



universität
wien

MASTERARBEIT

Titel der Masterarbeit

„Common Network Structure in the
Regulatory Core of Trait Development“

verfasst von

Kristina Todtova

angestrebter akademischer Grad

Master of Science (MSc)

Wien, 2014

Studienkennzahl lt. Studienblatt:

A 066 834

Studienrichtung lt. Studienblatt:

Masterstudium Molekulare Biologie

Betreut von:

Univ.-Prof. Mag. Dr. Thomas Rattei

DANKSAGUNG

Ich möchte mich bei allen bedanken, die das Zustandekommen dieser Arbeit unterstützt haben. Allen voran möchte ich meinem Betreuer-Team danken, Stefanie Widder, Mihaela Pavlicev und Thomas Rattei. Sie haben mich in allen Lagen unterstützt, und ihr Feedback war zu jedem Zeitpunkt sehr wertvoll für mich. Ich möchte Ihnen auch für die vielen spannenden und erhellenden Diskussionen zur meiner Arbeit, aber auch zu vielen anderen Themen danken.

Ich bedanke mich auch bei allen Mitgliedern der Division for Computational Systems Biology, von denen jeder dafür gesorgt hat, dass die Zeit während meinem Masterarbeitspraktikum eine sehr schöne und inspirierende war.

Zuletzt möchte ich meinen Eltern, meinem lieben Bruder Juraj, meinen Freundinnen Babsi und Kat, sowie meinem Freund Michl dafür danken, dass sie immer für mich da waren und mich unterstützt haben.

ABSTRACT

Gene regulatory networks (GRN) play an important role in controlling the development and maintenance of traits such as body parts, organs, tissues or cell types. Small recurring sub-graphs, so-called network motifs, have been found in GRN at higher frequencies than in random networks, suggesting they could implement biologically relevant functions. Additionally, recent studies have implicated that network topology may be involved in the regulation of traits. Based on these findings, we investigate whether specific topological patterns can be associated with trait regulation.

We characterize the global and local structure of twelve human cell type-specific GRN and six mouse tissue-specific GRN. We also compare their structure and interaction repertoire and examine if network motifs co-occur with trait-determining regulators / transcription factors (TFs). For this purpose we have developed a new method to reconstruct cell type- and tissue-specific GRN from predicted TF binding sites in regulatory regions, expression screens (from RNA-seq) and binding motif occupancy profiles (from DNaseI-seq / DNaseI footprinting). By integrating prediction with different data types, we limit false-positive binding site discovery and identify the core active sub-network of each particular trait.

Our results propose that certain network motifs, especially the feed-forward loop could be functional in trait regulation. Furthermore, we show that the interaction repertoire can be used to classify GRN with related functions. Our study also highlights areas that would benefit from methodical improvement and standards such as TF binding site models and DNaseI footprinting algorithms. Finally, the network reconstruction method we developed will be of use for many future studies, including dynamical modeling of trait-specific regulation in health and disease.

KURZFASSUNG

Genregulatorische Netzwerke (GRN) spielen eine wichtige Rolle bei der Kontrolle der Entwicklung und bei der stabilen Aufrechterhaltung von Merkmalen, z.B. Körperteilen, Organen, Geweben und Zelltypen. Kleine, wiederholt vorkommende Graphen, die Netzwerk-Motive genannt werden, sind in GRN im Vergleich zu randomisierten Netzwerken angereichert. Des Weiteren lassen neuere Studien darauf schließen, dass die Topologie von Netzwerken für die Regulation von Merkmalen von Bedeutung sein könnte.

Hier charakterisieren wir die globale und lokale Struktur von GRN von zwölf humanen Zelltypen und sechs Maus-Geweben. Wir vergleichen die Struktur dieser GRN und ihr Interaktionsrepertoire, und untersuchen ob Netzwerk-Motive zusammen mit merkmalsbestimmenden Regulatoren / Transkriptionsfaktoren (TF) vorkommen. Zu diesem Zweck haben wir eine neue Methode zur Rekonstruktion zelltypspezifischer und gewebsspezifischer GRN entwickelt, die auf wahrscheinlichkeitsbedingter Voraussage von TF Bindungsstellen in regulatorischen Regionen, Expressions-Screens (RNA-seq) und Daten basiert, die die Besetzung von Bindungsstellen durch DNA-bindende Proteine beschreiben (DNase-seq / DNase footprinting). Die Integration verschiedener Daten-Typen ermöglicht es uns, falsch-positive Voraussagen zu gering zu halten und das aktive Sub-Netzwerk eines bestimmten Merkmal zu identifizieren.

Unsere Ergebnisse zeigen, dass spezifische Netzwerk-Motive, darunter besonders der Feed-forward Loop, eine Funktion bei der Regulation von Merkmalen haben könnte. Außerdem zeigen wir, dass das Interaktionsrepertoire von GRN zur funktionellen Klassifikation von verwandten GRN herangezogen werden kann. Unsere Studie zeigt weitere Gebiete auf, die von methodischen Verfeinerungen und einheitlichen Standards profitieren würden, darunter TF Bindungsstellen-Modellierung und DNase footprinting Algorithmen. Die von uns entwickelte Methode wird außerdem für Folgestudien, beispielsweise die dynamische Modellierung der Merkmalsregulation in gesunden und in Krankheitsstadien, von Nutzen sein.

Contents

1	INTRODUCTION	1
1.1	Outline.....	1
1.2	Characters and traits.....	1
1.2.1	Definition	1
1.2.2	Trait evolution	1
1.2.3	A model of trait evolution: cell types.....	2
1.2.4	Studying homology to elucidate trait evolution	3
1.3	Gene regulation in eukaryotes.....	3
1.3.1	General principles	4
1.3.2	Transcription factors	4
1.3.3	Regulatory elements	4
1.3.4	Cell type- and tissue-specific regulation	5
1.3.4.1	Gene expression patterns	5
1.3.4.2	Other mechanisms of regulation	6
1.4	Gene regulatory networks (GRN).....	6
1.4.1	General.....	7
1.4.2	Construction of GRN	7
1.4.3	Reconstruction of trait-specific GRN.....	8
1.4.3.1	RNA-seq.....	8
1.4.3.2	DNaseI-seq	9
1.4.4	Properties of biological networks	9
1.4.4.1	Global topology.....	9
1.4.4.2	Local topology: Network motifs.....	10
1.5	Objectives of this work	10
2	MATERIALS AND METHODS	12
2.1	Materials	12
2.1.1	Genomes	12
2.1.2	Annotations.....	13
2.1.3	TFBS models	13
2.1.4	NGS data	13
2.1.4.1	RNA-seq data	14
2.1.4.2	DNaseI-seq data	15
2.1.5	Software applications	15
2.2	Methods.....	15
2.2.1	Generation of reference network.....	17
2.2.1.1	Selection of TF set	17
2.2.1.2	Extraction of regulatory sequences	18
2.2.1.3	Binding motif detection in regulatory sequences.....	19
2.2.1.4	Genomic location of binding motifs.....	19
2.2.1.5	Reference network.....	20
2.2.2	RNAseq: Expression calls.....	20
2.2.1.6	Specific issues with Cufflinks.....	20
2.2.3	DNaseI footprinting.....	20
2.2.1.7	Evaluation of DNaseI footprinting.....	21
2.2.4	Cell type- and tissue-specific regulatory networks	21
2.2.1.8	Overlapping reference network and DNaseI footprints	21
2.2.1.9	Assigning cell type-/tissue-specific interactions	21
2.2.5	Active subnet	24
2.2.6	Calculation of standard network measures	28
2.2.7	Detection of network motifs.....	28

