM_M1AB, RPKM_M3AB,
↪  RPKM_M6AB, RPKM_M1Lam, RPKM_M3Lam, RPKM_M6Lam, RPKM_M1Fib, RPKM_M3Fib, RPKM_M6Fib, RPKM_M1Mono, RPKM_M3Mono, RPKM_M6Mono FROM data
↪  INNER JOIN col ON GeneSymbol = col")

md <- melt(data3)
md\£Type <- c(rep("Collagen I", 24), rep("Collagen IV", 24), rep("AB", 24), rep("Laminin", 24), rep("Fibronectin", 24), rep("Mono",
↪  24))
md\£GeneSymbol <- factor(md\£GeneSymbol, levels = c("MAF", "ETS2", "IRF8", "IRF4", "C1QA", "CD163", "CSF1R", "NFATC2"))

p1 <- ggplot(md) + theme_bw() +
        geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£colABTcPlastic", "£colCol1", "£colCol4", "£colFib", "£colLam", "£colMono"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR/Col_expression_up.pdf", width = 9, height = 4, useDingbats=FALSE)
p1
dev.off()

### Expression Level of Monocytes and Macrophages over differnt Integrins, Metalloproteinases###
```

```
mmp <- data.frame(c("MMP2", "MMP7", "MMP8", "MMP9", "MMP14"))
colnames(mmp) <- c("mmp")
mmp

data2 <- sqldf("SELECT GeneSymbol, RPKM_M1Col1, RPKM_M3Col1, RPKM_M6Col1, RPKM_M1Col4, RPKM_M3Col4, RPKM_M6Col4, RPKM_M1AB, RPKM_M3AB,
↪  RPKM_M6AB, RPKM_M1Lam, RPKM_M3Lam, RPKM_M6Lam, RPKM_M1Fib, RPKM_M3Fib, RPKM_M6Fib, RPKM_M1Mono, RPKM_M3Mono, RPKM_M6Mono FROM data
↪  INNER JOIN mmp ON GeneSymbol = mmp")

md <- melt(data2)
md
md\£Type <- c(rep("Collagen I", 15), rep("Collagen IV",15), rep("AB", 15), rep("Laminin", 15), rep("Fibronectin", 15), rep("Mono", 15))


p1 <- ggplot(md) + theme_bw() +
         geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£colABTcPlastic", "£colCol1", "£colCol4", "£colFib", "£colLam", "£colMono"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR/MMP_Mono_expression.pdf", width = 10, height = 4)
p1
dev.off()

itg <- data.frame(c("ITGA1", "ITGA2", "ITGA3", "ITGA4", "ITGA5", "ITGA6", "ITGA7", "ITGA8", "ITGAL", "ITGAX", "ITGB1", "ITGB2",
↪  "ITGB3"))
colnames(itg) <- c("itg")
itg

data2 <- sqldf("SELECT GeneSymbol, RPKM_M1Col1, RPKM_M3Col1, RPKM_M6Col1, RPKM_M1Col4, RPKM_M3Col4, RPKM_M6Col4, RPKM_M1AB, RPKM_M3AB,
↪  RPKM_M6AB, RPKM_M1Lam, RPKM_M3Lam, RPKM_M6Lam, RPKM_M1Fib, RPKM_M3Fib, RPKM_M6Fib, RPKM_M1Mono, RPKM_M3Mono, RPKM_M6Mono FROM data
↪  INNER JOIN itg ON GeneSymbol = itg")
data2

md <- melt(data2)
md
md\£Type <- c(rep("Collagen I", 36), rep("Collagen IV",36), rep("AB", 36), rep("Laminin", 36), rep("Fibronectin", 36), rep("Mono", 36))

p1 <- ggplot(md) + theme_bw() +
         geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£colABTcPlastic", "£colCol1", "£colCol4", "£colFib", "£colLam", "£colMono"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR/ITG_Mono_expression.pdf", width = 10, height = 4)
p1
dev.off()

data <- read.delim(file = "£{ANALYSISDIR2}/Logcpm_rmBE_data_dotplots.txt", sep = "\t", row.names = "X")

#calculate RPKM values
data\£RPKM_Mock_1 <- ((2^data\£exp129_Mock) / data\£TranscriptLength) * 1000
data\£RPKM_Mock_2 <- ((2^data\£exp130_Mock) / data\£TranscriptLength) * 1000
data\£RPKM_Mock_3 <- ((2^data\£exp132_Mock) / data\£TranscriptLength) * 1000
data\£RPKM_CTRL_1 <- ((2^data\£exp129_CTRL) / data\£TranscriptLength) * 1000
data\£RPKM_CTRL_2 <- ((2^data\£exp130_CTRL)/ data\£TranscriptLength) * 1000
data\£RPKM_CTRL_3 <- ((2^data\£exp132_CTRL) / data\£TranscriptLength) * 1000
data\£RPKM_1132_1 <- ((2^data\£exp129_1132) / data\£TranscriptLength) * 1000
data\£RPKM_1132_2 <- ((2^data\£exp130_1132) / data\£TranscriptLength) * 1000
data\£RPKM_1132_3 <- ((2^data\£exp132_1132) / data\£TranscriptLength) * 1000
data\£RPKM_2665_1 <- ((2^data\£exp129_2665) / data\£TranscriptLength) * 1000
data\£RPKM_2665_2 <- ((2^data\£exp130_2665) / data\£TranscriptLength) * 1000
data\£RPKM_2665_3 <- ((2^data\£exp132_2665) / data\£TranscriptLength) * 1000

KD <- data.frame(c("EGR2", "CHIT1", "ITGA3", "CCL22"))
colnames(KD) <- c("KD")
KD

d <- sqldf("SELECT GeneSymbol, RPKM_Mock_1, RPKM_Mock_2, RPKM_Mock_3, RPKM_CTRL_1, RPKM_CTRL_2, RPKM_CTRL_3, RPKM_1132_1, RPKM_1132_2,
↪  RPKM_1132_3, RPKM_2665_1, RPKM_2665_2, RPKM_2665_3 FROM data INNER JOIN KD ON GeneSymbol = KD")

md <- melt(d)
md\£Type <- c(rep("Mock", 12), rep("CTRL", 12), rep("siRNA-1", 12), rep("siRNA-2", 12))

p1 <- ggplot(md) + theme_bw() +
         geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£colMock", "£colCTRL", "£col1132", "£col2665"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR2/Expression_plot.pdf", width = 8, height = 4)
p1
dev.off()

EOF
chmod 750 "/loctmp/R.ETS.M1M3M6.£{_Date}.R"
R < /loctmp/R.ETS.M1M3M6.£{_Date}.R --no-save
rm /loctmp/R.ETS.M1M3M6.£{_Date}.R

### Dotplots after iDC GMCSF MCSF anova analysis ###
cat > "/loctmp/Dotplots_iDCGMCSFMCSF.R" << EOF

library(ggplot2)
library(sqldf)
library(ggpubr)
library(extrafont)
library(RColorBrewer)
library(reshape2)

data <- read.delim(file = "£{ANALYSISDIR}/M1M3M6_logcpm_rmBE_data_dotplots.txt", sep = "\t", row.names = "X")

GeneList <- c("CD163", "IRF4", "OAS2", "CD209", "CD1B", "ISG15", "IL1RN", "CD14", "HIF1A", "TLR7", "CEBPB", "EHF", "FOXO3")

Genes <- data.frame(GeneList)
Genes

### create rpkm values ###

data\£RPKM_M1iDC <- ((2^data\£M1_iDC) / data\£TranscriptLength) * 1000
data\£RPKM_M3iDC <- ((2^data\£M3_iDC) / data\£TranscriptLength) * 1000
data\£RPKM_M1GMCSF <- ((2^data\£M1_GMCSF) / data\£TranscriptLength) * 1000
data\£RPKM_M3GMCSF <- ((2^data\£M3_GMCSF) / data\£TranscriptLength) * 1000
data\£RPKM_M1MCSF <- ((2^data\£M1_MCSF) / data\£TranscriptLength) * 1000
data\£RPKM_M3MCSF <- ((2^data\£M3_MCSF) / data\£TranscriptLength) * 1000

d <- sqldf("SELECT GeneSymbol, RPKM_M1iDC, RPKM_M3iDC, RPKM_M1GMCSF, RPKM_M3GMCSF, RPKM_M1MCSF, RPKM_M3MCSF FROM data INNER JOIN Genes
↪  ON GeneSymbol = GeneList")

d

md <- melt(d)
md\£Type <- c(rep("iDC", 26), rep("GMCSF", 26), rep("MCSF", 26))
md\£GeneSymbol <- factor(md\£GeneSymbol, levels = c("ISG15", "OAS2", "IL1RN", "CEBPB", "TLR7", "CD14", "HIF1A", "CD163", "FOXO3",
↪  "CD1B", "CD209", "IRF4", "EHF"))

md

p1 <- ggplot(md) + theme_bw() +
        geom_point(aes(x = GeneSymbol, y =value, colour = Type), position = position_dodge(width = 0.5), size = 2) + scale_y_log10() +
  scale_colour_manual(values = c("£coliDC", "£colGMCSF", "£colMCSF"), name = "") +
  labs(x = " ", y = "RPKM") +
  theme(axis.text.x = element_text(family = "ArialMT", colour = "black", size = 14), axis.title = element_text(family = "ArialMT", size
↪  = 16, colour ="black"), panel.grid.minor = element_blank(),legend.title = element_text(family = "ArialMT", size = 16), legend.text
↪  = element_text(family = "ArialMT", size = 14), legend.key.size = unit(0.5, "cm"))

pdf(file = "£ANALYSISDIR/Dotplot_iDCGMCSFMCSF.pdf", width = 10, height = 4, useDingbats = FALSE)
p1
dev.off()


EOF
chmod 750 "/loctmp/Dotplots_iDCGMCSFMCSF.R"
R < /loctmp/Dotplots_iDCGMCSFMCSF.R --no-save
rm /loctmp/Dotplots_iDCGMCSFMCSF.R
```

Plots for gene ontology analysis:

```
#Trying to plot own GO barplot

setwd("~/Documents/Doktorarbeit/RNAseq_AllSamples")
library(ggplot2)
library(ggpubr)

down <- read.delim("~/Documents/Doktorarbeit/RNAseq_AllSamples/Metascape_ColDown_ColvsRest/Enrichment_GO/GO_AllLists.csv", sep = ",")
```

```
d <- down[,c(1:7,16)]

head(d, n=30)

data <- d[c(1,3,5,7,17),]
data$Logq <- -data$Log.q.value.

## Start plot ##

#set order of terms

data$Description <- factor(data$Description, levels = c("chemotaxis", "myeloid leukocyte mediated immunity", "regulated exocytosis",
↪  "myeloid leukocyte activation", "extracellular matrix organization"))

p1 <- ggplot(data) + theme_bw() +
        geom_bar(aes(x=Description, y=Logq, fill = Logq), stat="identity") +
        coord_flip() +
        labs(x = "", y = "log(qValue)") +
        scale_x_discrete(labels = c("Chemotaxis", "Myeloid Mediated Immunity", "Exocytosis", "Myeloid Leukocyte Activation",
        ↪  "Extracellular Matrix Organization")) +
        scale_fill_gradient2(,low="grey70", high="dodgerblue4", name = "log(qValue)")+
                theme( panel.grid.minor = element_blank(), axis.text.x = element_text(family = "ArialMT", colour = "black", size =
                ↪  16), axis.title = element_text(family = "ArialMT", size = 20), axis.text.y = element_text(size = 18),
                ↪  legend.title = element_text(family = "ArialMT", size = 22), legend.text = element_text(family = "ArialMT", size =
                ↪  20), legend.key.size = unit(1, "cm"), plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), "cm"), panel.spacing =
                ↪  unit(c(0.25, 1, 0.25, 4), "cm"))
#top, right, bottom, left

pdf("GeneOntology_colvsRest_down.pdf", width = 12, height = 4)
plot(p1)
dev.off()

## Performing the same plot for Collagen up-regulated Genes ##

up <- read.delim("~/Documents/Doktorarbeit/RNAseq_AllSamples/Metascape_ColUP_ColvsRest/Enrichment_GO/GO_AllLists.csv", sep = ",")

d <- up[,c(1:7,16)]

data <- d[c(2,3,26,38,85),]
data$Logq <- -data$Log.q.value.

data$Description <- factor(data$Description, levels = data$Description[order(data$Logq)])

p1 <- ggplot(data) + theme_bw() +
        geom_bar(aes(x=Description, y=Logq, fill = Logq), stat="identity") +
        coord_flip() +
        labs(x = "", y = "log(qValue)") +
        scale_x_discrete(labels=c("Endocytosis", "Hemostasis", "Nucleosome Assembly", "Inflammatory Response", "Cell Cycle")) +
        scale_fill_gradient2(,low="grey70", high="firebrick4", name = "log(qValue)") +
        theme( panel.grid.minor = element_blank(), axis.text.x = element_text(family = "ArialMT", colour = "black", size = 16),
        ↪  axis.title = element_text(family = "ArialMT", size = 20), axis.text.y = element_text(size = 18), legend.title =
        ↪  element_text(family = "ArialMT", size = 22), legend.text = element_text(family = "ArialMT", size = 20), legend.key.size =
        ↪  unit(1, "cm"), plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), "cm"), panel.spacing = unit(c(0.25, 1, 0.25, 4), "cm"))


pdf("GeneOntology_colvsRest_up.pdf", width = 10.45, height = 4)
plot(p1)
dev.off()

## GO Analysis for Genes down-regulated by Collagen and EGR2 KD ##

egr2 <- read.delim("~/Documents/Doktorarbeit/EGR2KD/Metascape_EGR2KD_Down_Col_Down/Enrichment_GO/GO_AllLists.csv", sep = ",")

d <- egr2[,c(1:7,16)]

data <- d[c(1,4,8,16,28),]
data$Logq <- -data$Log.q.value.

data$Description <- factor(data$Description, levels = data$Description[order(data$Logq)])

p1 <- ggplot(data) + theme_bw() +
        geom_bar(aes(x=Description, y=Logq, fill = Logq), stat="identity") +
        coord_flip() +
        labs(x = "", y = "log(qValue)") +
        scale_x_discrete(labels = c("Focal Adhesion", "System Development", "Leukocyte Migration", "Cell Morphogenesis",
        ↪  "Extracellular Structure Organization"))+
        scale_fill_gradient2(,low="grey70", high="dodgerblue4", name = "log(qValue)") +
        theme( panel.grid.minor = element_blank(), axis.text.x = element_text(family = "ArialMT", colour = "black", size = 16),
        ↪  axis.title = element_text(family = "ArialMT", size = 20), axis.text.y = element_text(size = 18), legend.title =
        ↪  element_text(family = "ArialMT", size = 22), legend.text = element_text(family = "ArialMT", size = 20), legend.key.size =
        ↪  unit(1, "cm"))
```

```
pdf("GeneOntology_colvsRest_down_egr2kd_down.pdf", width = 12, height = 4)
plot(p1)
dev.off()
```

ATAC-seq and ChIPseq analysis:

```
#! /usr/bin/env bash
#File: BasicAnalysis_ATAC_Macrophages.sh by Claudia Kiesewetter on 04.10.2018
#setting homer environment

DIR_PKG="/misc/software/ngs"
PATH_PERL=/misc/software/package/perl/perl-5.26.1/bin
PATH_SAMTOOLS=£{DIR_PKG}/samtools/samtools-1.6/bin
PATH_HOMER=£{DIR_PKG}/homer/v4.9/bin
PATH_R=/misc/software/package/RBioC/3.4.3/bin
PATH_BASH=/opt/bin/bash
export PATH=£{PATH_BASH}:£{PATH_R}:£{PATH_PERL}:£{PATH_SAMTOOLS}:£{PATH_HOMER}:£{PATH}
export PATH
BEDTOOLS="/misc/software/ngs/bedtools/bedtools2-2.27.1/bin/bedtools"

## Directories ##

WORKDIR="/misc/data/analysis/project_MAC_CK"
PEAKDIR="/misc/data/analysis/project_MAC_CK/Peak_Files/MAC"
PEAKDIRAB="/misc/data/analysis/project_MAC_CK/Peak_Files/MAC/ATAC_CK_AB_TcPlastic_M7"
PEAKDIRCOL1="/misc/data/analysis/project_MAC_CK/Peak_Files/MAC/ATAC_CK_Collagen_I_M7"
PEAKDIRCOL4="/misc/data/analysis/project_MAC_CK/Peak_Files/MAC/ATAC_CK_Collagen_IV_M7"
PEAKDIRHC="/misc/data/analysis/project_MAC_CK/Peak_Files/HC"
ABvsColIPeaks="/misc/data/analysis/project_MAC_CK/Peak_Files/MAC/ATAC_CK_AB_TcPlastic_M7/M7_ABvsColI.getDiffPeaksRep.txt"
ColIvsABPeaks="/misc/data/analysis/project_MAC_CK/Peak_Files/MAC/ATAC_CK_Collagen_I_M7/M7_ColIvsAB.getDiffPeaksRep.txt"
MOTIFDIR="/misc/data/analysis/project_MAC_CK/Motifs/ATAC"
TAGDIR="/misc/data/processedData/tagDir/chromatin/GRCh38/ATAC/MOMACDC"
TAGDIRChIP="/misc/data/processedData/tagDir/chromatin/GRCh38/ChIP/MOMACDC"
TAGDIREGR2="/misc/data/processedData/tagDir/chromatin/GRCh38/ChIP/MOMACDC/publishedData"
TAGDIRINPUT="/misc/data/processedData/tagDir/DNA/GRCh38/Input/MOMACDC/Dec2"
BigWigDIRATAC="/misc/data/processedData/bigWig/chromatin/GRCh38/ATAC/MOMACDC"
FOOTDIR="/misc/data/analysis/project_MAC_CK/Peak_Files/MAC/Footprints"
RDIR="$WORKDIR/R/ATAC"
TMPDIR="/loctmp"
FIGURESDIR="/misc/data/analysis/project_MAC_CK/Figures_ATAC"
IGVDIR="/misc/data/analysis/project_MAC_CK/IGV_Sessions"
KMEANDIR="$PEAKDIR/Kmeans_ABvsColI_EGR2KD"
HEATDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6/Heatmaps"
CHAINHG38="/misc/software/ngs/genome/chainFiles/hg38ToHg19.over.chain"
HG19TRANS="/misc/data/analysis/generalStuff/annotation/hg19/gencode.v19.transcripts.txt"
HG19GTEX="/misc/data/analysis/generalStuff/annotation/hg19/GTEx_v7_eQTL_WBA.snpgenes.1000.bed"
BLACKLIST_HG38="/misc/data/analysis/generalStuff/annotation/GRCh38/hg38.blacklist.bed"
GAPS_HG38="/misc/data/analysis/generalStuff/annotation/GRCh38/gaps.hg38.bed"
CHROMSIZES="/misc/software/viewer/IGV/IGVTools_2.3.98/genomes/GRCh38.PRI_p10.chrom.sizes"