2.2.8	Comparison of active subnets	29
2.2.8.1	Jaccard index of networks based on common edges	29
2.2.8.2	Normalized network degree clustering	29
2.2.8.3	Multidimensional scaling of networks	29
2.2.9	Functional enrichment analysis	29
2.2.9.1	Functional enrichment in active subnets	30
2.2.9.2	Functional enrichment in network motifs	30
3	RESULTS	31
3.1	Reference networks	31
3.1.1	Human reference network	31
3.1.2	Mouse reference network	33
3.2	Identification of expressed TF through RNA-seq	34
3.2.1	FPKM distribution	34
3.2.2	Housekeeping genes are expressed above expression cutoff	35
3.2.3	Total number of expressed genes is comparable across samples	37
3.2.4	Expression calling on transcription factors	37
3.3	DNaseI footprinting in DNaseI-seq data	38
3.3.1	Comparison of DNaseI footprinting results with published DNaseI footprints	38
3.3.2	Combination of DNaseI footprints with reference networks	39
3.3.3	DNaseI footprints in introns / intergenic regions and exons	41
3.4	Sample-specific networks	42
3.4.1	Distribution of weighted interaction types	42
3.4.2	Comparison of Hsmm cell type-specific network with muscle regulatory network	45
3.5	Active sub-networks	46
3.5.1	Characterization of active sub-networks using standard network measures	46
3.5.2	Distribution of weighted interaction types	47
3.5.3	Network motifs in active sub-networks	49
3.5.4	Comparison of active sub-networks	50
3.5.4.1	Jaccard index of active sub-networks	50
3.5.4.2	Normalized network degree clustering	51
3.5.4.3	Multidimensional scaling	52
3.5.5	Analysis of functional specificity	55
3.5.5.4	Functional enrichment in active sub-networks	56
3.5.5.5	Functional enrichment in significant network motifs	58
4	DISCUSSION	61
4.1	Active sub-networks	61
4.1.1	Establishing connection between network motifs and functional interpretation of regulatory networks	61
4.1.2	Establishing specificity and functional relatedness based on network interactions	63
4.2	Sample-specific networks	64
4.2.1	Distribution of interaction types implies advantages of species-specific TFBS models	64
4.2.2	Interaction types and weights pave the way for studies of network-wide dynamics	64
4.3	Methodic choices and considerations	65
4.3.1	Reference network	65
4.3.2	Expression calling on RNA-seq data	66
4.3.3	DNaseI footprinting	66
4.3.4	Sample-specific networks	67
4.3.5	Active sub-network	69
4.3.6	Network motif detection	69
4.3.7	Functional enrichment analysis	69

4.4	Scope of the study	70
4.5	The common structure of the dynamical core of traits	72
5	CONCLUSION & OUTLOOK	74
	REFERENCES	77

1 INTRODUCTION

1.1 Outline

In this work we investigate transcription regulatory networks that underlie traits. Our aim is to unravel topological design principles within such networks. Determination of the regulatory core network of traits is a central question of both developmental and evolutionary biology. Here we apply bioinformatic methods to address this question. To introduce the topic, we first outline existing knowledge on trait evolution and homology. We then focus on the regulation of gene expression as one of the main mechanisms that govern trait development and explain how it can be modeled in gene regulatory networks. Finally, we outline the role of our project in this context and introduce its detailed objectives.

1.2 Characters and traits

1.2.1 Definition

The terms character and trait are used to describe specific, observable, quasi-independent features of an organism. They can be described at different levels of organization, from body parts, organs and tissues, to cell types. Moreover, the word trait is also used to refer to behaviors, and to psychological properties of humans.

The textbook definitions of the words character and trait differ. While “character” refers in general to the abovementioned features, “trait” often refers to specific manifestations of a character (*i.e.* when the character is a human eye, blue and brown eyes would be traits), or to the measurements on characters.¹ Despite this difference, both terms are frequently used as synonyms. Both of them refer to phenotypic features that are subject to selection pressures. In this work, we use the words trait and character interchangeably, referring to features rather than their variants.

1.2.2 Trait evolution

The huge variety of organisms and their traits suggests questions about which forces drive this diversity, and in particular, what mechanisms underlie the origination of new traits.

When traits are examined more closely, it becomes apparent that they often serve specific functions which are reflected in their morphology and other properties. For example, red blood cells of mammals deliver oxygen throughout the body. Their typical bi-concave shape and the flexibility of their cell membrane optimize flow of the cell in blood vessels, enable efficient and fast diffusion of oxygen and allow them to bend within narrow capillaries. Generally, a trait’s presence in a living organism over long evolutionary periods suggests that it is non-detrimental and likely to carry a function. This function is relevant in a particular environment the organism lives in, therefore traits can acquire new functions, as well as become less useful, or even futile under changing environmental conditions.

Novel traits, also called evolutionary novelties, are characters without homology to other characters, neither within the same organism nor in an ancestor. The line between novelty and adaptation is thin however, therefore it is sometimes difficult to define homology rigorously, as evidenced by the common debates on which characters present novelties and which are adaptations.

According to one definition, novelties can arise by three different modes: divergence of a repeated element, *de novo* origination, or fusion of ancestral characters.²

In any of these cases, the establishment of a novel character requires permanent changes at the molecular level. The formation of traits is governed by developmental programs, *i.e.* by sets of regulators which control gene expression in a trait-specific spatio-temporal manner. This means that the development and maintenance of traits is associated with specific gene regulatory networks (GRN) which induce the expression of the appropriate genes and / or repress others. Consequently, the emergence of novelties can thus be driven by the modification of ancestral gene regulatory networks, by their *de novo* assembly, or by a combination thereof.³ An emerging trait needs to become somewhat independent of the underlying regulatory level in order to generate trait-specific variation and thus represent separately evolvable entity.

The importance of trait-specific GRN for trait evolution is also supported by sequencing studies which have shown that the body-patterning genes of complex animal species are surprisingly similar. While these findings highlight the historical continuity of genes in developmental regulation, they also imply that new traits may often arise by changing the existing gene regulation rather than by the invention of new genes and proteins.⁴

1.2.3 A model of trait evolution: cell types

The emergence of novel traits and the corresponding changes of regulation have been studied in a particular, simple model of trait evolution, namely in cell types. These studies propose that ancient cell types are characterized by multi-functionality and state that derived cell types have lost functions, or acquired new ones through the modification of ancestral functions, while becoming more specialized.⁵ These changes of functionality are caused by altered gene expression, where the confinement of multi-functionality is acquired by the inactivation of ancestral regulators, and more specialized functions by the activation of additional regulators. The evolution of new cell types is therefore characterized by the segregation and divergence of functions. Interestingly, an analogous change can also be seen in differentiation of cell lineages during development: Multipotent stem cells give rise to daughter cells with more specialized functions and less differentiation potential.

The aforementioned regulatory changes during cell type evolution suggest that the regulatory networks of related cell types should be similar. Similarity can be based on the activity of the same or orthologous regulators, or on the similarity of regulatory interactions. Within this work, we investigate whether and to what extent regulatory interactions among related cell types are shared.

1.2.4 Studying homology to elucidate trait evolution

In order to reconstruct trait evolution, it is necessary to compare different species and determine homologous traits. Returning to the example of cell types, some of the morphological and functional similarities between cell types may be obvious, but true evolutionary relationships are often obscure. Gene expression profiles can aid the classification of homologous traits. For example, the comparison of combinations of transcription factors involved in photoreceptor development has shed light on eye evolution.⁶ In this context it is preferable to compare regulatory networks rather than single genes to infer homology. This is because inference from single genes may be misleading. Co-option of genes can make analogous characters that evolved independently look like homologies when single genes are used to classify homology. Conversely, fast-evolving species may lose regulators that are well-conserved in candidate homologous traits in other species, making them appear non-homologous.⁵

The basic view of homology defines two characters as homologous when they arose through modifications of an original feature in a common ancestor. This definition implies some degree of morphological correspondence between the derived traits and their predecessors. In the case of evolutionary novelties however, this correspondence to ancestral traits does not exist. In this context, the term deep homology has been coined to describe “the sharing of the genetic regulatory apparatus that is used to build morphologically and phylogenetically disparate animal features”.⁷ This concept applies on the one hand to evolutionary novelties for which the wiring of regulatory genes precedes a visible phenotypic trait⁸, and on the other hand, to dissimilar traits that evolved by the recruitment and modification of the same gene regulatory circuits in a new context.

Deep homology once again highlights the importance of the underlying GRN for trait development. The significance of GRN also serves as a basis for the hypothesis of Character Identity Networks (ChINs) which states that the essential characteristics of homologous traits are regulated by a highly conserved, invariant gene network.⁹ Systematic investigations of ChINs are still missing and only few have been proposed up to this point, such as the regulatory network that governs segment polarity in insects. Candidate ChIN genes are trait-specific regulators whose importance has been demonstrated by knock-out or induction experiments, for instance using transcription factors which induce muscle cell differentiation.

1.3 Gene regulation in eukaryotes

As discussed, insights from trait evolution and development have pointed towards the regulation of gene expression as one of the main points of control for trait formation. The following sections are therefore devoted to introducing principles of eukaryotic gene regulation as well as two of its defining elements, namely transcription factors and regulatory regions. At its basis, our GRN modeling approach focuses on these two elements, however we also briefly present other mechanisms that impact gene expression and have been implicated in the cell type- and tissue- (*i.e.* trait-) specific regulation.

1.3.1 General principles

The expression of eukaryotic genes is regulated at multiple levels, such as transcription initiation, elongation, processing, transport, stability and translation of mRNA.¹⁰ Of these regulatory mechanisms, transcriptional regulation has received the most attention and is one that the best understood. Transcriptional regulation includes modes of control at the level of chromatin structure, such as histone modification and nucleosome remodeling, or at the level of protein-DNA interaction *i.e.* the binding of activators or repressors to regulatory elements in the DNA.

Eukaryotic genes are transcribed through the enzymatic action of RNA polymerase II. Prior to transcription, general transcription factors need to bind the core promoter which is located upstream of the gene and recruit RNA polymerase II to the DNA. The rate of transcription is influenced by transcription factors (TFs), proteins which bind regulatory elements in the DNA in a sequence-specific manner and thereby either hinder transcription (repressors) or enable/enhance it (activators). Unless activating TFs bind, transcription occurs only at a low basal level.¹¹

1.3.2 Transcription factors

Transcription factors are key players in transcriptional regulation. Their molecular structure is organized modularly into functional domains: DNA-binding domain, trans-activating domain and ligand-binding domain. DNA-binding domains interact physically with short DNA sequences (binding motifs) typically of 6-12 bp length. Even though TFs bind specific sequence motifs preferentially, certain nucleotides within these motifs can vary much, and binding sites are therefore also called degenerate. As the consensus binding motifs of TFs are short matching sequences are quite likely to arise by random mutation. While this may serve as an evolutionary mechanism for the generation of new potential regulatory protein-DNA interactions, it is unclear whether many of the matching sequences are bound by TF *in vivo* and whether this binding constitutes a functional regulatory interaction.¹⁴ For this reason, much effort has been invested into creating genome-wide maps of functional regulatory regions.

Besides binding DNA directly, TFs can dimerize or interact with other TFs, bind ligands, cofactors or general transcription factors. Usually these types of interactions occur via the ligand-binding or trans-activating domains. The joint binding of multiple TFs as well as their interactions with other proteins during transcriptional regulation are also referred to as combinatorial control of transcription. It is thought that combinatorial control effectively integrates a multitude of signals to fine tune expression and “meet the regulatory demand of a higher organism”, while it also makes it possible to create complex regulatory patterns for a high number of genes with only a small number of regulatory proteins.¹²

1.3.3 Regulatory elements

Regulatory elements, *i.e.* the DNA regions that are bound by TFs, also play a very important role in transcriptional regulation. One type, proximal regulatory regions can be found within proximity of transcription start sites (TSS) and core promoters. The other type of regulatory elements is located distally in regions that are non-adjacent to the gene they regulate. Such

regulatory elements / regions include enhancers, silencers, insulators and locus control regions.¹⁰ Enhancers can increase the rate of transcription of a promoter-containing gene, irrespective of their distance or orientation to it. Silencers have an inhibitory effect on transcription and otherwise share the properties of enhancers. They can be located both within proximal promoter regulatory regions and enhancers but also separately from them.¹⁰ Insulators work as boundary elements which limit transcription to specific regions of the DNA. Finally, locus control regions contain multiple of the aforementioned regulatory elements and coordinate the joint transcription control of entire loci and gene clusters.

The action of distally located regulatory elements is conferred through looping of the DNA and the interaction of DNA-bound proteins and / or RNA polymerase II which come into special proximity through the looping.¹³

The distal regulatory elements received less attention until recently, mainly due to the methodical difficulties (below)¹⁴. Recent studies revealed that distal regulatory regions contribute markedly to the combinatorial control of transcription.¹⁰ Studies of binding site occupancy have even put forward the idea that differential activity of distal enhancers might be involved in the development and differentiation of diverse cell types.¹³

Despite their importance for transcriptional regulation, study of distal regulatory elements poses various challenges. It is easy to estimate the location of proximal regulatory regions through the location of protein coding genes. In contrast, distal regulatory elements are harder to identify. Additionally, finding their target genes can also be problematic. High throughput chromosome conformation capture techniques contribute to elucidating enhancer interactions with other genomic locations,¹⁵ but genome-wide maps of enhancer-target gene relationships are still missing. In this study we thus focus on proximal regulatory regions to model GRN.

Regulatory elements are often conserved between not too distant species, such as rodents and humans. This means that nucleotide sequences in such regions can be highly similar, which is also exploited to identify regulatory elements in human DNA. However, it was estimated that a considerable proportion of functional binding sites in humans are not functional in rodents.¹⁶ Slight changes in ortholog binding sites of two *Drosophila* species were shown to affect the binding of TFs quantitatively, but little or no effect on gene expression patterns was observed.¹⁷ Still, mutations within regulatory regions and TF binding sites are believed to be the main drivers of phenotypic evolution.¹⁸

1.3.4 Cell type- and tissue-specific regulation

1.3.4.1 Gene expression patterns

As already outlined, the coordinated activity of TFs controls the expression of trait-specific genes during development and prompts cell type-specific differentiation or tissue-specific developmental programs. Not only development is guided by sets of TFs however, also established cell fates are maintained by the activity of TFs. Deregulations of TF activity can change gene expression and thereby lead to malfunctions and diseases such as cancer. To elucidate which regulatory mechanisms lead to disease, gene expression has been studied in great detail and typical patterns have been identified for diverse cancers. Furthermore, patterns of gene expression are used in the so-called molecular fingerprinting approach which identifies

homology and functional relatedness between cell types.⁵

1.3.4.2 Other mechanisms of regulation

Different other regulatory mechanisms are employed to generate trait- or condition-specific gene expression, such as those for cell types and tissues. While some of these mechanisms enable or repress transcription, others occur co-transcriptionally or post-transcriptionally.