### Sets ###

PEAKSETS="AllSamples AB_TcPlastic_M7 Collagen_I_M7 Collagen_IV_M7 Laminin_M7 Col4_Lam Fibronectin_M7 EGR2KD_CTRL EGR2KD_Mock
↪  EGR2KD_1132 EGR2KD_2665"
TAGDIRSETATAC="$TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIR/ATAC_CK_Fibronectin_M7 $TAGDIR/ATAC_CK_Laminin_M7 $TAGDIR/ATAC_CK_Col4_Lam $TAGDIR/ATAC_CK_EGR2KD_CTRL
↪  $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665"
TAGDIRSETS="$TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIR/ATAC_CK_Fibronectin_M7 $TAGDIR/ATAC_CK_Laminin_M7 $TAGDIR/ATAC_CK_163_ColIV_Laminin
↪  $TAGDIRChIP/H3K27ac/ChIP_CK_19_22_26_MAC_AB_H3K27ac $TAGDIRChIP/H3K27ac/ChIP_CK_H3K27ac_MAC_Col $TAGDIREGR2/MAC_EGR2
↪  $TAGDIRChIP/Dec2/ChIP_CK_13_15_MAC $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_AB $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_Col1
↪  $TAGDIRChIP/PU1/MAC_PU.1_Ch35"
SETS="$TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 $TAGDIR/ATAC_CK_Fibronectin_M7
↪  $TAGDIR/ATAC_CK_Laminin_M7 $TAGDIR/ATAC_CK_163_ColIV_Laminin $TAGDIR/ATAC_CK_EGR2KD_CTRL $TAGDIR/ATAC_CK_EGR2KD_Mock
↪  $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 $TAGDIRChIP/H3K27ac/ChIP_CK_19_22_26_MAC_AB_H3K27ac
↪  $TAGDIRChIP/H3K27ac/ChIP_CK_H3K27ac_MAC_Col $TAGDIREGR2/MAC_EGR2 $TAGDIRChIP/Dec2/ChIP_CK_13_15_MAC
↪  $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_AB $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_Col1 $TAGDIRChIP/PU1/MAC_PU.1_Ch35"
TAGDIRSETMAF="$TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_AB $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_Col1 $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_Col4"

### Setting colors ###

colCol1="deeppink"
colCol4="violetred3"
colAB="slateblue3"
colLam="orchid1"
```

```
colFib="plum3"
colCol4Lam="darkmagenta"
colCTRL="dodgerblue3"
colMock="royalblue2"
col1132="gold2"
col2665="goldenrod3"
colEGR2="forestgreen"
colDec2="mediumorchid"
colMAFAB="orange"
colHC="darkorchid4"
colBMM="darkgoldenrod3"

###############
### Mapping ###
###############

#Samples were mapped with Michaels script: mapATAC.sh to the human genome version hg38

mapATAC.sh -g hg38 -f /misc/data/rawData/chromatin/ATAC/MOMACDC/ATAC_CK_177_S7_R1_001.fastq.gz -r
↪  /misc/data/rawData/chromatin/ATAC/MOMACDC/ATAC_CK_177_S7_R2_001.fastq.gz -d MOMACDC -n ATAC_CK_177_M8_AB -t 12 -s NEXTERA ;
↪  mapATAC.sh -g hg38 -f /misc/data/rawData/chromatin/ATAC/MOMACDC/ATAC_CK_178_S8_R1_001.fastq.gz -r
↪  /misc/data/rawData/chromatin/ATAC/MOMACDC/ATAC_CK_178_S8_R2_001.fastq.gz -d MOMACDC -n ATAC_CK_178_M8_Collagen_IV -t 12 -s
↪  NEXTERA ; mapATAC.sh -g hg38 -f /misc/data/rawData/chromatin/ATAC/MOMACDC/ATAC_CK_179_S9_R1_001.fastq.gz -r
↪  /misc/data/rawData/chromatin/ATAC/MOMACDC/ATAC_CK_179_S9_R2_001.fastq.gz -d MOMACDC -n ATAC_CK_179_M8_Col4_Lam -t 12 -s NEXTERA ;
↪  mapATAC.sh -g hg38 -f /misc/data/rawData/chromatin/ATAC/MOMACDC/ATAC_CK_180_S10_R1_001.fastq.gz -r
↪  /misc/data/rawData/chromatin/ATAC/MOMACDC/ATAC_CK_180_S10_R2_001.fastq.gz -d MOMACDC -n ATAC_CK_180_M8_Laminin -t 12 -s NEXTERA ;

##############################################
#### Create merged Bigwigs for replicates ####
##############################################

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_146_AB_TcPlastic.bigwig $BigWigDIRATAC/ATAC_CK_158_AB_TcPlastic.bigwig -chr $CHROMSIZES
↪  -o $IGVDIR/ATAC_CK_AB_TcPlastic.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_147_Collagen_I.bigwig $BigWigDIRATAC/ATAC_CK_159_Collagen_I.bigwig -chr $CHROMSIZES -o
↪  $IGVDIR/ATAC_CK_Collagen_I.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_148_Collagen_IV.bigwig $BigWigDIRATAC/ATAC_CK_160_Collagen_IV.bigwig -chr $CHROMSIZES
↪  -o $IGVDIR/ATAC_CK_Collagen_IV.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_149_Fibronektin.bigwig $BigWigDIRATAC/ATAC_CK_161_Fibronektin.bigwig -chr $CHROMSIZES
↪  -o $IGVDIR/ATAC_CK_Fibronecin.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_150_Laminin.bigwig $BigWigDIRATAC/ATAC_CK_162_Laminin.bigwig -chr $CHROMSIZES -o
↪  $IGVDIR/ATAC_CK_Laminin.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ ATAC_CK_163_ColIV_Laminin.bigwig $BigWigDIRATAC/ATAC_CK_179_M8_Col4_Lam.bigwig -chr
↪  $CHROMSIZES -o $IGVDIR/ATAC_CK_Col4_Lam.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_145_Monocytes.bigwig $BigWigDIRATAC/ATAC_CK_132_Monocytes.bigwig -chr $CHROMSIZES -o
↪  $IGVDIR/ATAC_CK_Mono.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_169_EGR2KD_2_CTRL.bigwig  $BigWigDIRATAC/ATAC_CK_174_EGR2KD_3_CTRL.bigwig -chr
↪  $CHROMSIZES -o $IGVDIR/ATAC_CK_EGR2KD_CTRL.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_168_EGR2KD_2_Mock.bigwig  $BigWigDIRATAC/ATAC_CK_173_EGR2KD_3_Mock.bigwig -chr
↪  $CHROMSIZES -o $IGVDIR/ATAC_CK_EGR2KD_Mock.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_170_EGR2KD_2_1132.bigwig  $BigWigDIRATAC/ATAC_CK_175_EGR2KD_3_1132.bigwig -chr
↪  $CHROMSIZES -o $IGVDIR/ATAC_CK_EGR2KD_1132.Average.bigwig

myAverageBigWig.pl -bw $BigWigDIRATAC/ATAC_CK_171_EGR2KD_2_2665.bigwig   $BigWigDIRATAC/ATAC_CK_176_EGR2KD_3_2665.bigwig -chr
↪  $CHROMSIZES -o $IGVDIR/ATAC_CK_EGR2KD_2665.Average.bigwig

################################################
#### Merging Tag Directories for Replicates ####
################################################

echo "Merging Tag directories of replicates"

makeTagDirectory $TAGDIR/ATAC_CK_AB_TcPlastic_M7 -d $TAGDIR/ATAC_CK_146_AB_TcPlastic $TAGDIR/ATAC_CK_158_AB_TcPlastic
↪  $TAGDIR/ATAC_CK_177_M8_AB

makeTagDirectory $TAGDIR/ATAC_CK_Collagen_I_M7 -d $TAGDIR/ATAC_CK_147_Collagen_I $TAGDIR/ATAC_CK_159_Collagen_I

makeTagDirectory $TAGDIR/ATAC_CK_Collagen_IV_M7 -d $TAGDIR/ATAC_CK_148_Collagen_IV $TAGDIR/ATAC_CK_160_Collagen_IV
↪  $TAGDIR/ATAC_CK_178_M8_Collagen_IV

makeTagDirectory $TAGDIR/ATAC_CK_Fibronectin_M7 -d $TAGDIR/ATAC_CK_149_Fibronektin $TAGDIR/ATAC_CK_161_Fibronektin

makeTagDirectory $TAGDIR/ATAC_CK_Laminin_M7 -d $TAGDIR/ATAC_CK_150_Laminin $TAGDIR/ATAC_CK_162_Laminin $TAGDIR/ATAC_CK_180_M8_Laminin
makeTagDirectory $TAGDIR/ATAC_CK_Col4_Lam -d $TAGDIR/ATAC_CK_163_ColIV_Laminin $TAGDIR/ATAC_CK_179_M8_Col4_Lam

makeTagDirectory $TAGDIR/ATAC_CK_Mono -d $TAGDIR/ATAC_CK_132_Monocytes $TAGDIR/ATAC_CK_135_Monocytes

makeTagDirectory $TAGDIR/ATAC_CK_EGR2KD_CTRL -d $TAGDIR/ATAC_CK_169_EGR2KD_2_CTRL $TAGDIR/ATAC_CK_174_EGR2KD_3_CTRL

makeTagDirectory $TAGDIR/ATAC_CK_EGR2KD_Mock -d $TAGDIR/ATAC_CK_168_EGR2KD_2_Mock $TAGDIR/ATAC_CK_173_EGR2KD_3_Mock

makeTagDirectory $TAGDIR/ATAC_CK_EGR2KD_1132 -d $TAGDIR/ATAC_CK_170_EGR2KD_2_1132 $TAGDIR/ATAC_CK_175_EGR2KD_3_1132

makeTagDirectory $TAGDIR/ATAC_CK_EGR2KD_2665 -d $TAGDIR/ATAC_CK_171_EGR2KD_2_2665 $TAGDIR/ATAC_CK_176_EGR2KD_3_2665

makeTagDirectory $TAGDIR/HC_6_7_8 -d $TAGDIR/ATAC_CK-72_HC_6 $TAGDIR/ATAC_CK-90_HC_7 $TAGDIR/ATAC_CK-108_HC_8

### Creating one tag directory for all Samples ###

makeTagDirectory $TAGDIR/ATAC_CK_AllSamples -d $TAGDIR/ATAC_CK_146_AB_TcPlastic $TAGDIR/ATAC_CK_158_AB_TcPlastic
↪  $TAGDIR/ATAC_CK_177_M8_AB $TAGDIR/ATAC_CK_147_Collagen_I $TAGDIR/ATAC_CK_159_Collagen_I $TAGDIR/ATAC_CK_148_Collagen_IV
↪  $TAGDIR/ATAC_CK_160_Collagen_IV $TAGDIR/ATAC_CK_178_M8_Collagen_IV $TAGDIR/ATAC_CK_149_Fibronektin
↪  $TAGDIR/ATAC_CK_161_Fibronektin $TAGDIR/ATAC_CK_150_Laminin $TAGDIR/ATAC_CK_162_Laminin $TAGDIR/ATAC_CK_180_M8_Laminin
↪  $TAGDIR/ATAC_CK_163_ColIV_Laminin $TAGDIR/ATAC_CK_179_M8_Col4_Lam

#####################
### Finding Peaks ###
#####################

### ATAC Peaks were defined with Script: findATACPeaksL4.sh, where the required fold change over background was set to 4, to exclude
↪  unspecific peak finding.

echo "Finding Peaks for all ATAC Samples:"

for i in £{PEAKSETS[@]}
do
  mkdir -p $PEAKDIR/ATAC_CK_$i
  /misc/data/analysis/project_MAC_CK/findATACpeaksL4.sh $TAGDIR/ATAC_CK_$i $PEAKDIR/ATAC_CK_$i
done

#Hofbauer Cells
/misc/data/analysis/project_MAC_CK/findATACpeaksL4.sh $TAGDIR/HC_6_7_8 $PEAKDIRHC/HC_6_7_8_Peaks

##############################################
#### Filtering ATAC Peaks by mappability ####
##############################################

for i in £{PEAKSETS[@]}
do
  echo "$i"
  pos2bed.pl $PEAKDIR/ATAC_CK_$i/ATAC_CK_$i.L4.intersect.peaks.txt > $PEAKDIR/ATAC_CK_$i/ATAC_CK_$i.L4.intersect.peaks.bed
  $BEDTOOLS intersect -a $PEAKDIR/ATAC_CK_$i/ATAC_CK_$i.L4.intersect.peaks.bed -b $BLACKLIST_HG38 -v >
 ↪  £{TMPDIR}/ATAC_CK_$i.L4.intersect.peaks.blacklist.bed
  filter4Mappability.sh -p £{TMPDIR}/ATAC_CK_$i.L4.intersect.peaks.blacklist.bed -g GRCh38 -f 0.8 -s 75
  cp £{TMPDIR}/ATAC_CK_$i.L4.intersect.peaks.blacklist.mapScoreFiltered.txt
 ↪  £{PEAKDIR}/ATAC_CK_$i/ATAC_CK_$i.L4.intersect.peaks.blacklist.mapScoreFiltered.txt
done

#Hofbauer cells

intersectBed -a $PEAKDIRHC/HC_6_7_8_Peaks/HC_6_7_8.L4.intersect.peaks.bed -b $BLACKLIST_HG38 -v >
↪  $TMPDIR/HC_6_7_8.L4.intersect.peaks.blacklist.bed
filter4Mappability.sh -p $PTMPDIR/HC_6_7_8.L4.intersect.peaks.blacklist.bed -g GRCh38 -f 0.8 -s 75
cp £{TMPDIR}/HC_6_7_8.L4.intersect.peaks.blacklist.mapScoreFiltered.txt
↪  £{PEAKDIRHC}/HC_6_7_8_Peaks/HC_6_7_8.L4.intersect.peaks.blacklist.mapScoreFiltered.txt

###########################################################
#### Basic annotation of these Peak files using HOMER ####
###########################################################

for i in £{PEAKSETS[@]}
do
  echo "Annotating $i Peak File"
  annotatePeaks.pl £{PEAKDIR}/ATAC_CK_$i/ATAC_CK_$i.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 >
 ↪  £{PEAKDIR}/ATAC_CK_$i/ATAC_CK_$i.L4.intersect.peaks.blacklist.mapScoreFiltered.ann.txt
done

#####################################
#### Scatterplots for Replicates ####
#####################################

annotatePeaks.pl $PEAKDIR/ATAC_CK_AB_TcPlastic_M7/ATAC_CK_AB_TcPlastic_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38
↪  -size 1000 -log -d $TAGDIR/ATAC_CK_146_AB_TcPlastic $TAGDIR/ATAC_CK_158_AB_TcPlastic $TAGDIR/ATAC_CK_177_M8_AB >
↪  $PEAKDIR/ATAC_CK_AB_TcPlastic_M7/ATAC_CK_AB_TcPlastic_M7.blacklist.mapScoreFiltered.scatter_size1000_log.txt
```