An example of tissue- and cell type-specific regulation is the differential binding of TFs to alternative binding sites. This can often be explained by differential chromatin accessibility. Chromatin can have an open conformation, accessible to the transcription machinery and TFs or a closed conformation that inhibits transcription. Accessibility depends on ATP-dependent chromatin remodelers, histone modifications, or on methylation of the DNA, which is associated with transcriptional repression. Another factor determining differential TF binding is post-translational modification of the TF, or the cofactor interactions which change the TF's binding affinity.¹⁹ Such changes can guide a TF to bind preferentially bind one specific site instead of another, even though both have an open chromatin conformation.

What is more, eukaryotic mRNAs have splice variants that are produced through alternative splicing. Alternative splicing is a mode of pre-mRNA processing that occurs co-transcriptionally through the enzymatic action of proteins and small nuclear ribonucleic particles (snRNPs) that together form the spliceosome complex. During splicing, specific sections of a pre-mRNA (introns) are excised or left to remain within the mature mRNA (exons). Some of the exons can be skipped or included, some have alternative 3' or 5' ends (splice sites) and others are mutually exclusive. For certain introns it is also possible to remain within mature transcripts (intron retention). Alternative splicing offers a wealth of combinatorial options that is utilized to make tissue- and cell type-specific protein isoforms. The specific exon-makeup of a tissue-specific protein that was produced through alternative splicing can influence with which other proteins it interacts and therefore contributes to tissue-specific function.²⁰ Also the way in which TFs modulate transcriptional regulation can be changed by alternative splicing.²¹

Another mechanism of condition-specific regulation is the selective expression of miRNAs. miRNAs are non-coding, single-stranded RNAs of short length. They bind to specific target mRNAs which either hinders their target's translation or directs it towards degradation. Tissue-specific expression of miRNAs has been demonstrated along with significant reductions of their target's expression in the respective tissue.²²

1.4 Gene regulatory networks (GRN)

In the following, we introduce gene regulatory networks and explain the basic method how we reconstruct them. Then, we then present general principles of two next-generation sequencing - based methods (RNA-seq and DNaseI-seq) which are applied in our approach for trait-specific GRN reconstruction. Finally, we outline some general insights from the study of biological networks which also apply to GRN, such as the recurrence of network motifs.

1.4.1 General

Transcription regulatory networks are a subtype of biological networks and are often also referred to as gene regulatory networks. Using GRN, the complex process of transcriptional regulation can be condensed to a simple and comprehensive model. Molecular mechanisms of gene regulation become less important, instead the focus shifts to how signals are processed by the wiring of regulators and genes.

Typically, GRN are depicted as graphs in which the nodes / vertices represent regulators and genes, and the edges between nodes represent regulatory interactions. Interactions in GRN are usually directed, *i.e.* have a defined regulator and target and they can be further described with weights or interaction types (e.g. activating or inhibitory).

While the experimental establishment of regulatory interactions via methods such as yeast-one-hybrid assays or chromatin immunoprecipitation (ChIP) has long been arduous and time-consuming, more recent high throughput methods and computational prediction tools have greatly facilitated this process. Computational prediction cannot replace empirical evidence, but it can guide experimental interest towards potential interactions. Even though GRN can be constructed more readily now, new challenges arise in the form of spurious and weakly supported interactions which have to be managed to keep the error in the network as low as possible.

Furthermore, for many established interactions it is not known whether their regulatory effect is activating or inhibitory, yet this information is significant for quantitative modeling of information processing within GRN. Examining the interaction type experimentally involves a combination of gene expression measurement and protein-DNA binding assays. Besides their laboriousness, it is hard to establish the effect of combinatorial control with such methods, because various combinations of regulators have to be examined separately. In this work, we address this challenge by developing a new computational approach to infer interaction types based on the integration of experimental evidence from gene expression and binding motif occupancy.

GRN can be composed solely of regulators, such as TFs, or also include structural genes or other genes that are not involved in transcriptional regulation. Other transcriptional regulators, such as miRNAs, can also be included, but the identification of their targets relies on prediction. The composition and complexity of GRN generally depends on the subject of study and on data availability. In this work, we focus on TFs because this allows us to apply a rather straightforward model and keep prediction to a minimum.

1.4.2 Construction of GRN

As already mentioned, the development of genome-wide, NGS-based methods and computational tools has spurred GRN reconstruction. Moreover, various databases have been established which offer large collections of interactions, derived from systematic interaction assays.

Computational approaches apply mathematical models to infer the most likely GRN from expression data, rather than empirically establishing physical interaction. The approach we used

in this work is based on genome sequence data and the prediction of TF binding sites (TFBS) in regulatory regions, which is also referred to as probabilistic modeling.

Probabilistic modeling involves prediction of so-called binding motifs in genomic DNA. Binding motifs are based on the nucleotide composition of experimentally derived binding sequences. To obtain a binding motif, an alignment of these sequences is created first and a position frequency matrix (PFM) is constructed. It contains the frequency of each nucleotide at each position in the binding site, and serves as the basis for positional weight matrices (PWMs). PWMs reflect the likelihood for each nucleotide to arise at a certain position in the binding site. They are used for binding motif prediction by comparing the probability of the binding motif arising purely by chance based on the background distribution of nucleotides, versus the probability of it being an instance of the TFBS described by the probabilities in the PWM. The ratio of likelihoods is log-transformed and converted to a p-value and motif instances meeting a certain cutoff are used to construct the GRN. A motif with a significant p-value stands for an interaction between the regulator, which can bind the motif, and the target, in whose regulatory region the motif is located.

1.4.3 Reconstruction of trait-specific GRN

The presence of a predicted binding motif in a regulatory region means that the regulatory protein could bind to the given DNA sequence in principle. Consequently, a network generated by probabilistic modeling represents the sum of potentially possible interactions. In this work, we refer to such general networks as “reference networks”.

Probabilistic modeling likely has a high rate false-positive of prediction, because binding motifs are short and may arise randomly without having true regulatory functions. However, even if false-positives are controlled, only a part of the potential interactions from a general reference network may be realized in a given cell type or tissue. Therefore, new GRN reconstruction approaches are under development. Their focus lies on identifying those interactions that are present in specific cell types or tissues. For example, gene expression measurements (e.g. by microarrays or RNA-seq) have been applied to extract interactions that are active in specific cell types or tissues from a reference network.^{23,24} In this case, only interactions of those regulators that are expressed are included in the trait-specific GRN.

Furthermore, cell type-specific GRNs were reconstructed *de novo* from screens of genome-wide protein-DNA binding, obtained via DNaseI-seq data.²⁵ In this approach, interactions are assigned when a so-called DNaseI footprint overlaps predicted binding motifs.

As we combine both abovementioned methods and use data from RNA-seq and DNaseI-seq / DNaseI footprinting to identify the cell type- and tissue-specific interactions in reference networks, we explain the basic principles of these two NGS methods below

1.4.3.1 RNA-seq

The main idea behind RNA-seq is to measure the abundance of mRNA transcripts to infer expression in a quantitative manner. For this purpose, RNA is isolated from the biological sample under study, poly-A selected and transcribed into cDNA. From this point on, RNA-seq follows a

general next-generation sequencing protocol: cDNA is fragmented and adaptor sequences that match universal primers are ligated to the fragments, which can then be amplified by PCR and become templates for sequencing. Sequenced reads can be assembled to transcripts or aligned to reference genomes, if available. The number of reads per gene is then normalized to account for a) the number of reads in the experiment and b) transcript length (as longer transcripts will have more reads mapping to them by default). Finally, normalized measures of read abundance are used as a proxy to estimate the expression of mRNA.