```
annotatePeaks.pl $PEAKDIR/ATAC_CK_Collagen_I_M7/ATAC_CK_Collagen_I_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -size
↪  1000 -log -d $TAGDIR/ATAC_CK_147_Collagen_I $TAGDIR/ATAC_CK_159_Collagen_I >
↪  $PEAKDIR/ATAC_CK_Collagen_I_M7/ATAC_CK_Collagen_I_M7.blacklist.mapScoreFiltered.scatter_size1000_log.txt

annotatePeaks.pl $PEAKDIR/ATAC_CK_Collagen_IV_M7/ATAC_CK_Collagen_IV_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -size
↪  1000 -log -d $TAGDIR/ATAC_CK_148_Collagen_IV $TAGDIR/ATAC_CK_160_Collagen_IV $TAGDIR/ATAC_CK_178_M8_Collagen_IV >
↪  $PEAKDIR/ATAC_CK_Collagen_IV_M7/ATAC_CK_Collagen_IV_M7.blacklist.mapScoreFiltered.scatter_size1000_log.txt

annotatePeaks.pl $PEAKDIR/ATAC_CK_Fibronectin_M7/ATAC_CK_Fibronectin_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -size
↪  1000 -log -d $TAGDIR/ATAC_CK_149_Fibronektin $TAGDIR/ATAC_CK_161_Fibronektin >
↪  $PEAKDIR/ATAC_CK_Fibronectin_M7/ATAC_CK_Fibronectin.blacklist.mapScoreFiltered.scatter_size1000_log.txt

annotatePeaks.pl $PEAKDIR/ATAC_CK_Laminin_M7/ATAC_CK_Laminin_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -size 1000
↪  -log -d $TAGDIR/ATAC_CK_150_Laminin $TAGDIR/ATAC_CK_162_Laminin $TAGDIR/ATAC_CK_180_M8_Laminin >
↪  $PEAKDIR/ATAC_CK_Laminin_M7/ATAC_CK_Laminin_M7.blacklist.mapScoreFiltered.scatter_size1000_log.txt

annotatePeaks.pl $PEAKDIR/ATAC_CK_Col4_Lam/ATAC_CK_Col4_Lam.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -size 1000 -log -d
↪  $TAGDIR/ATAC_CK_163_ColIV_Laminin $TAGDIR/ATAC_CK_179_M8_Col4_Lam >
↪  $PEAKDIR/ATAC_CK_Col4_Lam/ATAC_CK_Col4_Lam.blacklist.mapScoreFiltered.scatter_size1000_log.txt

annotatePeaks.pl $PEAKDIR/ATAC_CK_EGR2KD_Mock/ATAC_CK_EGR2KD_Mock.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -size 1000
↪  -log -d $TAGDIR/ATAC_CK_164_EGR2KD_1_Mock $TAGDIR/ATAC_CK_168_EGR2KD_2_Mock $TAGDIR/ATAC_CK_173_EGR2KD_3_Mock >
↪  $PEAKDIR/ATAC_CK_EGR2KD_Mock/ATAC_CK_EGR2KD_Mock.scatter_size1000_log.txt

annotatePeaks.pl $PEAKDIR/ATAC_CK_EGR2KD_CTRL/ATAC_CK_EGR2KD_CTRL.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -size 1000
↪  -log -d $TAGDIR/ATAC_CK_165_EGR2KD_1_CTRL $TAGDIR/ATAC_CK_169_EGR2KD_2_CTRL $TAGDIR/ATAC_CK_174_EGR2KD_3_CTRL >
↪  $PEAKDIR/ATAC_CK_EGR2KD_CTRL/ATAC_CK_EGR2KD_CTRL.scatter_size1000_log.txt

annotatePeaks.pl $PEAKDIR/ATAC_CK_EGR2KD_1132/ATAC_CK_EGR2KD_1132.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -size 1000
↪  -log -d $TAGDIR/ATAC_CK_166_EGR2KD_1_1132 $TAGDIR/ATAC_CK_170_EGR2KD_2_1132 $TAGDIR/ATAC_CK_175_EGR2KD_3_1132 >
↪  $PEAKDIR/ATAC_CK_EGR2KD_1132/ATAC_CK_EGR2KD_1132.scatter_size1000_log.txt

annotatePeaks.pl $PEAKDIR/ATAC_CK_EGR2KD_2665/ATAC_CK_EGR2KD_2665.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -size 1000
↪  -log -d $TAGDIR/ATAC_CK_167_EGR2KD_1_2665 $TAGDIR/ATAC_CK_171_EGR2KD_2_2665 $TAGDIR/ATAC_CK_176_EGR2KD_3_2665 >
↪  $PEAKDIR/ATAC_CK_EGR2KD_2665/ATAC_CK_EGR2KD_2665.scatter_size1000_log.txt

annotatePeaks.pl $PEAKDIRHC/HC_6_7_8_Peaks/HC_6_7_8.L4.intersect.peaks.blacklist.mapScoreFiltered.ann.txt hg38 -size 1000 -log -d
↪  $TAGDIR/ATAC_CK-72_HC_6 $TAGDIR/ATAC_CK-90_HC_7 $TAGDIR/ATAC_CK-108_HC_8 > $PEAKDIRHC/HC_6_7_8_scatter_filtered_size1000_log.txt

#### Creating the plots in R ####

_DATE=$(date +%s)
for i in £{PEAKSETS[@]}; do
  cat > "/loctmp/R.ScatterplotsATAC.$i.P.£{_DATE}.R" <<EOF
  library(ggplot2)
  library(RColorBrewer)
  library(extrafont)

data <- read.delim(file = "£{PEAKDIR}/ATAC_CK_£i/ATAC_CK_£iL4.intersect.peaks.blacklist.mapScoreFiltered.scatter_size1000_log.txt", sep
↪  = "\t")

col <- brewer.pal(11, "RdBu")
col2 <- densCols(x = data[,20], y = data[,21], colramp = colorRampPalette(rev(col)))

#Starting the plot:

lb1 <- paste("r^2 == ", round(cor(data[,20], data[,21], method = "pearson"),2))

p1 <- ggplot(data, aes(x = data[,20], y = data[,21])) + theme_classic() +
  geom_point(aes(colour = col2)) +
  scale_color_identity() +
  annotate("text", x = 9, y = 3, label = lb1, parse = TRUE, size = 10, family = "Arial") +
  labs(y = "M7",
            x = "M6") +
          theme(axis.text = element_text(colour = "gray40", size = 22, family = "Arial"),
                axis.title = element_text(colour = "black", size = 26, family = "Arial"),
                plot.title = element_text(colour = "black", face = "bold", size = 16, family = "Arial"))

  png(file = "£{RDIR}/Scatterplot_£i.png", width = 800, height = 800)
  plot(p1)
  dev.off()

EOF

chmod 750 "/loctmp/R.ScatterplotsATAC.$i.P.£{_DATE}.R"
R < /loctmp/R.ScatterplotsATAC.$i.P.£{_DATE}.R  --no-save
rm /loctmp/R.ScatterplotsATAC.$i.P.£{_DATE}.R

done

### Plots for three replicates ###
```

```
#PEAKSETS="EGR2KD_CTRL EGR2KD_Mock EGR2KD_1132 EGR2KD_2665"
_DATE=$(date +%s)
for i in £{PEAKSETS2[@]}; do
  cat > "/loctmp/R.ScatterplotsATAC.$i.P.£{_DATE}.R" <<EOF
  library(ggplot2)
  library(RColorBrewer)
  library(extrafont)

  data <- read.delim(file = "£{PEAKDIR}/ATAC_CK_£i/ATAC_CK_£i.blacklist.mapScoreFiltered.scatter_size1000_log.txt", sep = "\t")

#Plot between replicate 1 and 2
  col <- brewer.pal(11, "RdBu")
  col2 <- densCols(x = data[,20], y = data[,21], colramp = colorRampPalette(rev(col)))

  #Starting the plot:

  lb1 <- paste("r^2 == ", round(cor(data[,20], data[,21], method = "pearson"),2))

  p1 <- ggplot(data, aes(x = data[,20], y = data[,21])) + theme_classic() +
  geom_point(aes(colour = col2)) +
  scale_color_identity() +
  annotate("text", x = 9, y = 3, label = lb1, parse = TRUE, size = 10, family = "Arial") +
  labs(y = "Replicate 1",
       x = "Replicate 2") +
    theme(axis.text = element_text(colour = "gray40", size = 22, family = "Arial"),
          axis.title = element_text(colour = "black", size = 26, family = "Arial"),
          plot.title = element_text(colour = "black", face = "bold", size = 16, family = "Arial"))

  png(file = "£{RDIR}/Scatterplot_£i.12.png", width = 800, height = 800)
  plot(p1)
  dev.off()

#Plot between replicate 1 and 3
  col <- brewer.pal(11, "RdBu")
  col2 <- densCols(x = data[,20], y = data[,22], colramp = colorRampPalette(rev(col)))

  #Starting the plot:

  lb1 <- paste("r^2 == ", round(cor(data[,20], data[,22], method = "pearson"),2))

  p1 <- ggplot(data, aes(x = data[,20], y = data[,22])) + theme_classic() +
  geom_point(aes(colour = col2)) +
  scale_color_identity() +
  annotate("text", x = 9, y = 3, label = lb1, parse = TRUE, size = 10, family = "Arial") +
  labs(y = "Replicate 1",
       x = "Replicate 3") +
    theme(axis.text = element_text(colour = "gray40", size = 22, family = "Arial"),
          axis.title = element_text(colour = "black", size = 26, family = "Arial"),
          plot.title = element_text(colour = "black", face = "bold", size = 16, family = "Arial"))

  png(file = "£{RDIR}/Scatterplot_£i.13.png", width = 800, height = 800)
  plot(p1)
  dev.off()

  #Plot between replicate 2 and 3
    col <- brewer.pal(11, "RdBu")
    col2 <- densCols(x = data[,21], y = data[,22], colramp = colorRampPalette(rev(col)))

    #Starting the plot:

    lb1 <- paste("r^2 == ", round(cor(data[,21], data[,22], method = "pearson"),2))

    p1 <- ggplot(data, aes(x = data[,21], y = data[,22])) + theme_classic() +
    geom_point(aes(colour = col2)) +
    scale_color_identity() +
    annotate("text", x = 9, y = 3, label = lb1, parse = TRUE, size = 10, family = "Arial") +
    labs(y = "Replicate 2",
         x = "Replicate 3") +
      theme(axis.text = element_text(colour = "gray40", size = 22, family = "Arial"),
            axis.title = element_text(colour = "black", size = 26, family = "Arial"),
            plot.title = element_text(colour = "black", face = "bold", size = 16, family = "Arial"))

    png(file = "£{RDIR}/Scatterplot_£i.23.png", width = 800, height = 800)
    plot(p1)
    dev.off()

EOF
chmod 750 "/loctmp/R.ScatterplotsATAC.$i.P.£{_DATE}.R"
  R < /loctmp/R.ScatterplotsATAC.$i.P.£{_DATE}.R  --no-save
  rm /loctmp/R.ScatterplotsATAC.$i.P.£{_DATE}.R
done
```

```
###################################################
#### Annotation for t-SNE Plot of all samples ####
###################################################

annotatePeaks.pl $PEAKDIR/ATAC_CK_AllSamples/ATAC_CK_AllSamples.L4.intersect.peaks.blacklist.mapScoreFiltered.txt hg38 -d
↪  $TAGDIR/ATAC_CK_146_AB_TcPlastic $TAGDIR/ATAC_CK_158_AB_TcPlastic $TAGDIR/ATAC_CK_177_M8_AB $TAGDIR/ATAC_CK_147_Collagen_I
↪  $TAGDIR/ATAC_CK_159_Collagen_I $TAGDIR/ATAC_CK_148_Collagen_IV $TAGDIR/ATAC_CK_160_Collagen_IV $TAGDIR/ATAC_CK_178_M8_Collagen_IV
↪  $TAGDIR/ATAC_CK_149_Fibronektin $TAGDIR/ATAC_CK_161_Fibronektin $TAGDIR/ATAC_CK_150_Laminin $TAGDIR/ATAC_CK_162_Laminin
↪  $TAGDIR/ATAC_CK_163_ColIV_Laminin $TAGDIR/ATAC_CK_179_M8_Col4_Lam -nogene -noann >
↪  $PEAKDIR/ATAC_CK_AllSamples/ATAC_CK_AllSamples.L4.intersect.peaks.blacklist.mapScoreFiltered.ann.txt

### Using R to remove Batch Effect from this table ###

DATE=$(date +%s)

cat > "/loctmp/R.ATAC.£{_DATE}.R" <<EOF
library(ggplot2)
library(Rtsne)
library(extrafont)
library(edgeR)

#specifying groups of samples
group <- factor(c(rep("AB",3), rep("Collagen_I", 2), rep("Collagen_IV", 3), rep("Fibronectin", 2), rep("Laminin",2),
↪  rep("CollagenIV_Laminin", 2)))

#Donor Batch
donor <- factor(c(1,2,3,1,2,1,2,3,1,2,1,2,2,3))

data <- read.delim(file = "£PEAKDIR/ATAC_CK_AllSamples/ATAC_CK_AllSamples.L4.intersect.peaks.blacklist.mapScoreFiltered.ann.txt", sep =
↪  "\t")
d <- data[,8:21]

#create design matrix:
design <- model.matrix(~0 + group)

rmBE <- removeBatchEffect(d, design = design, batch = donor)
colnames(rmBE) <- c("AB1", "AB2", "AB3", "Col1_1", "Col1_2", "Col4_1", "Col4_2", "Col4_3", "Fib_1", "Fib2", "Lam1", "Lam2", "Col4Lam2",
↪  "Col4Lam3")
write.table(rmBE, file = "£PEAKDIR/ATAC_CK_AllSamples/ATAC_AllSamples_rmBE.txt", sep = "\t", col.names = NA)

#specify colors used
colors = c("£colAB", "£colCol1", "£colCol4", "£colFib", "£colLam", "£colCol4Lam")
colors_group <- c(rep("£colAB", 3), rep("£colCol1", 2), rep("£colCol4", 3), rep("£colFib", 2), rep("£colLam", 2), rep("£colCol4Lam",
↪  2))

pdf(file = "£FIGURESDIR/MDS_ATAC_rmBE.pdf", width = 10, height = 10)
plotMDS(rmBE)
dev.off()

EOF
chmod 750 "/loctmp/R.ATAC.£{_DATE}.R"
R < /loctmp/R.ATAC.£{_DATE}.R --no-save
rm /loctmp/R.ATAC.£{_DATE}.R

####################################
#### Finding Differential Peaks ####
####################################

getDifferentialPeaksReplicates.pl -p
↪  $PEAKDIR/ATAC_CK_AB_TcPlastic_M7/ATAC_CK_AB_TcPlastic_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt -t
↪  $TAGDIR/ATAC_CK_146_AB_TcPlastic $TAGDIR/ATAC_CK_158_AB_TcPlastic -b $TAGDIR/ATAC_CK_147_Collagen_I
↪  $TAGDIR/ATAC_CK_159_Collagen_I $TAGDIR/ATAC_CK_148_Collagen_IV $TAGDIR/ATAC_CK_160_Collagen_IV >
↪  $PEAKDIR/ATAC_CK_AB_TcPlastic_M7/M7_ABvsColI.getDiffPeaksRep.txt

getDifferentialPeaksReplicates.pl -p
↪  $PEAKDIR/ATAC_CK_Collagen_I_M7/ATAC_CK_Collagen_I_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt -t
↪  $TAGDIR/ATAC_CK_147_Collagen_I $TAGDIR/ATAC_CK_159_Collagen_I $TAGDIR/ATAC_CK_148_Collagen_IV $TAGDIR/ATAC_CK_160_Collagen_IV -b
↪  $TAGDIR/ATAC_CK_146_AB_TcPlastic $TAGDIR/ATAC_CK_158_AB_TcPlastic >
↪  $PEAKDIR/ATAC_CK_Collagen_I_M7/M7_ColIvsAB.getDiffPeaksRep.txt

# siRNA vs CTRL #
getDifferentialPeaksReplicates.pl -p
↪  $PEAKDIR/ATAC_CK_EGR2KD_AllSamples/ATAC_CK_EGR2KD_AllSamples.L4.intersect.peaks.blacklist.mapScoreFiltered.txt -t
↪  $TAGDIR/ATAC_CK_166_EGR2KD_1_1132 $TAGDIR/ATAC_CK_170_EGR2KD_2_1132 $TAGDIR/ATAC_CK_175_EGR2KD_3_1132
↪  $TAGDIR/ATAC_CK_167_EGR2KD_1_2665 $TAGDIR/ATAC_CK_171_EGR2KD_2_2665 $TAGDIR/ATAC_CK_176_EGR2KD_3_2665 -b
↪  $TAGDIR/ATAC_CK_164_EGR2KD_1_Mock $TAGDIR/ATAC_CK_168_EGR2KD_2_Mock $TAGDIR/ATAC_CK_173_EGR2KD_3_Mock
↪  $TAGDIR/ATAC_CK_165_EGR2KD_1_CTRL $TAGDIR/ATAC_CK_169_EGR2KD_2_CTRL $TAGDIR/ATAC_CK_174_EGR2KD_3_CTRL >
↪  $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.txt

# CTRL vs siRNA #
```

```
getDifferentialPeaksReplicates.pl -p
↪  $PEAKDIR/ATAC_CK_EGR2KD_AllSamples/ATAC_CK_EGR2KD_AllSamples.L4.intersect.peaks.blacklist.mapScoreFiltered.txt -t
↪  $TAGDIR/ATAC_CK_164_EGR2KD_1_Mock $TAGDIR/ATAC_CK_168_EGR2KD_2_Mock $TAGDIR/ATAC_CK_173_EGR2KD_3_Mock
↪  $TAGDIR/ATAC_CK_165_EGR2KD_1_CTRL $TAGDIR/ATAC_CK_169_EGR2KD_2_CTRL $TAGDIR/ATAC_CK_174_EGR2KD_3_CTRL -b
↪  $TAGDIR/ATAC_CK_166_EGR2KD_1_1132 $TAGDIR/ATAC_CK_170_EGR2KD_2_1132 $TAGDIR/ATAC_CK_175_EGR2KD_3_1132
↪  $TAGDIR/ATAC_CK_167_EGR2KD_1_2665 $TAGDIR/ATAC_CK_171_EGR2KD_2_2665 $TAGDIR/ATAC_CK_176_EGR2KD_3_2665 >
↪  $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.txt

#Hofbauer Cells
getDifferentialPeaks $PEAKDIRHC/HC_6_7_8_Peaks/HC_6_7_8.L4.intersect.peaks.blacklist.mapScoreFiltered.txt $TAGDIR/HC_6_7_8
↪  $TAGDIR/ATAC_MAC_M7 > $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.txt

#### Creating a common ATAC Peak set of macrophages ####
cd $PEAKDIR
mergePeaks -d given -matrix $PEAKDIR/logs/common_PeakSet.stats
↪  $PEAKDIRAB/ATAC_CK_AB_TcPlastic_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt
↪  $PEAKDIRCOL1/ATAC_CK_Collagen_I_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt
↪  $PEAKDIRCOL4/ATAC_CK_Collagen_IV_M7.L4.intersect.peaks.blacklist.mapScoreFiltered.txt > $PEAKDIR/Merged_AB_Col1_Col4.txt

#Column 8 contains information about which peaks overlap
touch $PEAKDIR/Common_PeakSet.txt
awk -F "\t" 'BEGIN {OFS = "\t"} £8 == 3 {print £1,£2,£3,£4,£5,£6}' $PEAKDIR/Merged_AB_Col1_Col4.txt >> $PEAKDIR/Common_PeakSet.txt