One of the major challenges for the alignment of RNA-seq reads to eukaryotic genomes is splicing. Genome sequences contain introns, while reads do not, and splice junctions have to be inferred within them. Furthermore, reads that map to multiple transcripts with stretches of the same intron / exon structure may hinder the quantification of expression on the transcript-level. The past few years have brought about the development of specialized mapping tools for eukaryotic RNA-seq reads, e.g. Tophat²⁶ and STAR²⁷ which can infer splice junctions. The issue of reads that map to multiple transcripts may be resolved in the future with new sequencing techniques that produce longer reads that cover whole mRNA molecules (e.g. the Pacific Biosciences system²⁸).

1.4.3.2 DNaseI-seq

DNaseI footprinting as a technique to detect protein-DNA binding preceded the present use in the context of next-generation sequencing.²⁹ The novelty of DNaseI-seq consists of subjecting DNaseI-cleaved genomic sequences to high-throughput sequencing and footprinting on a genome-wide level. The mapping of sequenced reads or their 5' ends (DNaseI cleavage sites) to reference genomes makes it possible to identify so-called DNaseI hypersensitive sites (DHS). DHS are sites cleaved preferentially by DNaseI. They contain open chromatin which can also be bound by regulatory proteins and are therefore used as markers for functional regulatory regions. In addition, it was discovered that short without mapping DNaseI cleavage sites are located within DHS. They are enriched for binding motifs of regulatory regions and overlap binding sites of TF derived from ChIP-seq experiments.³⁰ For this reason, these short sequences are thought to be bound by DNA-binding proteins *in vivo* and have been termed "DNaseI footprints". Algorithms that detect DNaseI footprints within DNaseI-seq data are therefore used to identify binding sites occupied by regulatory proteins in specific conditions.

1.4.4 Properties of biological networks

1.4.4.1 Global topology

It has been shown that different networks found in nature share global topological properties. The various biological networks (e.g. protein-protein interaction, metabolic, signaling and transcriptional regulation networks) exhibit the so-called *small world property* and *scale-freeness*.³¹ Small world means that a node in the network can be reached from any other node in relatively few steps, and scale-freeness refers to the distribution of interactions among the nodes. Within a scale-free network, the degree distribution (*i.e.* the number of interactions per node) follows a power-law. In such networks, most of the nodes have a low degree, while a few have a very high number of connections and are therefore termed *hubs*.

Scale-freeness distinguishes natural networks from random networks, because the degree distribution of the latter is Poissonian and therefore less unequal.³¹ Generally, scale-freeness is interpreted as an evolved property that supports the robustness of biological networks. Because of the scarcity of hubs, even when many node perturbations occur, they are unlikely to hit the well-connected hubs. Much of the network structure is likely to be preserved upon few node deletions. Additionally, gene duplication contributes to the robustness of scale-freeness. As proteins / nodes with few interactions make up the majority, they are more likely duplicated than hubs. Thereby, the hubs to which they are connected receive even more interactions while other nodes remain weakly connected.³¹

A further hallmark of biological networks and GRN is modularity. This structural feature refers to grouping of nodes into modules, in which nodes are more interconnected within, than with nodes outside the module. The members of a module often participate in performing a common task or enabling a function.

Besides being modular, GRN have also been found hierarchically organized. Combining the hierarchical with a dynamical perspective, they can be divided into different parts: layers of nodes that only react to signals or amplify them without generating impulses themselves, and a “dynamical core” which defines the dynamics of the whole network.³² Based on the concept of dynamical cores, we identify the core active sub-network in cell type- and tissue-specific GRN.

1.4.4.2 Local topology: Network motifs

One further property that distinguishes natural networks like GRN from random networks is found in their local topology. Certain small sub-graphs are found significantly more frequently in natural networks than in random ones. These sub-graphs, also termed network motifs, have been observed in diverse types of networks: gene regulation, neuronal connectivity, food webs, electronic circuits and the world wide web.³³ The causes of their prevalence in such different networks are unclear. While it has been proposed that certain motifs may be suited to perform specific functions or are able to produce a range of functionally usable dynamics which may put them under positive selection, another explanation for their frequency may be an easy mode of generation through node duplication.

In this work, we investigate the presence of network motifs in the core active sub-networks of trait-specific GRN. We also examine whether they co-occur with TFs that are involved in trait development which would imply a role for topological patterns in the regulation of traits.

1.5 Objectives of this work

The finding of enriched network motifs has sparked interest in possible functional dynamics of these small sub-graphs. It has been demonstrated that different topologies of sub-graphs can produce the same dynamical outcome.³⁴ Regulatory dynamics corresponds to a phenotype. From the perspective of the phenotypic trait, this finding is interesting because it means that even at the level of regulatory interactions (not just synonymous substitution) a degree of

change can accumulate without the change in the phenotype. Such a hidden change could lay the groundwork for the subsequent development and individualization of a new trait.

Due to its important role in evolution, trait individualization has also been studied in a mathematical model.³⁵ This study has demonstrated that the development of a new trait requires a transition state in which both the ancestral and the derived trait are present. Even though this means that trait individualization involves a complex and evolutionary costly intermediate stage, this cost is biologically more likely than the emergence of a new trait in one step, without sharing any qualities of the ancestral state.

When these two results are regarded together, they suggest a role for regulatory network topologies in trait evolution. We therefore asked: “Is there molecular evidence for a role of specific topological or dynamical regulation patterns in trait evolution?”

To address this question, we survey regulatory core networks that underlie traits in 18 phenotypically distinct samples, 12 human cell types and 6 mouse tissues. We characterize global and local network structure, and investigate the occurrence of network motifs in these networks.

With regard to trait-specific regulation, we aim to find out whether regulators that drive trait development co-occur with specific topological patterns / network motifs.

Another question that we aim to clarify is whether GRN from related samples share similar structural properties, and whether there are similarities in their interaction repertoire. Given a common evolutionary origin and the current findings from cell type evolution it would be reasonable to find that related cell types share a higher number of interactions than unrelated ones. We want to find out whether the traits can be characterized by the interactions within the regulatory core.

With respect to methodology, we aim to develop an approach that will capture the core active sub-network and generate networks with edges that have interaction types. Inhibitory interactions in particular are essential in trait development, and an approach that can assign interaction types can be valuable for many future applications.

2 MATERIALS AND METHODS

In the following we will describe our workflow to generate and analyze cell type- and tissue-specific gene regulatory networks. Our principal idea was to first construct a reference network of all potential TF-TF interactions predicted from the sequence, irrespective of the tissue, cell type or differentiation status of the cell, and subsequently to use sample-specific experimental data to derive a sub-network that would reflect active regulatory interactions in a specific sample.

In more detail, our approach to generate cell type- and tissue-specific gene regulatory networks involved following steps:

- a) construction of reference network based on the prediction of binding motifs in regulatory sequences
- b) “mapping” of cell type- or tissue-specific gene expression and binding motif occupancy onto the reference network to derive a sample-specific regulatory network
- c) selection of interactions based on a set of defined criteria to find the core active sub-network (see below for criteria)

We then carried out different analyses on the generated active sub-networks to assess their specificity in regard to cell type or tissue function and to investigate their topological properties:

- a) functional profiling of TFs participating in the generated networks by gene ontology
- b) comparison of a generated network to a published cell type specific network
- c) structural characterization using global network measures
- d) detection of network motifs
- e) determination of positions of cell type / tissue marker TF relative to network motifs

We first provide detailed information about the different data types that have been used in this study and then move on to explain the different parts of our workflow in detail.

2.1 Materials

The construction of the reference network required transcription factor binding site (TFBS) models, genomic sequences and annotations. Inference of sample-specific networks was based on combining the reference network with next generation sequencing (NGS) data from the whole genome transcriptomes (RNA-seq) and regulatory site occupancy screens (DNaseI-seq), as schematically depicted in Figure 1.

2.1.1 Genomes

Genome sequences were downloaded in FASTA format from Ensembl³⁶. The human genome release 73 and the mouse genome release 74 were used. We selected these releases for their compatibility with the GENCODE³⁷ annotations used in this study (see list of Ensembl genome releases and GENCODE annotations: <http://www.encodegenes.org/releases/>).

2.1.2 Annotations

Annotations were obtained in gtf-format from GENCODE. We worked with GENCODE release 18 for human data and with GENCODE release M2 for mouse data to ensure consistency with the annotations used for the alignment/mapping of RNA-seq data.

2.1.3 TFBS models

After evaluating different TFBS database choices, we decided to use Homo Sapiens Comprehensive Model Collection (HOCOMOCO)³⁸ because it was open to the public and offered a comparatively large set of TFBS models. We obtained TFBS models formatted as probability matrices in MEME text format from the HOCOMOCO website on 13.05.2013 (available at: http://autosome.ru/HOCOMOCO/download_helper.php?path=download/HOCOMOCov9_AD_MEME.txt&name=HOCOMOCov9_AD_MEME.txt). For the mouse TF-TF interaction reference network, we used the mapping of HOCOMOCO TFBS models to mouse TFs provided by the authors of the database (available at: http://autosome.ru/HOCOMOCO/download_helper.php?path=download/HOCOMOCov9_motifs2UniProt.xlsx&name=HOCOMOCov9_motifs2UniProt.xlsx). In the following, human TFs will be referred to with names written in UPPERCASE (the convention generated by Hugo Gene Nomenclature Committee (HGNC), e.g. AHR) and mouse TFs will be written with the first character in uppercase (the convention generated by Mouse Genome Informatics (MGI), e.g. Ahr).

2.1.4 NGS data

RNA-seq expression data and DNaseI seq data used in this study were generated in the Human and Mouse ENCODE projects³⁹ (see Table 1 for overview of the samples). For both types of data, we worked with ready-made alignments whenever possible. A part of the RNA-seq alignments from the official ENCODE release were not compatible with the current version of RNA-seq analysis tool Cufflinks⁴⁰ (ver 2.1.1). For these samples, we obtained newly aligned RNA-seq data from the authors of original data (available at <ftp://ftp2.cshl.edu/gingeraslab/tracks/Alex/Important/ENCODE2/BAM/>).

	Sample name	Description	DNaseI-seq: data availability	RNAseq: No. of replicates
Human cell types	A549	Epithelial cell line derived from a lung carcinoma tissue. (PMID: 175022), "This line was initiated in 1972 by D.J. Giard, et al. through explant culture of lung carcinomatous tissue from a 58-year-old caucasian male." - ATCC, newly promoted to tier 2: not in 2011 analysis	*	2
	CD20+	B cells from donors RO01778 and RO01794	***	2
	Hepg2	Hepatocellular carcinoma	***	4
	Hsimm	Skeletal muscle myoblasts	***	4

	Huvec	Umbilical vein endothelial cells	*	4
	Imr90	Fetal lung fibroblasts	**	2
	K562	Leukemia, "The continuous cell line K-562 was established by Lozzio and Lozzio from the pleural effusion of a 53-year-old female with chronic myelogenous leukemia in terminal blast crises." – ATCC	***	4
	Lhcnm2	Skeletal myoblasts derived from satellite cells from the pectoralis major muscle of a 41 year old caucasian heart transplant donor, immortalized with lox-hygro-hTERT ("LH"), and Cdk4-neo ("CN"), Zhu et al. (2007) in Aging Cell, vol. 6, pp 515-523	*	1
	Lhcnm2 Myc 7d / 4d	Lhcnm2 cells harvested after myogenic differentiation, see http://genome.ucsc.edu/ENCODE/protocols/cell/human/LHCNM2_Wold_protocol.pdf	*	1
	Monocd14	Monocytes-CD14+ are CD14-positive cells from human leukapheresis production, from donor RO 01746 (draw 1 ID is RO 01746, draw 2 ID is RO 01826)	*	2
	Nhlf	Lung fibroblasts	***	4
	SK-N-SH-RA	Neuroblastoma cell line, treatment: differentiated with retinoic acid ⁱ (Biedler, et al. Morphology and Growth, Tumorigenicity, and Cytogenetics of Human Neuroblastoma Cells in Continuous Culture. Cancer Research 33, 2643-2652, November 1973.)	***	2
Mouse tissues	Heart	Heart, adult, 8 weeks	*	2
	Kidney	Heart, adult, 8 weeks	*	2
	Liver	Heart, adult, 8 weeks	*	2
	Lung	Heart, adult, 8 weeks	*	2
	Thymus	Heart, adult, 8 weeks	*	2
	Whole Brain	Whole brain, embryonic day 14	*	2

Table 1: Overview of human and mouse samples. *: aligned DNaseI seq reads, **: Neph DGF, ***: both aligned DNaseI seq reads & Neph DGF. For all DNaseI samples with aligned reads, only one replicate was available. For human, the sample descriptions are taken from ENCODE controlled vocabulary: <http://genome.ucsc.edu/cgi-bin/hgEncodeVocab?type=cellType>, for mouse, samples are described according to the sample names in: ftp://ftp2.cshl.edu/gingeraslab/tracks/Alex/Important/ENCODE2/BAM/Mouse_CSHL/

2.1.4.1 RNA-seq data

We worked with poly-A-selected paired-end data with 75 bp read length generated on the Illumina GAI or GAIx platform. The human reference for the alignment was hg19, and as an annotation GENCODE release 18 was used. The mouse reference for the alignment was mm10, the annotation was GENCODE release M2.

ⁱ treatment with retinoic acid induces neuronal-like differentiation in SK-N-SH

2.1.4.2 DNase-seq data

DNase-seq data were obtained as alignments to hg19 for human samples and alignments to mm9 for mouse samples. Additional data with ready-made DNase footprints identified using the digital genomic footprinting (DGF) algorithm were obtained from published data (available at: ftp://ftp.ebi.ac.uk/pub/databases/ensembl/encode/integration_data_jan2011/byDataType/footprints/jan2011/)

2.1.5 Software applications

Our workflow made use of a number of published programs and applications which we will refer to in detail in the Methods.

2.2 Methods

Before describing each step of our approach in detail, we give an outline of the whole workflow in this section. As already mentioned, we started with the construction of a reference network. After selecting a set of TF from the database that was consistent with the annotations used in this study (*i.e.* excluding TF that were not present in the annotations), we derived regulatory regions for each TF from the genome. We defined regulatory regions as intronic or intergenic sequences inside a ± 5 kb interval around the transcription start sites (TSS; *i.e.*, mRNA start) of a TF of interest. In these regions, we scanned for putative TF binding site locations. For each TF that had a binding site in another TF's regulatory region, we made an edge in the reference net. Each such edge signifies a potential directed regulatory interaction with the first TF acting as a regulator and the second TF as a target (Figure 1).