### Annotating common Peak File ###
annotatePeaks.pl $PEAKDIR/Common_PeakSet.txt hg38 > $PEAKDIR/Common_PeakSet.ann.txt

#### Creating bed file and Basic Annotation of differential Peaks ####

PEAKDIRAB="£{PEAKDIR}/ATAC_CK_AB_TcPlastic_M7"
PEAKDIRCOL1="£{PEAKDIR}/ATAC_CK_Collagen_I_M7"

pos2bed.pl $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.txt -bed
annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.txt hg38 > $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.ann.txt

pos2bed.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.txt -bed
annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.txt hg38 > $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.ann.txt

pos2bed.pl $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.txt -bed
annotatePeaks.pl $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.txt hg38 > $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.ann.txt

pos2bed.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.txt -bed
annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.txt hg38 > $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.ann.txt

#####################################################################################
### Venn Diagram of Genes close to differential ATAC Peaks and Differential Genes ###
#####################################################################################

awk -F "\t" '{print £1,£16,£8}' £{PEAKDIRCOL1}/M7_ColIvsAB.getDiffPeaksRep.ann.txt > $TMPDIR/awk1.txt
awk -F " " '{print £1,£2,£3}' $TMPDIR/awk1.txt >> $TMPDIR/awk2.txt

cat > "/loctmp/R.VennDiagragm.DifferenialPeaks.R" << EOF
library(venneuler)
library(VennDiagram)
library(sqldf)

ColDiffPeaks <- read.delim("£{PEAKDIRCOL1}/M7_ColIvsAB.getDiffPeaksRep.ann.txt", sep = "\t")
head(ColDiffPeaks)
nrow(ColDiffPeaks) #2579
colnames(ColDiffPeaks) <- c("PeakID", "chr", "start", "end", "strand", "score", "Ratio", "Annotation", "DetailedAnnotation",
↪  "Distance", "PromoterID", "Entrez", "Unigene", "refseq", "Ensemble", "GeneSymbol", "GeneAlias", "Description", "GeneType")
colGenes <- ColDiffPeaks\£GeneSymbol
summary(duplicated(colGenes))
#False: 2139

ABDiffPeaks <- read.delim("£{PEAKDIRAB}/M7_ABvsColI.getDiffPeaksRep.ann.txt", sep = "\t")
head(ABDiffPeaks)
colnames(ABDiffPeaks) <- c("PeakID", "chr", "start", "end", "strand", "score", "Ratio", "Annotation", "DetailedAnnotation", "Distance",
↪  "PromoterID", "Entrez", "Unigene", "refseq", "Ensemble", "GeneSymbol", "GeneAlias", "Description", "GeneType")
summary(duplicated(ABDiffPeaks\£GeneSymbol))
#False: 11051

### Loading RNA-seq data ###
ColvsRest <- read.delim(file = "£COLDIR/Table_ManhattanWardZscores_AllSamples_pairwiseComparisons.logFC2.FDR0.05.txt", sep = "\t")

Col_Up <- sqldf("SELECT GeneSymbol FROM ColvsRest WHERE clusterID = '1'")
colnames(Col_Up) <- c("GeneName")
#523

Col_Down <- sqldf("SELECT GeneSymbol FROM ColvsRest WHERE clusterID = '2'")
colnames(Col_Down) <- c("GeneName")
#791
```

```
### Calculate overlap ###

Col <- sqldf("SELECT GeneSymbol FROM ColDiffPeaks INNER JOIN Col_Up ON GeneSymbol = GeneName")
summary(duplicated(Col))

#128

Ab <- sqldf("SELECT GeneSymbol FROM ABDiffPeaks INNER JOIN Col_Down ON GeneSymbol = GeneName")
summary(duplicated(Ab))
#605

venncol <- venneuler(c(ColU=395, ATACCol=2139, "ColU&ATACCol"=128))
venncol\£labels <- rep("", length(venncol\£labels)) #without any labels, will be added manually later

pdf(file = "£{FIGURESDIR}/Venn_ColUp_ATAC_ColvsAB_GeneSymbols.pdf", width = 8, height = 8)
plot(venncol, col=c("£colCol1", "£col1132"))
dev.off()

vennab <- venneuler(c(ColD=186, ATACAB=11051, "ColD&ATACAB"=605))
vennab\£labels <- rep("", length(vennab\£labels)) #without any labels, will be added manually later

pdf(file = "£{FIGURESDIR}/Venn_ColDown_ATAC_ABvsCol_GeneSymbols.pdf", width = 8, height = 8)
plot(vennab, col=c("£colCol1", "£colCTRL"))
dev.off()

EOF
chmod 750 "/loctmp/R.VennDiagragm.DifferenialPeaks.R"
R < /loctmp/R.VennDiagragm.DifferenialPeaks.R  --no-save
rm /loctmp/R.VennDiagragm.DifferenialPeaks.R


############################################################################
#### Searching for enriched transcription factor motives in these peaks ####
############################################################################

findMotifsGenome.pl $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.txt hg38 $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep -size 200 -len
↪  8,9,10,11,12,13,14 -p 12

findMotifsGenome.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.txt hg38 $MOTIFDIR/M7_ColIvsAB_getDiffPeaksRep -size 200 -len
↪  8,9,10,11,12,13,14 -p 12

findMotifsGenome.pl $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.txt hg38 $MOTIFDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep -size 200 -len
↪  8,9,10,11,12,13,14 -p 12

findMotifsGenome.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.txt hg38 $MOTIFDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep -size 200 -len
↪  8,9,10,11,12,13,14 -p 12

findMotifsGenome.pl $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.txt hg38 $MOTIFDIR/ATAC_HCvsMAC_getDiffPeaksRep -size 200 -len
↪  8,9,10,11,12,13,14 -p 10

##############################################################################
### Footprints for Transcription Factor Motives found for differential Peaks ###
##############################################################################

### Centering ABvsColIPeaks on pu1 Motif found ###

annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/homerResults/motif1.motif -size given >
↪  $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.PU1.txt

annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.PU1.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/M7_ABvsColI_getDiffPeaksRep.centered.PU1.hist.txt

### Centering ABvsColIPeaks on cebbpa Motif found ###

annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/homerResults/motif2.motif -size given >
↪  $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.CEBPA.txt

annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.CEBPA.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/M7_ABvsColI_getDiffPeaksRep.centered.CEBPA.hist.txt

### Centering ABvsColIPeaks on junb Motif found ###

annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/homerResults/motif2.motif -size given >
↪  $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.JUNB.txt
```

```
annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.JUNB.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/M7_ABvsColI_getDiffPeaksRep.centered.JUNB.hist.txt

### Centering ABvsColIPeaks on EGR2 Motif ###

# EGR2 motif is 9bp long #
annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/homerResults/motif4.motif -size given >
↪  $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.EGR2.txt

annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.EGR2.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/M7_ABvsColI_getDiffPeaksRep.centered.EGR2.hist.txt

### Centering ABvsColIPeaks on bhlh Motif found ###

# bhlh motif is 8bp long #
annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/homerResults/motif6.motif -size given >
↪  $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.Dec2.txt

annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.centered.Dec2.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/M7_ABvsColI_getDiffPeaksRep.centered.Dec2.hist.txt

### Centering ColIvsABPeaks on MAF motif ###

#own motif
annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/M7_ColIvsAB_getDiffPeaksRep/homerResults/motif3.motif -size given >
↪  $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.centered.MAF.txt

annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.centered.MAF.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/M7_ColIvsAB_getDiffPeaksRep.centered.MAF.hist.txt

### Centering ColIvsABPeaks on RUNX motif ###

annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/M7_ColIvsAB_getDiffPeaksRep/homerResults/motif5.motif -size given >
↪  $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.centered.RUNX.txt

annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.centered.RUNX.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/M7_ColIvsAB_getDiffPeaksRep.centered.RUNX.hist.txt

### Centering ColIvsABPeaks on PU1/IRF motif ###

annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/M7_ColIvsAB_getDiffPeaksRep/homerResults/motif2.motif -size given >
↪  $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.centered.PU1IRF.txt

annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.centered.PU1IRF.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/M7_ColIvsAB_getDiffPeaksRep.centered.PU1IRF.hist.txt

### Centering ColIvsABPeaks on ETS motif ###

annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/M7_ColIvsAB_getDiffPeaksRep/homerResults/motif1.motif -size given >
↪  $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.centered.ETS.txt

annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.centered.ETS.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/M7_ColIvsAB_getDiffPeaksRep.centered.ETS.hist.txt

### Centering CTRLvsSiRNA Peaks on EGR2 motif ###

annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep/homerResults/motif4.motif -size given >
↪  $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.EGR2.txt

annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.EGR2.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_EGR2KD_CTRL $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 >
↪  $FOOTDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.EGR2.hist.txt

### Centering CTRLvsSiRNA Peaks on bHLH motif ###

annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep/homerResults/motif5.motif -size given >
↪  $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.bhlh.txt
```

```
annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.bhlh.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_EGR2KD_CTRL $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 >
↪  $FOOTDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.bhlh.hist.txt

### Centering CTRLvsSiRNA Peaks on pu1 motif ###

annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep/homerResults/motif1.motif -size given >
↪  $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.PU1.txt

annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.PU1.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_EGR2KD_CTRL $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 >
↪  $FOOTDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.PU1.hist.txt

### Centering CTRLvsSiRNA Peaks on bZIP motif ###

annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep/homerResults/motif2.motif -size given >
↪  $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.bZIP.txt

annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.bZIP.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_EGR2KD_CTRL $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 >
↪  $FOOTDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.bZIP.hist.txt

### Centering CTRLvsSiRNA Peaks on CEBPA motif ###

annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep/homerResults/motif3.motif -size given >
↪  $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.CEBPA.txt

annotatePeaks.pl $PEAKDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.CEBPA.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_EGR2KD_CTRL $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 >
↪  $FOOTDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.CEBPA.hist.txt

### Centering siRNAvsCTRL Peaks on ETS/IRF motif ###

annotatePeaks.pl $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep/homerResults/motif1.motif -size given >
↪  $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.PU1IRF.txt

annotatePeaks.pl $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.PU1IRF.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_EGR2KD_CTRL $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 >
↪  $FOOTDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.PU1IRF.hist.txt

### Centering siRNAvsCTRL Peaks on ETS motif ###

annotatePeaks.pl $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep/homerResults/motif2.motif -size given >
↪  $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.ETS.txt

annotatePeaks.pl $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.ETS.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_EGR2KD_CTRL $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 >
↪  $FOOTDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.ETS.hist.txt

### Centering siRNAvsCTRL Peaks on RUNX motif ###

annotatePeaks.pl $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.txt hg38 -center
↪  $MOTIFDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep/homerResults/motif3.motif -size given >
↪  $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.runx.txt

annotatePeaks.pl $PEAKDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.runx.txt hg38 -size 550 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_EGR2KD_CTRL $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 >
↪  $FOOTDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.runx.hist.txt

### Centering Hofbauer Cell ATAC Data on ETS motif ###

annotatePeaks.pl $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.txt hg38 -center
↪  $MOTIFDIR/ATAC_HCvsMAC_getDiffPeaksRep/homerResults/motif1.motif -size given >
↪  $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.centered.ETS.txt
annotatePeaks.pl $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.centered.ETS.txt hg38 -size 550 -hist 1 -len 1 -d $TAGDIR/HC_6_7_8
↪  $TAGDIR/ATAC_MAC_M7 > $FOOTDIR/ATAC_HCvsMAC_getDiffPeaks.centered.ETS.txt

### Centering Hofbauer Cell ATAC Data on bZIP motif ###

annotatePeaks.pl $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.txt hg38 -center
↪  $MOTIFDIR/ATAC_HCvsMAC_getDiffPeaksRep/homerResults/motif2.motif -size given >
↪  $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.centered.bZIP.txt
annotatePeaks.pl $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.centered.bZIP.txt hg38 -size 550 -hist 1 -len 1 -d $TAGDIR/HC_6_7_8
↪  $TAGDIR/ATAC_MAC_M7 > $FOOTDIR/ATAC_HCvsMAC_getDiffPeaks.centered.bZIP.txt

### Centering Hofbauer Cell ATAC Data on NR4A1 motif ###
```

```
annotatePeaks.pl $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.txt hg38 -center
↪  $MOTIFDIR/ATAC_HCvsMAC_getDiffPeaksRep/homerResults/motif3.motif -size given >
↪  $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.centered.NR4A1.txt
annotatePeaks.pl $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.centered.NR4A1.txt hg38 -size 550 -hist 1 -len 1 -d $TAGDIR/HC_6_7_8
↪  $TAGDIR/ATAC_MAC_M7 > $FOOTDIR/ATAC_HCvsMAC_getDiffPeaks.centered.NR4A1.txt

### Centering Hofbauer Cell ATAC Data on MAF motif ###

annotatePeaks.pl $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.txt hg38 -center
↪  $MOTIFDIR/ATAC_HCvsMAC_getDiffPeaksRep/homerResults/motif4.motif -size given >
↪  $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.centered.MAF.txt
annotatePeaks.pl $PEAKDIRHC/ATAC_HCvsMAC_getDiffPeaks.centered.MAF.txt hg38 -size 550 -hist 1 -len 1 -d $TAGDIR/HC_6_7_8
↪  $TAGDIR/ATAC_MAC_M7 > $FOOTDIR/ATAC_HCvsMAC_getDiffPeaks.centered.MAF.txt

##### AB vs Collagen I ######
_DATE=$(date +%s)

#### PU1, MAF, NFAT --> 11bp motif ####

FOOTSETAB="PU1 NFAT MAF"
for i in £{FOOTSETAB[@]}; do
cat > "/loctmp/R.Footprints.ABvsColI.$i.R" << EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/M7_ABvsColI_getDiffPeaksRep.centered.£i.hist.txt", sep = "\t")
colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-5

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "ATAC Tag count")+
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-5,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪  element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ABvsColI_getDiffPeaksRep.centered.£i.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-5,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪  element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ABvsColI_getDiffPeaksRep.centered.£i.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.ABvsColI.$i.R"
R < /loctmp/R.Footprints.ABvsColI.$i.R --no-save
rm /loctmp/R.Footprints.ABvsColI.$i.R
done

### 10bp motives ###

FOOTSETAB="CEBPA JUNB EGR2 Dec2"
for i in £{FOOTSETAB[@]}; do
  cat > "/loctmp/R.Footprints.£{_DATE}.$i.R" <<EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/M7_ABvsColI_getDiffPeaksRep.centered.£i.hist.txt", sep = "\t")
colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-5
```

```
p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
        labs(x="", y = "ATAC Tag count")+
  geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
  xlim(-300,300) +
  scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
  theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
      axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major
↪  = element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ABvsColI_getDiffPeaksRep.centered.£i.hist.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
        labs(x="", y = "")+
  xlim(-10,10) +
  geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
  theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
      axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major
↪  = element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ABvsColI_getDiffPeaksRep.centered.£i.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.$i.R"
R < /loctmp/R.Footprints.£{_DATE}.$i.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.$i.R
done

### 9bp motif PU1:IRF ###

cat > "/loctmp/R.Footprints.£{_DATE}.ETS.R" <<EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/M7_ABvsColI.getDiffPeaksRep.centered.PU1IRF.hist.txt", sep = "\t")
colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-4

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "ATAC Tag count")+
geom_vline(xintercept=c(-4,5), linetype="dashed", color="black", size=.25, alpha=.5) +
geom_hline(yintercept=c(8.88), linetype = "dashed", color = "firebrick2", size = .25, alpha = .75) +
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪  element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ABvsColI_getDiffPeaksRep.centered.PU1IRF.hist.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-4,5), linetype="dashed", color="black", size=.25, alpha=.5) +
geom_hline(yintercept=c(8.88), linetype = "dashed", color = "firebrick2", size = .25, alpha = .75) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ABvsColI_getDiffPeaksRep.centered.PU1IRF.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.ETS.R"
R < /loctmp/R.Footprints.£{_DATE}.ETS.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.ETS.R

### RUNX 8bp motif ###

cat > "/loctmp/R.Footprints.£{_DATE}.ColIvsAB.R" << EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/M7_ABvsColI_getDiffPeaksRep.centered.RUNX.hist.txt", sep = "\t")
colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-4

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "ATAC Tag count")+
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-4,4), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ABvsColI_getDiffPeaksRep.centered.RUNX.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-4,4), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ABvsColI_getDiffPeaksRep.centered.RUNX.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.ColIvsAB.R"
R < /loctmp/R.Footprints.£{_DATE}.ColIvsAB.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.ColIvsAB.R

#### Footprint for Collagen vs AB Peaks ###

## ETS MEF2C 10 bp long ##

FOOTSETCol3="ETS MEF2C EGR2"
for i in £{FOOTSETCol3[@]}; do
  cat > "/loctmp/R.Footprints.£{_DATE}.$i.R" <<EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/M7_ColIvsAB_getDiffPeaksRep.centered.£i.hist.txt", sep = "\t")
colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-5

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "ATAC Tag count")+
geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ColIvsAB_getDiffPeaksRep.centered.£i.hist.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ColIvsAB_getDiffPeaksRep.centered.£i.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.$i.R"
R < /loctmp/R.Footprints.£{_DATE}.$i.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.$i.R
done

### ETS/IRF 9bp ###
#+NFI-halfsite Motif 8 9bp

FOOTSETCol1="PU1IRF NFI"
for i in £{FOOTSETCol1[@]}; do
  cat > "/loctmp/R.Footprints.£{_DATE}.$i.R" <<EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/M7_ColIvsAB_getDiffPeaksRep.centered.£i.hist.txt", sep = "\t")
colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-4

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "ATAC Tag count")+
geom_vline(xintercept=c(-4,5), linetype="dashed", color="black", size=.25, alpha=.5) +
geom_hline(yintercept=c(6.5), linetype = "dashed", color = "firebrick2", size = .25, alpha = .75) +
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ColIvsAB_getDiffPeaksRep.centered.£i.hist.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-4,5), linetype="dashed", color="black", size=.25, alpha=.5) +
geom_hline(yintercept=c(6.5), linetype = "dashed", color = "firebrick2", size = .25, alpha = .75) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ColIvsAB_getDiffPeaksRep.centered.£i.zoomedIn.pdf", width = 5, height = 5)
```

```
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.$i.R"
R < /loctmp/R.Footprints.£{_DATE}.$i.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.$i.R
done

### MAF, NFAT own motif 11bp ###

FOOTSETCol2="MAF NFAT"
for i in £{FOOTSETCol2[@]}; do
  cat > "/loctmp/R.Footprints.£{_DATE}.ColIvsAB.R" << EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/M7_ColIvsAB_getDiffPeaksRep.centered.£i.hist.txt", sep = "\t")
colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-5

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "ATAC Tag count")+
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-5,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪  element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ColIvsAB_getDiffPeaksRep.centered.£i.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-5,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪  element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ColIvsAB_getDiffPeaksRep.centered.£i.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.ColIvsAB.R"
R < /loctmp/R.Footprints.£{_DATE}.ColIvsAB.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.ColIvsAB.R
done

### RUNX 8bp motif ###

cat > "/loctmp/R.Footprints.£{_DATE}.ColIvsAB.R" << EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/M7_ColIvsAB_getDiffPeaksRep.centered.RUNX.hist.txt", sep = "\t")
colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-4

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "ATAC Tag count")+
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-4,4), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ColIvsAB_getDiffPeaksRep.centered.RUNX.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= ATAC_AB_Coverage), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= ATAC_AB_Coverage), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= ATAC_Col1_Coverage), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= ATAC_Col1_Coverage), alpha = 0.2, fill = "£colCol4") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-4,4), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.M7_ColIvsAB_getDiffPeaksRep.centered.RUNX.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.ColIvsAB.R"
R < /loctmp/R.Footprints.£{_DATE}.ColIvsAB.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.ColIvsAB.R

##### Footprints for Knock Down data #####
_DATE=$(date +%s)

### ETS motif 11bp ###

cat > "/loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R" << EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.PU1.hist.txt", sep = "\t")
#colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-5

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "ATAC Tag count")+
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-5,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.PU1.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-5,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.PU1.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R"
R < /loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R
```

```
### bZIP motif 9bp long ###

cat > "/loctmp/R.Footprints.£{_DATE}.ETS.R" <<EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.bZIP.hist.txt", sep = "\t")
colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos", "ATAC_Col1_neg")
z\£Distance=z\£Distance-4

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "ATAC Tag count")+
geom_vline(xintercept=c(-4,5), linetype="dashed", color="black", size=.25, alpha=.5) +
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪  element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.bZIP.hist.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-4,5), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪  element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.bZIP.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.ETS.R"
R < /loctmp/R.Footprints.£{_DATE}.ETS.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.ETS.R

### 10bp long motifs ###

FOOTSETCTRL="CEBPA EGR2 bhlh"
for i in £{FOOTSETCTRL[@]}; do
  cat > "/loctmp/R.Footprints.£{_DATE}.$i.R" <<EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.£i.hist.txt", sep = "\t")
z\£Distance=z\£Distance-5

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
        labs(x="", y = "ATAC Tag count")+
  geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
  xlim(-300,300) +
  scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
  theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
      axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major
↪  = element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.£i.hist.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
        labs(x="", y = "")+
  xlim(-10,10) +
  geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
  theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
      axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major
↪  = element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_CTRLvsSiRNA_getDiffPeaksRep.centered.£i.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.$i.R"
R < /loctmp/R.Footprints.£{_DATE}.$i.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.$i.R
done

#### siRNAvsCTRL

##PU1/IRF 12bp long ##

cat > "/loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R" << EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.PU1IRF.hist.txt", sep = "\t")
z\£Distance=z\£Distance-6

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "ATAC Tag count")+
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-6,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.PU1IRF.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-6,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.PU1IRF.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R"
R < /loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R

## ETS 10bp long ##

cat > "/loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R" << EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.ETS.hist.txt", sep = "\t")
z\£Distance=z\£Distance-5

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "ATAC Tag count")+
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.ETS.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.ETS.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R"
R < /loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R

## RUNX 8bp motif ##

cat > "/loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R" << EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.runx.hist.txt", sep = "\t")
z\£Distance=z\£Distance-4

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "ATAC Tag count")+
xlim(-300,300) +
scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-4,4), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.runx.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colCTRL", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colCTRL", alpha = 0.4) +
geom_line(data = z, aes(y= z[,8]), col = "£col1132", size = 0.25) +
geom_area(data = z, aes(y= z[,8]), alpha = 0.2, fill = "£col1132") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-4,4), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.EGR2KD_siRNAvsCTRL_getDiffPeaksRep.centered.runx.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()
```

```
EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R"
R < /loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.ABvsColI.PU1.R

### R Script Footprints for Hofbauer Cell Data ###

### 9 bp ###

FOOTSETHC="ETS bZIP NR4A1 MAF NFATC1Known"
## -> Distance -5

for TF in £{FOOTSETHC[@]}; do
  cat > "/loctmp/R.Footprints.$TF.R" <<EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/ATAC_HCvsMAC_getDiffPeaksRep.centered.£TF.txt", sep = "\t")
z\£Distance=z\£Distance-5

p2 <- ggplot(z, aes(x= z[,1])) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colHC", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colHC", alpha = 0.4) +
geom_line(data = z, aes(y= z[,5]), col = "£colBMM", size = 0.25) +
geom_area(data = z, aes(y= z[,5]), alpha = 0.2, fill = "£colBMM") +
labs(x="", y = "ATAC Tag count")+
xlim(-100,100) + ylim(0,25) +
#scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪   element_blank())

pdf(file = "£FOOTDIR/Footprint.ATAC_HCvsMAC_getDiffPeaksRep.£TF.pdf", width = 8, height = 8, useDingbats = FALSE)
plot(p2)
dev.off()

pdf("Legend_Footprint_HC.pdf", width = 2, height = 2)
par(family = "ArialMT")
plot.new()
legend("top", legend = c("CTRL", "1132"), col = c("£colHC", "£colBMM"), lwd = 5, lty = 1, bty = "n")
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x= z[,1])) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colHC", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colHC", alpha = 0.4) +
geom_line(data = z, aes(y= z[,5]), col = "£colBMM", size = 0.25) +
geom_area(data = z, aes(y= z[,5]), alpha = 0.2, fill = "£colBMM") +
labs(x="", y = "ATAC Tag count")+
  xlim(-10,10) +
  geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
  theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
      axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major
↪   = element_blank())

pdf(file = "£FOOTDIR/Footprint.ATAC_HCvsMAC_getDiffPeaksRep.£TF.zoomedIn.pdf", width = 8, height = 8, useDingbats = FALSE)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.$TF.R"
R < /loctmp/R.Footprints.$TF.R --no-save
rm /loctmp/R.Footprints.$TF.R
done

## Konw PU-1 IRF ##
#Distance -6
_DATE=$(date +%s)
cat > "/loctmp/R.Footprints.£{_DATE}.HC.PU1.R" << EOF
library(ggplot2)
library(extrafont)

z <- read.delim(file = "£FOOTDIR/ATAC_HCvsMAC_getDiffPeaksRep.centered.PU1IRFKnown.txt", sep = "\t")
z\£Distance=z\£Distance-6

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colHC", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colHC", alpha = 0.4) +
geom_line(data = z, aes(y= z[,5]), col = "£colBMM", size = 0.25) +
geom_area(data = z, aes(y= z[,5]), alpha = 0.2, fill = "£colBMM") +
labs(x="", y = "ATAC Tag count")+
xlim(-100,100) +
#scale_x_continuous(breaks = c(-300,-150,0,150,300), labels = c(-300,-150,0,150,300)) +
geom_vline(xintercept=c(-6,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪  element_blank())

pdf(file = "£FOOTDIR/Footprint.ATAC_HCvsMAC_getDiffPeaksRep.centered.PU1IRFKnown.pdf", width = 5, height = 5)
plot(p2)
dev.off()

### Zoomed in region of Footprint ###

p2 <- ggplot(z, aes(x=Distance)) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colHC", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colHC", alpha = 0.4) +
geom_line(data = z, aes(y= z[,5]), col = "£colBMM", size = 0.25) +
geom_area(data = z, aes(y= z[,5]), alpha = 0.2, fill = "£colBMM") +
labs(x="", y = "")+
xlim(-10,10) +
geom_vline(xintercept=c(-6,6), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 16, colour = "black", family = "ArialMT"),
    axis.title = element_text(size = 18, colour = "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major =
↪  element_blank())

pdf(file = "£FOOTDIR/Footprint.ATAC_HCvsMAC_getDiffPeaksRep.centered.PU1IRFKnown.zoomedIn.pdf", width = 5, height = 5)
plot(p2)
dev.off()

EOF
chmod 750 "/loctmp/R.Footprints.£{_DATE}.HC.PU1.R"
R < /loctmp/R.Footprints.£{_DATE}.HC.PU1.R --no-save
rm /loctmp/R.Footprints.£{_DATE}.HC.PU1.R

############################################
### Histogramme with confidence interval ###
############################################

SETS="$TAGDIR/ATAC_CK_Mono $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIR/ATAC_CK_Fibronectin_M7 $TAGDIR/ATAC_CK_Laminin_M7 $TAGDIR/ATAC_CK_Col4_Lam $TAGDIR/ATAC_CK_EGR2KD_CTRL
↪  $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665
↪  $TAGDIRChIP/H3K27ac/ChIP_CK_19_22_26_MAC_AB_H3K27ac $TAGDIRChIP/H3K27ac/ChIP_CK_H3K27ac_MAC_Col4
↪  $TAGDIRChIP/H3K27ac/ChIP_CK_H3K27ac_MAC_Col4  $TAGDIREGR2/MAC_EGR2 $TAGDIRChIP/Dec2/ChIP_CK_13_15_MAC
↪  $TAGDIRChIP/MAF/ChIP_CK_cMAF_AB $TAGDIRChIP/MAF/ChIP_CK_cMAF_Col1"

for i in £{SETS}; do
  NAMEBASE=£{i##*/}
        NAME=£{NAMEBASE%.*}
  annotatePeaks.pl $PEAKDIRAB/M7_ABvsColI.getDiffPeaksRep.txt hg38 -size 2000 -hist 25 -ghist -d $i >
 ↪  $PEAKDIR/M7.ABvsColI.getDiffPeaksRep.$NAME.ghist.txt
  annotatePeaks.pl $PEAKDIRCOL1/M7_ColIvsAB.getDiffPeaksRep.txt hg38 -size 2000 -hist 25 -ghist -d $i >
 ↪  $PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.$NAME.ghist.txt
done

### Histogram of ATAC data ###
#ABvsColI#
cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_AB_TcPlastic_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 2
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_Collagen_I_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result2 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 3
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_Col4_Lam.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result3 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colAB") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colCol1")+
      geom_ribbon(data = result3, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result3, colour = "£colCol4Lam")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ATAC-seq coverage") +
      xlim(-500, 500) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪  element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ABvsColI.getDiffPeaksRep.ATAC.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"

# With monocytes #

cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_146_AB_TcPlastic.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 2
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_147_Collagen_I.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result2 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 3
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_145_Monocytes.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result3 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"#
  #Data 4
   data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_158_AB_TcPlastic.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result4 <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 5
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_159_Collagen_I.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result5 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 6
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.AK17_Monos_0h.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result6 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"

  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colAB") +
      geom_ribbon(data = result4, aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result4, colour = "£colAB") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colCol1")+
      geom_ribbon(data = result5, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result5, colour = "£colCol1")+
      geom_ribbon(data = result3, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result3, colour = "£colMono")+
      geom_ribbon(data = result6, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result6, colour = "£colMono")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ATAC-seq coverage") +
      xlim(-500, 500) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪  element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ABvsColI.getDiffPeaksRep.ATACMono.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"

# with EGR2 Knock down #

cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_AB_TcPlastic_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 2
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_Collagen_I_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result2 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 3
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_EGR2KD_CTRL.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result3 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 4
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_EGR2KD_1132.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result4 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 5
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_EGR2KD_2665.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result5 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 6
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_Collagen_IV_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result6 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 7
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_Mono.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result7 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 8
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_EGR2KD_Mock.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result8 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colAB") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colCol1")+
      geom_ribbon(data = result3, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result3, colour = "£colCTRL")+
      geom_ribbon(data = result4, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result4, colour = "£col1132")+
      geom_ribbon(data = result5, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result5, colour = "£col2665")+
      geom_ribbon(data = result6, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result6, colour = "£colCol4")+
      geom_ribbon(data = result7, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result7, colour = "£colMono")+
      geom_ribbon(data = result8, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result8, colour = "£colMock")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ATAC-seq coverage") +
      xlim(-500, 500) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪  element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ABvsColI.getDiffPeaksRep.ATAC.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"

#ColIsAB#
```

```
cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ATAC_CK_AB_TcPlastic_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 2
  data <- read.table("£PEAKDIR/M7.ColIsAB.getDiffPeaksRep.ATAC_CK_Collagen_I_M7.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result2 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 3
  data <- read.table("£PEAKDIR/M7.ColIsAB.getDiffPeaksRep.ATAC_CK_Col4_Lam.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result3 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colAB") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colCol1")+
      geom_ribbon(data = result3, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result3, colour = "£colCol4Lam")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ATAC-seq coverage") +
      xlim(-500, 500) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪  element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ColIsAB.getDiffPeaksRep.ATAC.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"

#ColvsAB with monocytes
cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_146_AB_TcPlastic.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
```

```
    error <- qnorm(0.975)*s/sqrt(n)
    #upper and lower boundaries
    ci975 <- m+error
    ci025 <- m-error
    d <- as.numeric(as.character(rownames(tdata)))
    result <- data.frame(cbind(d,m, s,ci975,ci025))
    colband <- "gray"
   #Data 2
   data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_147_Collagen_I.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result2 <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
   #Data 3
   data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.all_MO.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE, check.names =
↪  FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result3 <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"#
   #Data 4
    data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_158_AB_TcPlastic.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
    tdata <- t(data[,-1])
    m <- apply(tdata, 1, mean)
    s <- apply(tdata, 1, sd)
    n <- nrow(as.matrix(tdata))
    #calculating the confidence interval
    error <- qnorm(0.975)*s/sqrt(n)
    #upper and lower boundaries
    ci975 <- m+error
    ci025 <- m-error
    d <- as.numeric(as.character(rownames(tdata)))
    result4 <- data.frame(cbind(d,m, s,ci975,ci025))
    colband <- "gray"
   #Data 5
   data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_159_Collagen_I.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result5 <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
   #Data 6
   data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.AK17_Monos_0h.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result6 <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
```

```
    #Start plot
     p <- ggplot(result, aes(d,m)) +
        geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
        geom_line(data = result, colour = "£colAB") +
        geom_ribbon(data = result4, aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
        geom_line(data = result4, colour = "£colAB") +
        geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
        geom_line(data=result2, colour = "£colCol1")+
        geom_ribbon(data = result5, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
        geom_line(data=result5, colour = "£colCol1")+
        geom_ribbon(data = result3, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
        geom_line(data=result3, colour = "£colMono")+
        geom_ribbon(data = result6, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
        geom_line(data=result6, colour = "£colMono")+
        theme_bw(base_size=12) +
        xlab("Distance from peak center") +
        ylab("ATAC-seq coverage") +
        xlim(-500, 500) +
        theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪   element_blank(), panel.grid.major = element_blank())
     pdf(file="£FIGURESDIR/M7.ColvsAB.getDiffPeaksRep.ATACMono.hist.pdf", height=4, width=4)
     plot(p)
     dev.off()
  EOF
  chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
  R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
  rm "£{TMPDIR}/R.kmeans.Histogramme.R"

  # with EGR2 Knock down #

  cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
     library(ggplot2)
     library(reshape)
    #Data 1
     data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_AB_TcPlastic_M7.ghist.txt", header=T, sep="\t",
↪     stringsAsFactors=FALSE, check.names = FALSE)
     tdata <- t(data[,-1])
     m <- apply(tdata, 1, mean)
     s <- apply(tdata, 1, sd)
     n <- nrow(as.matrix(tdata))
     #calculating the confidence interval
     error <- qnorm(0.975)*s/sqrt(n)
     #upper and lower boundaries
     ci975 <- m+error
     ci025 <- m-error
     d <- as.numeric(as.character(rownames(tdata)))
     result <- data.frame(cbind(d,m, s,ci975,ci025))
     colband <- "gray"
    #Data 2
    data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_Collagen_I_M7.ghist.txt", header=T, sep="\t",
↪     stringsAsFactors=FALSE, check.names = FALSE)
    tdata <- t(data[,-1])
    m <- apply(tdata, 1, mean)
    s <- apply(tdata, 1, sd)
    n <- nrow(as.matrix(tdata))
    #calculating the confidence interval
    error <- qnorm(0.975)*s/sqrt(n)
    #upper and lower boundaries
    ci975 <- m+error
    ci025 <- m-error
    d <- as.numeric(as.character(rownames(tdata)))
    result2 <- data.frame(cbind(d,m, s,ci975,ci025))
    colband <- "gray"
    #Data 3
    data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_EGR2KD_CTRL.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪     check.names = FALSE)
    tdata <- t(data[,-1])
    m <- apply(tdata, 1, mean)
    s <- apply(tdata, 1, sd)
    n <- nrow(as.matrix(tdata))
    #calculating the confidence interval
    error <- qnorm(0.975)*s/sqrt(n)
    #upper and lower boundaries
    ci975 <- m+error
    ci025 <- m-error
    d <- as.numeric(as.character(rownames(tdata)))
    result3 <- data.frame(cbind(d,m, s,ci975,ci025))
    colband <- "gray"
    #Data 4
    data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_EGR2KD_1132.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪     check.names = FALSE)
```