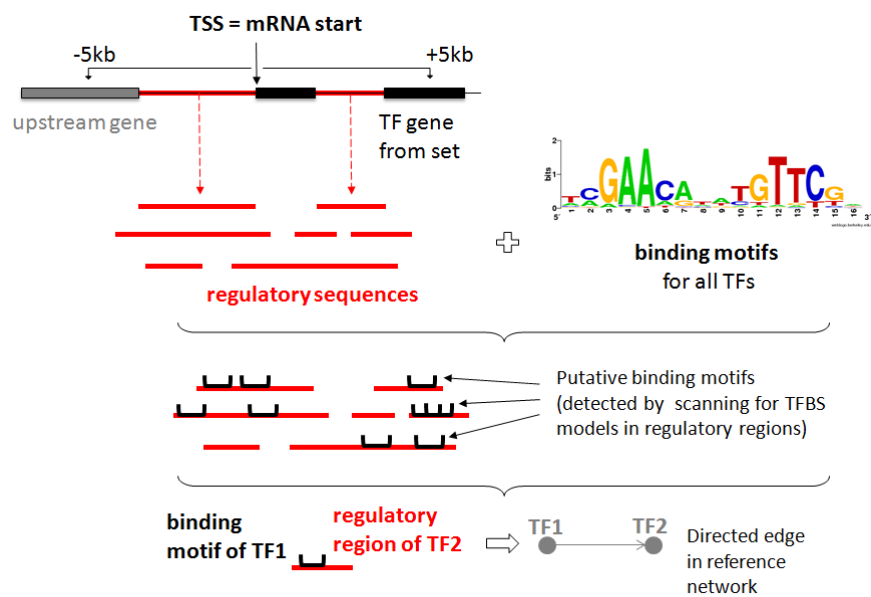


Figure 1: Construction of reference network. For each TF, regulatory regions are inferred on the basis of TSS location and intron/exon structure. TFBS locations are then detected in regulatory regions and based on this, the reference network is constructed. A directed edge in the reference network represents the presence of a TFBS of the regulator-TF in one of the regulatory regions of the target TF.

After creating the reference network, we turned to cell type- or-tissue- (henceforth sample-) specific expression (RNA-seq) and TF binding site occupancy (DNaseI-seq) profiles. We estimated expression for each of our TF's protein coding isoforms, and selected those isoforms that met a specific expression cutoff. Each TF with at least one protein coding isoform expressed above the cutoff was regarded as expressed.

For the DNaseI-seq data, we first detected DNaseI footprints and then intersected footprint locations with locations of TF binding sites from the reference network. Thereby we obtained TF binding sites that were occupied by their cognate TF in each specific sample.

Using gene expression and the occupied TF binding sites, we derived sample-specific interactions from the reference network and assigned interaction types (activation, inhibition) in order to generate a sample-specific regulatory network.

We selected the dynamically relevant interactions in each sample-specific regulatory net to obtain an active regulatory sub-network. For this purpose, we developed a set of selection criteria. Stages of the workflow from reference network to active sub-network are depicted in Figure 2.

As a last step, the active regulatory sub-networks for each cell type and tissue under study were characterized with respect to network topology, and analyzed with respect to its importance for the cell type or tissue function.

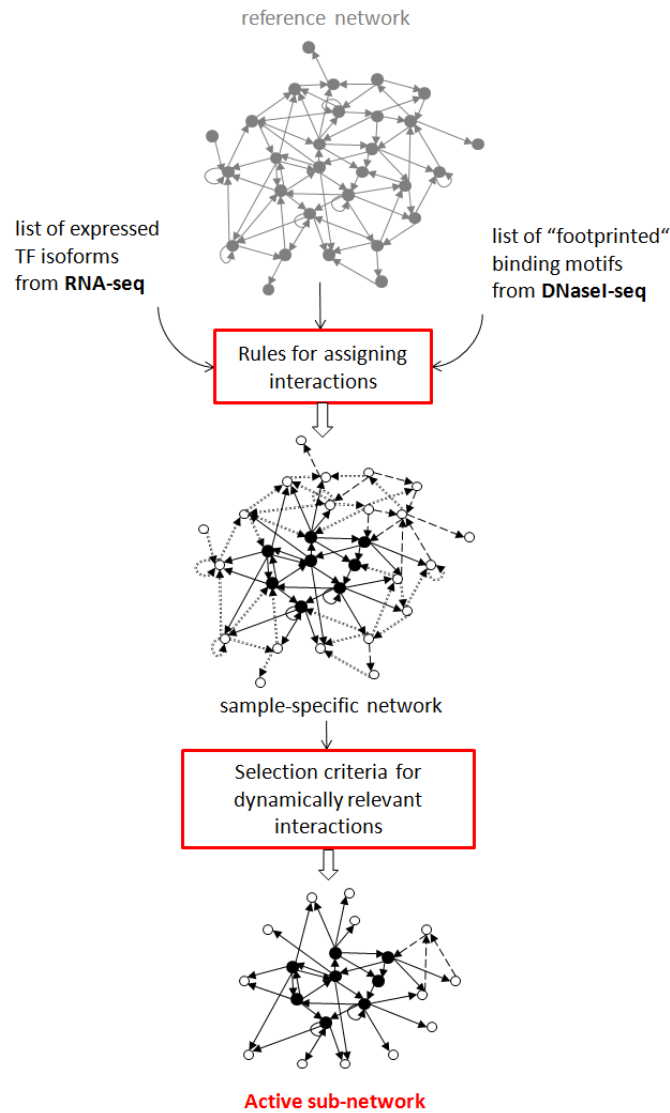


Figure 2: Steps in the workflow from the reference network to the active sub-network. The reference network is made up of predicted interactions (shown in grey). According to the evidence from RNA-seq and DNase-seq, distinct rules are applied to assign interaction types and weights to the predicted edges (weights shown with different line types). Interactions for the active sub-network are chosen using specific selection criteria that filter out dynamically irrelevant interactions.

2.2.1 Generation of reference network

2.2.1.1 Selection of TF set

Of the 426 TFBS models in the HOCOMOCO database, four models are assigned to more than one transcription factor, and 25 transcription factors are assigned two TFBS-models. We regarded TFs with shared TFBS-models separately to capture interactions in a 1:1 manner. One TFBS model, “ZBTB4!METH”, was excluded from the analysis as our study did not take into account DNA methylation states (ZBTB4 had two different TFBS models that captured its affinity to unmethylated and methylated binding sites). Another TF, NR2E3 was excluded for lacking protein coding isoforms. As a result, we worked with 404 TFs with one or more TFBS-models to generate the human reference network. For generating the mouse reference network, we used a mapping of 414 of HOCOMOCO TFBS-models to 393 mouse TFs, provided by the authors of the database (see chapter 2.1.3).

One of these TF, Hoxd4, was excluded because it was not present in the mouse genome

annotation applied in this study. Later, we also had to exclude Foxc2, Foxf1, Irf8 and Sox2 after analyzing RNA-seq data because their expression estimates were either not usable (marked “LOWDATA” in the Cufflinks output, see chapter 2.2.2), or no expression values were provided in the Cufflinks output. After these exclusions, we worked with 388 mouse TF with one or more TFBS models to generate the mouse reference network.

The TF names were provided in HOCOMOCO as UniProt IDs. To parse genome annotations we converted TF names to HGNC symbols (human) or MGI symbols (mouse) using Ensembl Biomart⁴¹, UniProt KB⁴², UniProt ID mapping (available at <http://www.UniProt.org/mapping/>) and MGI (available at <http://www.informatics.jax.org/batch>).