```
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result4 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 5
  data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_EGR2KD_2665.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result5 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 6
  data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_Collagen_IV_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result6 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 7
  data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_Mono.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result7 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Data 8
  data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC_CK_EGR2KD_Mock.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result8 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colAB") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colCol1")+
      geom_ribbon(data = result3, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result3, colour = "£colCTRL")+
      geom_ribbon(data = result4, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result4, colour = "£col1132")+
      geom_ribbon(data = result5, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result5, colour = "£col2665")+
      geom_ribbon(data = result6, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result6, colour = "£colCol4")+
      geom_ribbon(data = result7, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result7, colour = "£colMono")+
      geom_ribbon(data = result8, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result8, colour = "£colMock")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ATAC-seq coverage") +
      xlim(-500, 500) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪    element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ColIvsAB.getDiffPeaksRep.ATAC.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"


### Histone Acetylation Histogram ###
#ABvsColI#
cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ChIP_CK_19_22_26_MAC_AB_H3K27ac.ghist.txt", header=T, sep="\t",
↪    stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 2
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ChIP_CK_H3K27ac_MAC_Col1.ghist.txt", header=T, sep="\t",
↪    stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result2 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colAB") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colCol1")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ChIP-seq coverage") +
      xlim(-1000, 1000) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪    element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ABvsColI.getDiffPeaksRep.H3K27ac.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"

#ColIsAB#
cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ChIP_CK_19_22_26_MAC_AB_H3K27ac.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 2
  data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ChIP_CK_H3K27ac_MAC_Col1.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result2 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colAB") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colCol1")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ChIP-seq coverage") +
      xlim(-1000, 1000) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪  element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ColIvsAB.getDiffPeaksRep.H3K27ac.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"

### EGR2 and Dec2 ###
cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.MAC_EGR2.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 2
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ChIP_CK_13_15_MAC.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result2 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colEGR2") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colDec2")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ChIP-seq coverage") +
      xlim(-500, 500) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪    element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ABvsColI.getDiffPeaksRep.EGR2Dec2.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"

### cMAF ###
#ABvsColI#
cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ChIP_CK_cMAF_AB.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪    check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 2
  data <- read.table("£PEAKDIR/M7.ABvsColI.getDiffPeaksRep.ChIP_CK_cMAF_Col1.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪    check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result2 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colAB") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colCol1")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ChIP-seq coverage") +
      xlim(-500, 500) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪    element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ABvsColI.getDiffPeaksRep.cMAF.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"

#ColIvsAB#
cat >"£{TMPDIR}/R.kmeans.Histogramme.R" <<EOF
   library(ggplot2)
   library(reshape)
  #Data 1
```

```
   data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ChIP_CK_cMAF_AB.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
  #Data 2
  data <- read.table("£PEAKDIR/M7.ColIvsAB.getDiffPeaksRep.ChIP_CK_cMAF_Col1.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪  check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result2 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
  #Start plot
   p <- ggplot(result, aes(d,m)) +
      geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
      geom_line(data = result, colour = "£colAB") +
      geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
      geom_line(data=result2, colour = "£colCol1")+
      theme_bw(base_size=12) +
      xlab("Distance from peak center") +
      ylab("ChIP-seq coverage") +
      xlim(-500, 500) +
      theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪  element_blank(), panel.grid.major = element_blank())
   pdf(file="£FIGURESDIR/M7.ColIvsAB.getDiffPeaksRep.cMAF.hist.pdf", height=4, width=4)
   plot(p)
   dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.R"
R < "£{TMPDIR}/R.kmeans.Histogramme.R"  --no-save
rm "£{TMPDIR}/R.kmeans.Histogramme.R"


      ##################################################################
      ### Selecting clusters of ATAC Peaks based on Kmeans Clustering ###
      ##################################################################

##################################
### Clustering Col vs AB Peaks ###
##################################

### Annotating ATAC data on differnetial ATAC Peaks ###

annotatePeaks.pl $ColIvsABPeaks hg38 -size 300 -d $TAGDIRSETS -nogene -noann -cpu 6 >
↪  $PEAKDIRCOL1/M7_ColIvsAB_getDiffPeaksRep.ATACann.txt

## Selecting columns to cluster by transcription factor binding ##

tail -n +2 $PEAKDIRCOL1/M7_ColIvsAB_getDiffPeaksRep.ATACann.txt | cut -f1-6,18-19 > $TMPDIR/M7_ColIvsAB_getDiffPeaksRep.ATACann.txt

echo £'ID\tChr\tStart\tEnd\tStrand\tno\tChIP_MAF_AB\tChIP_cMAF_Col' | cat - $TMPDIR/M7_ColIvsAB_getDiffPeaksRep.ATACann.txt >
↪  $PEAKDIRCOL1/M7_ColIvsAB_getDiffPeaksRep.ATACann.table.txt

### generate several Kmeans cluster solutions ###

_DATE=$(date +%s)

for ((i=4;i<=10;i++));do
cat >"£{TMPDIR}/R.ATAC.kmeans.£{_DATE}.R" <<EOF
gc()
data <- read.delim("£PEAKDIRCOL1/M7_ColIvsAB_getDiffPeaksRep.ATACann.table.txt", row.names="ID")

#select needed columns
d <- data.matrix(data[,seq(6,7)])
```

```
#cluster with Kmeans
fit <- kmeans(d, £i,iter.max=500)
clusd <- data.frame(data, fit\£cluster)
sorted <- clusd[order(-clusd\£fit.cluster), ]
pos <- sorted[,c(1:4,8)]
write.table(pos, file="£{TMPDIR}/M7_ColIvsAB.ATACann.kmeans.£i.txt", sep="\t", col.names=FALSE, quote=FALSE)
EOF
chmod 750 "£{TMPDIR}/R.ATAC.kmeans.£{_DATE}.R"
R < £{TMPDIR}/R.ATAC.kmeans.£{_DATE}.R --no-save
rm £{TMPDIR}/R.ATAC.kmeans.£{_DATE}.R
done

### ghist plot of kmeans clustered data ###

_DATE=$(date +%s)
for KMEANS in £{TMPDIR}/M7_ColIvsAB.ATACann.kmeans.*.txt
do
        NAMEBASE=£{KMEANS##*/}
        NAME=£{NAMEBASE%.*}
        cat >"£{TMPDIR}/ATACghist.£{NAME}.£{_DATE}.sh" <<EOF
#!/bin/bash
#setting homer environment
export
↪  PATH=/misc/software/package/RBioC/3.4.3/bin:/misc/software/package/perl/perl-5.26.1/bin:/misc/software/ngs/samtools/samtools-1.6/bin:/misc/software/ngs/home
export PATH
cd £{TMPDIR}
annotatePeaks.pl £{KMEANS} hg38 -size 1000 -hist 25 -ghist -d £{TAGDIRSETMAF} > "£{TMPDIR}/£{NAME}.ghist.txt"
plotGHIST.sh -f "£{TMPDIR}/£{NAME}.ghist.txt" -h 600 -w 3000 -d "£{FIGURESDIR}" -n £{NAME} -m 15 -c dodgerblue4
EOF
        chmod 750 "£{TMPDIR}/ATACghist.£{NAME}.£{_DATE}.sh"
        echo "plotting ghist for £{NAME}"
        screen -dm -S £{NAME} bash -c "bash £{TMPDIR}/ATACghist.£{NAME}.£{_DATE}.sh"
done

# loop to check when screen sessions are done
#---------------------------------------------
for KMEANS in £{TMPDIR}/M7_ColIvsAB.ATACann.kmeans.*.txt ; do
        NAMEBASE=£{KMEANS##*/}
        NAME=£{NAMEBASE%.*}
while [ true ]; do     # Endless loop.
  pid=`screen -S £{NAME} -Q echo '£PID'`  # Get a pid.
  if [[ $pid = *"session"* ]] ; then # If there is none,
    echo -e "\tFinished sample £{NAME}"
    break                # Test next one.
  else
    sleep 10           # Else wait.
  fi
        done
done
#---------------------------------------------

## Select cluster 4 solution ##

cp £{TMPDIR}/M7_ColIvsAB.ATACann.kmeans.4.txt $KMEANDIR/M7_ColIvsAB.ATACann.kmeans.4.txt

### Selecting the different clusters ###

for ((i=1;i<=4;i++));do
        awk -v "key=$i" '£6 == key' £{KMEANDIR}/M7_ColIvsAB.ATACann.kmeans.4.txt > "£{KMEANDIR}/ColIvsAB.ATACann.4.£{i}.txt"
done

# determine cluster sizes:

for ((i=1;i<=4;i++));do
        echo "cluster.$i size: $(wc -l <£{KMEANDIR}/ColIvsAB.ATACann.4.£{i}.txt)"
done

#cluster.1 size: 587
#cluster.2 size: 1680
#cluster.3 size: 47
#cluster.4 size: 265

cp £{KMEANDIR}/ColIvsAB.ATACann.4.3.txt £{KMEANDIR}/ColIvsAB.ATACann.4.1.sorted.txt
cp £{KMEANDIR}/ColIvsAB.ATACann.4.4.txt £{KMEANDIR}/ColIvsAB.ATACann.4.2.sorted.txt
cp £{KMEANDIR}/ColIvsAB.ATACann.4.1.txt £{KMEANDIR}/ColIvsAB.ATACann.4.3.sorted.txt
cp £{KMEANDIR}/ColIvsAB.ATACann.4.2.txt £{KMEANDIR}/ColIvsAB.ATACann.4.4.sorted.txt

cat £{KMEANDIR}/ColIvsAB.ATACann.4.1.sorted.txt £{KMEANDIR}/ColIvsAB.ATACann.4.2.sorted.txt
↪  £{KMEANDIR}/ColIvsAB.ATACann.4.3.sorted.txt £{KMEANDIR}/ColIvsAB.ATACann.4.4.sorted.txt >
↪  $KMEANDIR/ColIvsAB.ATACann.kmeans.4.sorted.txt

## Ghist plot over sorted file ##
```

```
TAGDIRSETMAF="$TAGDIR/ATAC_CK_Mono $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIRChIP/MAF/ChIP_CK_cMAF_AB $TAGDIRChIP/MAF/ChIP_CK_cMAF_Col1 $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_Col4"

TAGDIRSETghist="$TAGDIR/ATAC_CK_Mono $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIR/ATAC_CK_Fibronectin_M7 $TAGDIR/ATAC_CK_Laminin_M7 $TAGDIR/ATAC_CK_Col4_Lam $TAGDIR/ATAC_CK_EGR2KD_CTRL
↪  $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 $TAGDIRChIP/MAF/ChIP_CK_cMAF_AB
↪  $TAGDIRChIP/MAF/ChIP_CK_cMAF_Col1 $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_Col4"

annotatePeaks.pl $KMEANDIR/ColIvsAB.ATACann.kmeans.4.sorted.txt hg38 -size 1000 -hist 25 -ghist -d £{TAGDIRSETghist} >
↪  "£{TMPDIR}/ColIvsAB.ATACann.kmeans.4.sorted.ghist.txt"
plotGHIST.sh -f "£{TMPDIR}/ColIvsAB.ATACann.kmeans.4.sorted.ghist.txt" -h 250 -w 2000 -d "£{FIGURESDIR}" -n
↪  ColIvsAB.ATACann.kmeans.4.sorted -m 20 -c dodgerblue4

       ##################################################################################
       ### Combining ATAC Data from Collagen and AB Macrophages with EGR2 Knock Down ###
       ##################################################################################

### Annotating ATAC data on differnetial ATAC Peaks ###

TAGDIRSETATAC="$TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIR/ATAC_CK_Fibronectin_M7 $TAGDIR/ATAC_CK_Laminin_M7 $TAGDIR/ATAC_CK_163_ColIV_Laminin $TAGDIR/ATAC_CK_EGR2KD_CTRL
↪  $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665"

annotatePeaks.pl $ABvsColIPeaks hg38 -size 300 -d $TAGDIRSETATAC -nogene -noann -cpu 6 >
↪  $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.EGR2KD.ATACann.txt

#clustering regions by CTRL, Mock and Knock Down:

tail -n +2 $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.EGR2KD.ATACann.txt | cut -f1-6,14-17 >
↪  $TMPDIR/M7_ABvsColI_getDiffPeaksRep.EGR2KD.ATACann.txt

echo £'ID\tChr\tStart\tEnd\tStrand\tno\tATAC_CTRL\tATAC_Mock\tATAC_1132\tATAC_2665' | cat -
↪  $TMPDIR/M7_ABvsColI_getDiffPeaksRep.EGR2KD.ATACann.txt > $PEAKDIRAB/M7_ABvsColI_getDiffPeaksRep.EGR2KD.ATACann.table.txt

### generate several Kmeans cluster solutions ###

_DATE=$(date +%s)

for ((i=3;i<=10;i++));do
cat >"£{TMPDIR}/R.ATAC.EGR2KD.kmeans.£{_DATE}.R" <<EOF
gc()
data <- read.delim("£{PEAKDIRAB}/M7_ABvsColI_getDiffPeaksRep.EGR2KD.ATACann.table.txt", row.names="ID")

#select needed columns
d <- data.matrix(data[,seq(6,9)])

#cluster with Kmeans
fit <- kmeans(d, £i,iter.max=500)
clusd <- data.frame(data, fit\£cluster)
sorted <- clusd[order(-clusd\£fit.cluster), ]
pos <- sorted[,c(1:4,10)]
write.table(pos, file="£{TMPDIR}/M7_ABvsColI.EGR2KD.ATACann.kmeans.£i.txt", sep="\t", col.names=FALSE, quote=FALSE)
EOF
chmod 750 "£{TMPDIR}/R.ATAC.EGR2KD.kmeans.£{_DATE}.R"
R < £{TMPDIR}/R.ATAC.EGR2KD.kmeans.£{_DATE}.R --no-save
rm £{TMPDIR}/R.ATAC.EGR2KD.kmeans.£{_DATE}.R
done

### ghist plot of kmeans clustered data ###

_DATE=$(date +%s)
for KMEANS in £{TMPDIR}/M7_ABvsColI.EGR2KD.ATACann.kmeans.*.txt
do
        NAMEBASE=£{KMEANS##*/}
        NAME=£{NAMEBASE%.*}
        cat >"£{TMPDIR}/£{NAME}.£{_DATE}.sh" <<EOF
#!/bin/bash
#setting homer environment
export
↪  PATH=/misc/software/package/RBioC/3.4.3/bin:/misc/software/package/perl/perl-5.26.1/bin:/misc/software/ngs/samtools/samtools-1.6/bin:/misc/software/ngs/homer/v4.
export PATH
cd £{TMPDIR}
annotatePeaks.pl £{KMEANS} hg38 -size 1000 -hist 25 -ghist -d £{TAGDIRSETATAC} > "£{TMPDIR}/£{NAME}.ghist.txt"
plotGHIST.sh -f "£{TMPDIR}/£{NAME}.ghist.txt" -h 6000 -w 6500 -d "£{FIGURESDIR}" -n £{NAME} -m 15 -c dodgerblue4
EOF
        chmod 750 "£{TMPDIR}/£{NAME}.£{_DATE}.sh"
        echo "plotting ghist for £{NAME}"
        screen -dm -S £{NAME} bash -c "bash £{TMPDIR}/£{NAME}.£{_DATE}.sh"
done

# loop to check when screen sessions are done
#---------------------------------------------
```

```
for KMEANS in £{TMPDIR}/M7_ABvsColI.EGR2KD.ATACann.kmeans.*.txt ; do
        NAMEBASE=£{KMEANS##*/}
        NAME=£{NAMEBASE%.*}
while [ true ]; do     # Endless loop.
  pid=`screen -S £{NAME} -Q echo '£PID'`  # Get a pid.
  if [[ $pid = *"session"* ]] ; then # If there is none,
    echo -e "\tFinished sample £{NAME}"
    break               # Test next one.
  else
    sleep 10           # Else wait.
  fi
        done
done
#--------------------------------------------

### Kmeans 8 seems to give nice results ###

cp $TMPDIR/M7_ABvsColI.EGR2KD.ATACann.kmeans.8.txt $KMEANDIR/M7_ABvsColI.EGR2KD.ATACann.kmeans.8.txt

### Selecting the different clusters ###

for ((i=1;i<=8;i++));do
        awk -v "key=$i" '£6 == key' £{KMEANDIR}/M7_ABvsColI.EGR2KD.ATACann.kmeans.8.txt >
        ↪  "£{KMEANDIR}/ABvsColI.EGR2KDann.8.£{i}.txt"
done

# determine cluster sizes:

for ((i=1;i<=8;i++));do
        echo "cluster.$i size: $(wc -l <£{KMEANDIR}/ABvsColI.EGR2KDann.8.£{i}.txt)"
done

#cluster.1 size: 2358
#cluster.2 size: 7617
#cluster.3 size: 263
#cluster.4 size: 706
#cluster.5 size: 13454
#cluster.6 size: 47
#cluster.7 size: 1381
#cluster.8 size: 4471

#################################################
### Histogramms over subclusters from kmenas 8 ###
#################################################

for ((i=1;i<=8;i++));do
  NAMEBASE=£{£{KMEANDIR}/ABvsColI.EGR2KDann.8.£{i}.txt##*/}
  NAME=£{NAMEBASE%.*}
        annotatePeaks.pl £{KMEANDIR}/ABvsColI.EGR2KDann.8.£{i}.txt hg38 -size 1000 -hist 25 -ghist -d
        ↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 > £{KMEANDIR}/ABvsColI.EGR2KDann.£{i}.£{NAME}.ghist.txt
  _DATE=$(date +%s)
        cat >"£{TMPDIR}/R.kmeans.£{i}.£{NAME}.£{_DATE}.R" <<EOF
     library(ggplot2)
     library(reshape)
     data <- read.table("£{KMEANDIR}/ABvsColI.EGR2KDann.£{i}.£{NAME}.ghist.txt", header=T, sep="\t", stringsAsFactors=FALSE,
↪   check.names = FALSE)
     tdata <- t(data[,-1])
     m <- apply(tdata, 1, mean)
     s <- apply(tdata, 1, sd)
     n <- nrow(as.matrix(tdata))
     #calculating the confidence interval
     error <- qnorm(0.975)*s/sqrt(n)
     #upper and lower boundaries
     ci975 <- m+error
     ci025 <- m-error
     d <- as.numeric(as.character(rownames(tdata)))
     result <- data.frame(cbind(d,m, s,ci975,ci025))
     colband <- "gray"
     colline <- "blue"
     p <- ggplot(result, aes(d,m)) +
               geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
               geom_line(colour = colline) +
               theme_bw(base_size=12) +
               xlab("Distance from peak center") +
               ylab("ChIPseq coverage") +
               xlim(-500, 500) #+
     #          ylim(0, 3)
     pdf(file="£{KMEANDIR}/£{NAME}.£{i}.hist.pdf", height=4, width=4)
     plot(p)
     dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.£{i}.£{NAME}.£{_DATE}.R"
```

```
R < "£{TMPDIR}/R.kmeans.£{i}.£{NAME}.£{_DATE}.R"  --no-save
rm "£{TMPDIR}/R.kmeans.£{i}.£{NAME}.£{_DATE}.R"
   done

# --> order of clusters: 6,3,4,7,1,8,2,5 ###

#combine cluster 6 and 3, since 6 only contains 47 Peaks

cat £{KMEANDIR}/ABvsColI.EGR2KDann.8.6.txt £{KMEANDIR}/ABvsColI.EGR2KDann.8.3.txt >
↪  £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.1.txt
cp £{KMEANDIR}/ABvsColI.EGR2KDann.8.4.txt £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.2.txt
cp £{KMEANDIR}/ABvsColI.EGR2KDann.8.7.txt £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.3.txt
cp £{KMEANDIR}/ABvsColI.EGR2KDann.8.1.txt £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.4.txt
cp £{KMEANDIR}/ABvsColI.EGR2KDann.8.8.txt £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.5.txt
cp £{KMEANDIR}/ABvsColI.EGR2KDann.8.2.txt £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.6.txt
cp £{KMEANDIR}/ABvsColI.EGR2KDann.8.5.txt £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.7.txt

cat £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.1.txt £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.2.txt
↪  £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.3.txt £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.4.txt
↪  £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.5.txt £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.6.txt
↪  £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.7.txt > $KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.txt

##############################
### Ghist over sorted file ###
##############################

TAGDIRSETATAC="$TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIR/ATAC_CK_Fibronectin_M7 $TAGDIR/ATAC_CK_Laminin_M7 $TAGDIR/ATAC_CK_Col4_Lam $TAGDIR/ATAC_CK_EGR2KD_CTRL
↪  $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665"
annotatePeaks.pl $KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.txt hg38 -size 1000 -hist 25 -ghist -d £{TAGDIRSETATAC} >
↪  "£{TMPDIR}/ABvsColI.EGR2KDann.ATACann.kmeans.sorted.ghist.txt"
plotGHIST.sh -f £{TMPDIR}/ABvsColI.EGR2KDann.ATACann.kmeans.sorted.ghist.txt -h 6000 -w 6500 -d £{FIGURESDIR} -n
↪  ABvsColI.kmeans8.sorted.ATACann -m 20 -c dodgerblue4

ATACSETS="$TAGDIR/ATAC_CK_Mono $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIR/ATAC_CK_Fibronectin_M7 $TAGDIR/ATAC_CK_Laminin_M7 $TAGDIR/ATAC_CK_163_ColIV_Laminin $TAGDIR/ATAC_CK_EGR2KD_CTRL
↪  $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665"

annotatePeaks.pl $KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.txt hg38 -size 1000 -hist 25 -ghist -d £{ATACSETS} >
↪  "£{TMPDIR}/ABvsColI.EGR2KDann.kmeans.ghist.txt"
plotGHIST.sh -f "£{TMPDIR}/ABvsColI.EGR2KDann.kmeans.ghist.txt" -h 6000 -w 6500 -d "£{FIGURESDIR}" -n ABvsColI.EGR2KDann.kmeans -m
↪  15 -c dodgerblue4

TAGDIRSETghist="$TAGDIR/ATAC_CK_Mono $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7
↪  $TAGDIR/ATAC_CK_Fibronectin_M7 $TAGDIR/ATAC_CK_Laminin_M7 $TAGDIR/ATAC_CK_Col4_Lam $TAGDIR/ATAC_CK_EGR2KD_CTRL
↪  $TAGDIR/ATAC_CK_EGR2KD_Mock $TAGDIR/ATAC_CK_EGR2KD_1132 $TAGDIR/ATAC_CK_EGR2KD_2665 $TAGDIRChIP/MAF/ChIP_CK_cMAF_AB
↪  $TAGDIRChIP/MAF/ChIP_CK_cMAF_Col1 $TAGDIRChIP/MAF/ChIP_CK_cMAF_M6_MAC_Col4"

annotatePeaks.pl $KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.txt hg38 -size 1000 -hist 25 -ghist -d £{TAGDIRSETghist} >
↪  "£{TMPDIR}/ABvsColI.kmeans8.sorted.cMAFEGR2.ghist.txt"
plotGHIST.sh -f "£{TMPDIR}/ABvsColI.kmeans8.sorted.cMAFEGR2.ghist.txt" -h 3030 -w 6500 -d "£{FIGURESDIR}" -n
↪  ABvsColI.cMAFEGR2.kmeans8.sorted -m 15 -c dodgerblue4

############################################
### Histogramme with confidence interval ###
############################################

for ((i=1;i<=7;i++)); do
  for TF in £{TAGDIRSETATAC}; do
    NAMEBASE=£{TF##*/}
          NAME=£{NAMEBASE%.*}
    annotatePeaks.pl £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.$i.txt hg38 -size 2000 -hist 25 -ghist -d $TF >
    ↪  $KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.$i.$NAME.ghist.txt
   done
done

### Histogram of ATAC data ###

for ((i=1;i<=7;i++)); do
  cat > "£{TMPDIR}/R.kmeans.Histogramme.$i.R" << EOF
   library(ggplot2)
   library(reshape)
  #Data 1
   data <- read.table("£KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.£i.ATAC_CK_AB_TcPlastic_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
```

```
 ci975 <- m+error
 ci025 <- m-error
 d <- as.numeric(as.character(rownames(tdata)))
 result <- data.frame(cbind(d,m, s,ci975,ci025))
 colband <- "gray"
 #Data 4
  data <- read.table("£KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.£i.ATAC_CK_EGR2KD_CTRL.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
  tdata <- t(data[,-1])
  m <- apply(tdata, 1, mean)
  s <- apply(tdata, 1, sd)
  n <- nrow(as.matrix(tdata))
  #calculating the confidence interval
  error <- qnorm(0.975)*s/sqrt(n)
  #upper and lower boundaries
  ci975 <- m+error
  ci025 <- m-error
  d <- as.numeric(as.character(rownames(tdata)))
  result4 <- data.frame(cbind(d,m, s,ci975,ci025))
  colband <- "gray"
#Data 5
   data <- read.table("£KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.£i.ATAC_CK_EGR2KD_Mock.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result5 <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
#Data 2
data <- read.table("£KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.£i.ATAC_CK_Collagen_I_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
tdata <- t(data[,-1])
m <- apply(tdata, 1, mean)
s <- apply(tdata, 1, sd)
n <- nrow(as.matrix(tdata))
#calculating the confidence interval
error <- qnorm(0.975)*s/sqrt(n)
#upper and lower boundaries
ci975 <- m+error
ci025 <- m-error
d <- as.numeric(as.character(rownames(tdata)))
result2 <- data.frame(cbind(d,m, s,ci975,ci025))
colband <- "gray"
#Data 3
data <- read.table("£KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.£i.ATAC_CK_Collagen_IV_M7.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
tdata <- t(data[,-1])
m <- apply(tdata, 1, mean)
s <- apply(tdata, 1, sd)
n <- nrow(as.matrix(tdata))
#calculating the confidence interval
error <- qnorm(0.975)*s/sqrt(n)
#upper and lower boundaries
ci975 <- m+error
ci025 <- m-error
d <- as.numeric(as.character(rownames(tdata)))
result3 <- data.frame(cbind(d,m, s,ci975,ci025))
colband <- "gray"
#Data 6
data <- read.table("£KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.£i.ATAC_CK_EGR2KD_1132.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
tdata <- t(data[,-1])
m <- apply(tdata, 1, mean)
s <- apply(tdata, 1, sd)
n <- nrow(as.matrix(tdata))
#calculating the confidence interval
error <- qnorm(0.975)*s/sqrt(n)
#upper and lower boundaries
ci975 <- m+error
ci025 <- m-error
d <- as.numeric(as.character(rownames(tdata)))
result6 <- data.frame(cbind(d,m, s,ci975,ci025))
colband <- "gray"
#Data 7
```

```
   data <- read.table("£KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.£i.ATAC_CK_EGR2KD_2665.ghist.txt", header=T, sep="\t",
↪  stringsAsFactors=FALSE, check.names = FALSE)
   tdata <- t(data[,-1])
   m <- apply(tdata, 1, mean)
   s <- apply(tdata, 1, sd)
   n <- nrow(as.matrix(tdata))
   #calculating the confidence interval
   error <- qnorm(0.975)*s/sqrt(n)
   #upper and lower boundaries
   ci975 <- m+error
   ci025 <- m-error
   d <- as.numeric(as.character(rownames(tdata)))
   result7 <- data.frame(cbind(d,m, s,ci975,ci025))
   colband <- "gray"
   #Start plot
    p <- ggplot(result, aes(d,m)) +
       geom_ribbon(aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
       geom_line(data = result, colour = "£colAB") +
       geom_ribbon(data = result4, aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
       geom_line(data = result4, colour = "£colCTRL") +
       geom_ribbon(data = result5, aes(x=d, ymax=ci975, ymin=ci025), fill=colband, alpha=.25) +
       geom_line(data = result5, colour = "£colCTRL") +
       geom_ribbon(data = result2, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
       geom_line(data=result2, colour = "£colCol1")+
       geom_ribbon(data = result3, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
       geom_line(data=result3, colour = "£colCol4")+
       geom_ribbon(data = result6, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
       geom_line(data=result6, colour = "£col1132")+
       geom_ribbon(data = result7, aes(d,m, ymax=ci975, ymin=ci025), fill=colband, alpha=.25)+
       geom_line(data=result7, colour = "£col2665")+
       theme_bw(base_size=12) +
       xlab("Distance from peak center") +
       ylab("ATAC-seq coverage") +
       xlim(-500, 500) +
       theme(axis.text = element_text(family = "ArialMT"), axis.title = element_text(family = "ArialMT"), panel.grid.minor =
↪  element_blank(), panel.grid.major = element_blank())
    pdf(file="£KMEANDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.£i.hist.pdf", height=4, width=4)
    plot(p)
    dev.off()
EOF
chmod 750 "£{TMPDIR}/R.kmeans.Histogramme.$i.R"
R < £{TMPDIR}/R.kmeans.Histogramme.$i.R  --no-save
rm £{TMPDIR}/R.kmeans.Histogramme.$i.R
done

##############################################
#### Footprint for EGR2 over subclusters ####
##############################################

for ((i=1;i<=7;i++)); do
  annotatePeaks.pl £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.$i.txt hg38 -center
  ↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/homerResults/motif4.motif -size given >
  ↪  £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.$i.centered.EGR2.txt
  annotatePeaks.pl £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.$i.centered.EGR2.txt hg38 -size 630 -hist 1 -len 1 -d
  ↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
  ↪  $FOOTDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.$i.centered.EGR2.hist.txt
done

#Last cluster alone
annotatePeaks.pl £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.7.txt hg38 -center
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/homerResults/motif4.motif -size given >
↪  £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.7.centered.EGR2.txt
#results in 3770 Peaks containing motif

annotatePeaks.pl £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.7.centered.EGR2.txt hg38 -size 630 -hist 1 -len 1 -d
↪  $TAGDIR/ATAC_CK_AB_TcPlastic_M7 $TAGDIR/ATAC_CK_Collagen_I_M7 $TAGDIR/ATAC_CK_Collagen_IV_M7 >
↪  $FOOTDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.7.centered.EGR2.hist.txt

for TF in £{FOOTDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.*.centered.EGR2.hist.txt; do
  NAMEBASE=£{TF##*/}
  NAME=£{NAMEBASE%.*}
  cat > "/loctmp/R.Footprints.£{_DATE}.£{NAME}.R" <<EOF
  library(ggplot2)
  library(extrafont)

  z <- read.delim(file = "£FOOTDIR/£{NAME}.txt", sep = "\t")
  colnames(z) <- c("Distance", "ATAC_AB_Coverage", "ATAC_AB_pos", "ATAC_AB_neg", "ATAC_Col1_Coverage", "ATAC_Col1_pos",
↪  "ATAC_Col1_neg")
  z\£Distance=z\£Distance-5

  p2 <- ggplot(z, aes(x= z[,1])) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= z[,5]), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= z[,5]), alpha = 0.2, fill = "£colCol4") +
  labs(x="", y = "ATAC Tag count")+
  xlim(-300,300) +
  scale_x_continuous(breaks = c(-300, -150,0,150,300), labels = c(-300,-150, 0, 150, 300)) +
  geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
theme(axis.text = element_text(size = 18, colour = "black", family = "ArialMT"), axis.title = element_text(size = 20, colour = "black",
↪  family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major = element_blank())

  pdf(file = "£FOOTDIR/Footprint.£{NAME}.pdf", width = 6, height = 6)
  plot(p2)
  dev.off()

  ### Zoomed in region of Footprint ###

  p2 <- ggplot(z, aes(x= z[,1])) + theme_bw() +
geom_line(data = z, aes(y= z[,2]), col = "£colAB", size = 0.25) +
geom_area(data = z, aes(y= z[,2]), fill = "£colAB", alpha = 0.4) +
geom_line(data = z, aes(y= z[,5]), col = "£colCol4", size = 0.25) +
geom_area(data = z, aes(y= z[,5]), alpha = 0.2, fill = "£colCol4") +
  labs(x="", y = "")+
  xlim(-10,10) +
  geom_vline(xintercept=c(-5,5), linetype="dashed", color="black", size=.25, alpha=.5) +
  theme(axis.text = element_text(size = 22, colour = "black", family = "ArialMT"), axis.title = element_text(size = 24, colour =
↪  "black", family = "ArialMT"), panel.grid.minor = element_blank(), panel.grid.major = element_blank())

  pdf(file = "£FOOTDIR/Footprint.£{NAME}.zoomedIn.pdf", width = 6, height = 6)
  plot(p2)
  dev.off()

EOF
  chmod 750 "/loctmp/R.Footprints.£{_DATE}.£{NAME}.R"
  R < /loctmp/R.Footprints.£{_DATE}.£{NAME}.R --no-save
  rm /loctmp/R.Footprints.£{_DATE}.£{NAME}.R
done

############################################################################
# Using Top 8 found motifs as known motifs in the motif finding algorithm #
############################################################################

for ((i=1;i<=7;i++)); do
  mkdir -p $MOTIFDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.$i
  findMotifsGenome.pl £{KMEANDIR}/ABvsColI.EGR2KDann.kmeans.8.sorted.$i.txt hg38 $MOTIFDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.$i
  ↪  -mknown $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/Top8_homerMotifs.motif -size 200 -nomotif -p 12 -h
done

DATE=$(date +%s)
for ((i=1;i<=7;i++)); do
  cat >> "$TMPDIR/knownMotifs.clusterABvsColI.$DATE.txt" <<EOF
£MOTIFDIR/ABvsColI.EGR2KDann.kmeans.8.sorted.£i/knownResults.txt
EOF
done

NAMES="Cluster1 Cluster2 Cluster3 Cluster4 Cluster5 Cluster6 Cluster7"
for NAME in £{NAMES[@]}; do
cat >>"£{TMPDIR}/knownMotifs.names.£{DATE}.txt" <<EOF
£{NAME}
EOF
done

## Creating necessary Table ##

mySummarizeMotifResults.pl $TMPDIR/knownMotifs.clusterABvsColI.$DATE.txt -namesFile £{TMPDIR}/knownMotifs.names.£{DATE}.txt -outDir
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep -hc -minp 0.000001 -minr 2 -limit 2

## Creating Table with short names for Plot ##

touch $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt
echo £'Motif.Name\tShort.Name' >> $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt
echo £'1-ACTTCCTCWTT,BestGuess:PB0058.1_Sfpi1_1/Jaspar(0.963)\tPU.1' >> $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt
echo £'2-VTTGCAWAAY,BestGuess:CEBPA/MA0102.3/Jaspar(0.925)\tCEBPA' >> $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt
echo £'3-GATGASTCAT,BestGuess:JunB(bZIP)/DendriticCells-Junb-ChIP-Seq(GSE36099)/Homer(0.992)\tJunB' >>
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt
echo £'8-GYGTGGGHGG,BestGuess:Egr2(Zf)/Thymocytes-Egr2-ChIP-Seq(GSE34254)/Homer(0.962)\tEgr2' >>
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt
echo £'9-KGGGAATTCCC,BestGuess:NFkB-p65(RHD)/GM12787-p65-ChIP-Seq(GSE19485)/Homer(0.917)\tNFkB' >>
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt
echo £'14-RYCATGTGRY,BestGuess:MITF(bHLH)/MastCells-MITF-ChIP-Seq(GSE48085)/Homer(0.903)\tbHLH' >>
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt
echo £'15-TCTATTTTTAGCT,BestGuess:Mef2d(MADS)/Retina-Mef2d-ChIP-Seq(GSE61391)/Homer(0.881)\tMef2d' >>
↪  $MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt
```

```
### create Plot in R ###

cat >"£{TMPDIR}/R.ballon.£{DATE}.R" <<EOF
library(ggplot2)
library(reshape2)
library(ggpubr)
r <- read.table("£MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/cleanedRatioTable.txt", header=T, sep="\t")
mr <- melt(r)
q <- read.table("£MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/cleanedqValueTable.txt", header=T, sep="\t")
mq <- melt(q)
s <- read.table("£MOTIFDIR/M7_ABvsColI_getDiffPeaksRep/short.motif.names.txt", header=T, sep="\t")
rq <- merge(mr,mq,by=c("Motif.Name","variable"))
head(rq)
table <- merge(s,rq,by="Motif.Name")
table\£logq <- log10(table\£value.y+0.00005)
table\£enr <- (table\£value.x+0.01)
head(table)
p <- ggballoonplot(table, x = "variable", y = "Short.Name", size = "enr", color="black", fill = "logq", size.range = c(4, 12), ggtheme
↪  = theme_bw()) +
  scale_fill_gradient2(low = "mediumblue", mid = "gray90",  high = "gray90", midpoint = -1) + scale_color_gradient2( low =
↪  "mediumblue", mid = "gray90",  high = "gray90", midpoint = -1) + scale_x_discrete(labels = c("Cluster 1", "Cluster 2", "Cluster 3",
↪  "Cluster 4", "Cluster 5", "Cluster 6", "Cluster 7")) +
  theme(axis.text = element_text(family = "ArialMT", colour = "black", size = 16), axis.title = element_text(family = "ArialMT", colour
↪  = "black", size = 18))
pdf(file="£{FIGURESDIR}/ballonplot.motifenrichment.ABvsColI.clusters.pdf", height=5, width=7, useDingbats=FALSE)
plot(p)
dev.off()
EOF
chmod 750 "£{TMPDIR}/R.ballon.£{DATE}.R"
R < £{TMPDIR}/R.ballon.£{DATE}.R  --no-save
rm £{TMPDIR}/R.ballon.£{DATE}.R
```

Plots showing correlation between collagen treatment and EGR2 knock down:

```
#!/usr/bin/env bash
#File: Intersection_Col_EGR2KD.sh by Claudia Kiesewetter on 1.10.2018
#Following Script: EGR2KD_DiffExpAnalysis.sh to look for overlaps between RNAseq analysis of collagen treated macrophages and cell
↪  containing EGR2 knock down
#Prerequisite: List of differnetially expressed genes by collagen treatment and running previous script in the same directory

#Setting the environment variables

PATH_R=/misc/software/package/RBioC/3.4.3/bin

export PATH=£{PATH_R}:£{PATH}
export PATH

AnalysisDir="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/EGR2KD"
COLDIR="/misc/data/analysis/project_MAC_CK/RNAseq_analysis/M1M3M6/Heatmaps"


### Setting colors for the different cell types ###

colMono="forestgreen"
colCol1="deeppink"
colCol4="violetred3"
colAB="slateblue3"
colLam="orchid1"
colFib="plum3"
colCol4Lam="hotpink"
colCTRL="dodgerblue3"
colMock="royalblue2"
col1132="gold2"
col2665="goldenrod3"

cd £{AnalysisDir}
_DATE=$(date +%s)

#Starting R script:

  cat > "/loctmp/R.IntersectionColEGR2KD.£{_DATE}.R" <<EOF
library(sqldf)
library(extrafont)
library(venneuler)
library(VennDiagram)
library(ggplot2)
library(ggpubr)
library(reshape2)

### Calculating overlap between Collagen treatment and EGR2KD ###

#Loading Datasets:

ColvsRest <- read.delim(file = "£COLDIR/Table_ManhattanWardZscores_AllSamples_pairwiseComparisons.logFC2.FDR0.05.txt", sep = "\t")

EGR2KD <- read.delim("£{AnalysisDir}/EGR2KD_AllSamples_Table_ManhattanWardZscores_stat_siRNAvsCTRL.txt", sep = "\t")

#Selecting genes from different clusters:
EGR2_Down <- sqldf("SELECT GeneSymbol FROM EGR2KD WHERE clusterID = '1'")
nrow(EGR2_Down)
EGR2_Up <- sqldf("SELECT GeneSymbol        FROM EGR2KD WHERE clusterID = '2'")
nrow(EGR2_Up)

Col_Up <- sqldf("SELECT GeneSymbol FROM ColvsRest WHERE clusterID = '1'")
colnames(Col_Up) <- c("GeneName")
nrow(Col_Up)

Col_Down <- sqldf("SELECT GeneSymbol FROM ColvsRest WHERE clusterID = '2'")
colnames(Col_Down) <- c("GeneName")
nrow(Col_Down)

#Calculating overlap
EGR2_Down_Col_Down <- sqldf("SELECT GeneSymbol FROM EGR2_Down INNER JOIN Col_Down ON GeneSymbol = GeneName")
nrow(EGR2_Down_Col_Down)

write.table(EGR2_Down_Col_Down, file = "£{AnalysisDir}/GeneSymbols_Intersection_EGR2KD_ColDown.txt", sep = "\t", row.names = FALSE,
↪  quote = FALSE)

EGR2_Up_Col_Up <- sqldf("SELECT GeneSymbol FROM EGR2_Up INNER JOIN Col_Up ON GeneSymbol = GeneName")
nrow(EGR2_Up_Col_Up)

write.table(EGR2_Up_Col_Up, file = "£{AnalysisDir}/GeneSymbols_Intersection_EGR2KD_ColUp.txt", sep = "\t", row.names = FALSE, quote =
↪  FALSE)

### Creating a Venn Diagrammm ###

#Overlap between Collagen and EGR2KD
#Down-regulated Genes

venn <- venneuler(c(ColD=791, EGR2D=509, "ColD&EGR2D"=329))
venn\£labels <- rep("", length(venn\£labels)) #without any labels, will be added manually later

pdf(file = "£{AnalysisDir}/Venn_ColDown_EGR2KD_Down.pdf", width = 8, height = 8)
par(family = "ArialMT")
plot(venn, col=c("£colCol1", "£col1132"))
dev.off()

#venn.diagram(x = list("Collagen" = Col_Down\£GeneName, "EGR2 KD" = EGR2_Down\£GeneSymbol), filename =
↪  "Venn_ColvsRest_EGR2KD_Down_ohneTitel2.png", imagetype = "png", fill = c("£colCol1", "£col1132"), fontfamily = "ArialMT",
↪  cat.fontfamily = "ArialMT", label.col = c("£colCol4"), cat.col = "white")

#Up-regulated Genes

venn <- venneuler(c(ColU=523, EGR2D=43, "ColD&EGR2D"=22))
venn\£labels <- rep("", length(venn\£labels)) #without any labels, will be added manually later

pdf(file = "£{AnalysisDir}/Venn_ColUp_EGR2KD_Up.pdf", width = 8, height = 8)
par(family = "ArialMT")
plot(venn, col=c("£colCol1", "£col1132"))
dev.off()

#venn.diagram(x = list("Collagen" = Col_Up\£GeneName, "EGR2 KD" = EGR2_Up\£GeneSymbol), filename =
↪  "Venn_ColvsRest_EGR2KD_Up_ohneTitel.png", imagetype = "png", fill = c("£colCol1", "£col2665"), fontfamily = "ArialMT",
↪  cat.fontfamily = "ArialMT", na = "none", ext.text = FALSE)

### Dotplot of all genes down-regulated by Collagen in EGR2 Knock down samples ###

data_box <- read.delim(file = "£{AnalysisDir}/Logcpm_rmBE_data_dotplots.txt", sep = "\t")

#create datasets for downregulated genes:
data_down <- sqldf("SELECT exp129_Mock, exp130_Mock, exp132_Mock, exp129_CTRL, exp130_CTRL, exp132_CTRL, exp129_1132, exp130_1132,
↪  exp132_1132, exp129_2665, exp130_2665, exp132_2665
FROM data_box
INNER JOIN Col_Down ON GeneSymbol = GeneName")
nrow(data_down)

data <- melt(data.matrix(data_down))
head(data)
```

```
#Calculating mean values for control samples and knock down samples:

data_down2 <- data.frame(rowMeans(data_down[,seq(1,6)]), rowMeans(data_down[, seq(7,12)]))
colnames(data_down2) <- c("CTRL", "siRNA")

#using the package reshape2 to melt dataset for dotplots:

data <- melt(data.matrix(data_down2))
head(data)

my_comparisons <- list(c("CTRL", "siRNA"))

p1 <- ggplot(data, aes(x = Var2, y = value)) + theme_classic() +
geom_violin(aes(fill = Var2), scale = "count", show.legend = FALSE) +
#geom_boxplot(aes(fill = Var2), width = 0.5)+
#geom_dotplot(aes(fill = Var2), binwidth = 0.15, binaxis = "y", stackdir = "center", stackratio = 0.85, show.legend = FALSE) +
stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = "crossbar", width = 0.7) +
scale_fill_manual(values=c("£colCTRL", "£col1132")) +
stat_compare_means(comparisons = my_comparisons, method = "wilcox.test", paired = TRUE, aes(label = ..p.signif..)) +
stat_compare_means(label.y = 16, label.x = 2, family = "ArialMT") +
labs(x = "", y = "mean Expression Values (log2)") +
theme(axis.text = element_text(family = "ArialMT", size = 20),
      axis.title = element_text(family = "ArialMT", size = 22),
      text = element_text(family = "ArialMT"),
      strip.text = element_text(family = "ArialMT"))
p1\£layers[[2]]\£aes_params\£textsize <- 16
p1\£layers[[3]]\£aes_params\£textsize <- 18

pdf(file = "£{AnalysisDir}/Violinplot_ColDownGenes_CTRLvssiRNA.pdf", width = 8, height = 8)
plot(p1)
dev.off()

EOF

chmod 750 "/loctmp/R.IntersectionColEGR2KD.£{_DATE}.R"
R < /loctmp/R.IntersectionColEGR2KD.£{_DATE}.R  --no-save
rm /loctmp/R.IntersectionColEGR2KD.£{_DATE}.R
```

Plots for miRNA-seq data:

```
#Script for Plots of qPCR data, from miRNA148a-3p inhbition

library(ggplot2)
library(RColorBrewer)
library(ggpubr)

data <- read.delim("qPCR.txt")
data$Experiment <- as.factor(data$Experiment)

#Dec2
p1 <- ggplot(data, aes(x = Type, y = Dec2))
p1 + theme_classic() +
  geom_dotplot(aes(fill = Experiment), binaxis = "y", binwidth = 0.07, stackdir = "center", show.legend = TRUE)+
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = "crossbar", width = 0.5) +
  scale_fill_manual(values = c("slateblue3", "deeppink")) +
  labs(title = "Dotplot of Dec2 Expression",
       subtitle = "Relative Analysis",
       y = "relative Expression normalized to GAPDH",
       x = "Cell Type") +
  theme(axis.text = element_text(colour = "gray40", size = 14),
        axis.title = element_text(colour = "black", size = 14, face = "bold"),
        plot.title = element_text(colour="black", size=18, face="bold"),
        plot.subtitle = element_text(colour = "black", size = 16))

my_comparisons <- list(c("AB CTRL", "AB Inh"), c("Col CTRL", "Col Inh"))

p1 <- ggplot(data, aes(x = Type, y = Dec2)) +
  theme_classic() +
  geom_dotplot(aes(fill = Type), binaxis = "y", stackdir = "center", show.legend = FALSE) +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = "crossbar", width = 0.5) +
  scale_x_discrete(labels = c("AB + CTRL", "AB + Inh.", "AB + Mock", "Col. I + CTRL", "Col. I + Inh.", "Col. I + Mock")) +
  scale_fill_manual(values = c("slateblue3", "slateblue3", "slateblue3", "deeppink", "deeppink","violetred", "forestgreen"))+
  theme(axis.text.y = element_text(size = 20, colour = "black"),
                  axis.title = element_text(size = 20),
                  title = element_text(colour="black", size=18, face="bold"),
                  axis.text.x = element_text(angle = 45, hjust = 1, size = 20, colour = "black")) +
  labs(x = " ", y = "Dec2 normalized to GAPDH")
```

```
png("Dotplot_qPCR_miR148KD.png", width = 700, height = 400)
plot(p1)
dev.off()
```

Plots for quantified Western Blot data:

```
#### Dotplots of Western Blot Quantifizierung ###

#mac
setwd("~/Documents/Doktorarbeit/Western_Blot_Quantifizierung")
library(ggplot2)
library(RColorBrewer)
library(ggpubr)

### Dec 2 Knock Down
Dec2KD <- read.delim("Summary_Western_Dec2KD.txt", sep = "\t")

p1 <- ggplot(Dec2KD, aes(x = Sample, y = Dec2.Aktin)) +
  theme_classic() +
  geom_dotplot(aes(fill = Experiment), binaxis = "y", stackdir = "center")+
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = "crossbar", width = 0.5)

Dec2KD$Sample <- factor(Dec2KD$Sample, levels = unique(Dec2KD$Sample))
p1 <- ggplot(Dec2KD, aes(x = Sample, y = Dec2.Aktin)) +
  theme_classic() +
  geom_dotplot(aes(fill = Sample), binaxis = "y", stackdir = "center", show.legend = FALSE) +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = "crossbar", width = 0.5) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("lightseagreen", "dodgerblue2", "indianred3","mediumvioletred","seagreen", "dodgerblue2",
 ↪  "indianred3","mediumvioletred", "seagreen", "dodgerblue2", "indianred3","mediumvioletred"))+
  theme(axis.text = element_text(size = 18), axis.title = element_text(size = 18)) +
  theme(title = element_text(colour="black", size=18, face="bold")) +
  labs(x = " ", y = "Dec2 normalized to Aktin")

png("Dotplot_Western_Dec2KD.png", width = 900, height = 600)
plot(p1)
dev.off()

###mir148a 3p Knock Down
miRKD <- read.delim("Summary_Western_miR148KD.txt", sep = "\t")
miRKD$Experiment <- as.factor(miRKD$Experiment)

p1 <- ggplot(miRKD, aes(x = Sample, y = Dec2.Aktin)) +
  theme_classic() +
  geom_dotplot(aes(fill = Experiment), binaxis = "y", stackdir = "center")+
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = "crossbar", width = 0.5)

my_comparisons <- list(c("AB ctrl", "AB inh"), c("col ctrl", "col inh"))

p1 <- ggplot(miRKD, aes(x = Sample, y = Dec2.Aktin)) +
  theme_classic() +
  geom_dotplot(aes(fill = Sample), binaxis = "y", stackdir = "center", show.legend = FALSE) +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = "crossbar", width = 0.5) +
  scale_x_discrete(labels = c("AB + CTRL", "AB + Inh.", "AB + Mock", "Col. I + CTRL", "Col. I + Inh.", "Col. I + Mock",
 ↪  "Monocytes")) +
  scale_fill_manual(values = c("slateblue3", "slateblue3", "slateblue3", "deeppink", "deeppink","violetred", "forestgreen"))+
    theme(axis.text.y = element_text(size = 20, colour = "gray40"),
                 axis.title = element_text(size = 20),
                 title = element_text(colour="black", size=18, face="bold"),
                 axis.text.x = element_text(angle = 45, hjust = 1, size = 20, colour = "black")) +
  labs(x = " ", y = "Dec2 normalized to Actin")

png("Dotplot_Western_miR148KD.png", width = 700, height = 400)
plot(p1)
dev.off()
```

Acknowledgements

Als erstes möchte ich mich herzlich bei Prof. Dr. Michael Rehli für die Ermöglichung dieser Arbeit und die großartige Unterstützung bedanken. Vielen Dank für die beständige Hilfe bei allen Problemen und die vielen, anregenden Diskussionen.

Vielen herzlichen Dank an die Arbeitsgruppen Kreutz und Hoffmann für die wertvolle Zusammenarbeit im H1. Vielen Dank an Prof. Dr. Marina Kreutz für die anregenden Diskussionen im Labor Seminar und an PD Dr. Petra Hoffmann für ihre Hilfe beim Erstellen unseres FACS Panels und ihrer Hilfe bei der Auswertung der erhaltenen Daten.

Ich möchte mich außerdem bei allen Mitgliedern der AG Rehli für ihre tatkräftige Unterstützung im Labor bedanken, besonders wenn es mal nicht so gut lief. Vielen herzlichen Dank Claudi für dein offenes Ohr, danke das du dir immer für mich Zeit genommen hast und ganz viel Schoki mit mir gefuttert hast. Luci und Johanna vielen Dank für eure wunderbare Hilfe mit den Experimenten im Labor. Vielen Dank für alles!

I was lucky to work with great people. My hearfelt thanks to our small lunch group, it was wonderful to relax with you. Thanks Carina for your kind help with all my questions about FACS and the great fun we had in- and outside of the lab. Thank you Sakhila for your creating such a great athmosphere and Eli, thanks for making me laugh. Thank you Karina for all your wonderful help with the experiments and the wonderful time we had. Thank you for everything!

Außerdem möchte ich mich bei all meinen wunderbaren Freunden bedanken, die mich durch die letzten Jahre begleitet haben. Vielen Dank an Robert, Martin und Katja, für die wunderbaren, spontanen Abende zusammen. Unsere unvergesslichen Wanderungen haben mir immer wieder Kraft gegeben weiter zu machen.

Meiner Familie danke ich von Herzen fur ihre Liebe und Unterstützung. Danke Mama & Papa, ohne Euch ware dieses Studium nie möglich gewesen! Vielen Dank für eure aufbauenden Worte und euer Vertrauen, durch euch hatte ich stets den Mut weiter zu machen. Vielen Dank!

Matthias du bist der wunderbarste Mensch den ich kenne. Vielen Dank für deine Liebe, deine unendliche Geduld mit mir und meinen Nerven und das du mir immer wieder Kraft zum weiter machen gegeben hast. Vielen Dank für Alles!