2.2.1.2 Extraction of regulatory sequences

We extracted FASTA sequences of regulatory regions from a whole-genome FASTA file using a custom-made Perl/BioPerl⁴³ script (*extract_reg_regions.pl*). Our definition of regulatory regions included intronic or intergenic regions inside a +/-5kb cutoff around the TSS (mRNA start) of a TF of interest.

The input of the script is an annotation in gtf-format. In this format, genomic features are described in a one-line-per-feature manner. Genomic features, for example exons or coding sequences, are described with chromosome number, start and stop coordinates and different tags, including name, gene or transcript type and various identifiers (Ensembl gene/transcript, HAVANA gene/transcript etc).

Before running the extraction script, the annotation was parsed for lines that described genomic features of the target TF. This parsed annotation was then supplied to the extraction script. Parsing the annotation shortened the runtime of the extraction script, because it made it unnecessary to parse the whole annotation file repeatedly while running the script.

The script’s output is a multi-FASTA file that lists all extracted regulatory regions. The identifier of each region includes the name of the transcript from which it was derived and its genomic location.

For the structure of *extract_reg_regions.pl* see Box 1; the full code of the script can be found in supplemental materials.

```
TOP section
    open FASTA genome, parsed annotation, original annotation
    specify cutoff for regulatory regions
    create out-, logfile names and filehandles
    associate each chromosome with its DNA sequence:
        e.g. chr1 -> "ACGGC..."
    create TRANSCRIPT SET (protein-coding transcripts of all TF)
        with transcript name, coordinates, strand, descriptive info

MAIN section
    for each transcript in TRANSCRIPT SET
        get start & end coordinates of each exon
        sort start coordinates ascendingly
        sort end coordinates ascendingly
        if (transcript is on +strand)
            TSS = first start coordinate
```

```

        upper boundary, lower boundary = TSS +/- cutoff
        remember intron coordinates inside boundaries based on the exon
        coordinates
    if (transcript is on -strand)
        TSS = last end coordinate
        upper boundary, lower boundary = TSS +/- cutoff
    remember intron coord. inside boundaries based on exon coord.
    SUB: print to log: TSS, boundaries, intron coordinates, transcript
    name and type
    SUB: get coordinates of intronic/intergenic regions inside boundaries
    SUB: get sequences with coordinates of intronic/intergenic regions
    SUB: print FASTA outfile: identifier of transcript + regulatory
        region sequences

SUBROUTINES
    get coordinates of intergenic regions inside boundaries {
        SUB: get exon coordinates
        SUB: make "exonic stretches"
        SUB: get intron/intergenic coordinates
    }
    get exon coordinates {
        get coordinates of exons of other genes that fall inside boundaries
        (from annotation)
    }
    make exonic stretches {
        get coordinates of longest possible exonic stretches by combining exon
        coordinates and selecting those that make the longest stretch
    }
    get intron/intergenic coordinates {
        get coordinates of seq that fall inside boundaries but not inside
        exonic stretches (=intronic/intergenic seq)
    }
    get sequences based on coordinates {
        get chr sequence from %Hash "chromosome No." -> "ACCTGATGAAGGC..."
        get regulatory regions as subsequences of chr sequence
        (get complementary sequence if (-strand))
    }
    print FASTA outfile {
        print ">ID;genomic location\nnewline"
        chunk sequences into 80 bp pieces
        print chunks
    }
}

```

Box 1: Pseudocode of extract_reg_regions.pl. Genomic coordinates of all TFs' protein coding transcripts and exons. Based on the TSS (mRNA start) of each such transcript, a +/- 5kb interval around the TSS is defined. Introns and intergenic regions inside these boundaries are output as regulatory regions.

2.2.1.3 Binding motif detection in regulatory sequences

On the extracted regulatory regions in FASTA format we detected putative binding motifs using FIMO⁴⁴ (Find Individual Motif Occurrences). To account for the distinct nucleotide composition present in the proximity of TSS, we adjusted background frequencies of nucleotides in human (A: 0.247, C: 0.251, G: 0.254, T: 0.248) and mouse (A: 0.255, C: 0.243, G: 0.246, T: 0.256) according to published data⁴⁵. FIMO was run with default parameters and a p-value threshold of e^{-5} .

2.2.1.4 Genomic location of binding motifs

In the FIMO output, binding motif locations were recorded relative to the start of the regulatory sequence. From these coordinates we calculated genomic locations using the coordinates of the regulatory regions in the corresponding FASTA header.

2.2.1.5 Reference network

The presence of a TF's binding motif in the regulatory sequence of a second TF was registered in the reference network as an edge. Each edge represents a potential directed regulatory interaction between the first TF as a regulator and the second TF as its target.

2.2.2 RNAseq: Expression calls

Transcript abundance was estimated from aligned RNAseq data using Cufflinks⁴⁰ (2.1.1). In the output of the program, expression estimates were reported in FPKM (Fragments Per Kilobase of transcript per Million mapped reads, a measure for paired end sequencing set-ups similar to the widely used RPKM (Reads Per Kilobase of transcript per Million mapped reads)).

We first sought to define actively expressed genes through an expression cutoff based on the FPKM distribution. Provided a deep enough sequencing, most annotated genes can be observed expressed at some level.⁴⁶ In such a case, the FPKM distribution would have a single peak at some point right of the value 0 FPKM and an appropriate cutoff for actively expressed genes could be chosen based on the location of the peak. In our samples however, the peak is at 0 FPKM (chapter 3.2.1, Figure 7), suggesting the samples are not sequenced deeply enough.

As the FPKM distributions in various samples did not point us to where to set an appropriate cutoff for expression, we resorted to the published expression cutoff⁴⁷ of FPKM > 1. We applied this cutoff to expression estimates of protein coding isoforms of our target TF. We considered a TF gene expressed if any of its protein coding isoform would meet the cutoff. Regarding replicates, we considered an isoform/transcript expressed either by decision of majority or, if this was impossible, by calculating the mean FPKM over isoforms and deciding upon cutoff.

2.2.1.6 Specific issues with Cufflinks

In some cases, Cufflinks did not report expression values for certain isoforms that were present in the annotation. After manual inspection, we concluded that this happened when a) two isoforms had the same intron/exon structure or b) a gene had only one exon and thereby only one isoform with the same start/end coordinates as the gene itself. A likely explanation for this was that the program resolved this ambiguity by assigning reads that aligned to such regions to only one of the two possibilities. Expression was then only estimated for one of the two isoforms by Cufflinks, or either the gene or the isoform. As a result of this, certain isoforms were missing from the output, leaving in some cases no isoforms to make the decision whether a TF was expressed or not. In these cases we decided to evaluate the gene's expression value to assess a TF's expression.

Furthermore, we filtered out transcripts that were marked with the tag "LOWDATA". In case of TFs where all isoforms were marked in this way, we used the expression estimates of the corresponding genes if their status was "OK".

2.2.3 DNaseI footprinting

DNaseI seq data were available to us either in the form of aligned reads from a DNaseI seq experiment or as already processed, ready-to-use footprints that had been detected by the

digital genomic footprinting algorithm (DGF).³⁰ From aligned DNaseI seq reads we detected footprints of 6-40bp width with a p-value cutoff of e^{-3} using the R package *DNaseR* (1.0.0)⁴⁸. To correct for multiple testing, the Benjamini-Hochberg procedure was used, it is set as the default in *DNaseR*.

2.2.1.7 Evaluation of DNaseI footprinting

We compared DNaseI footprints derived by the *DNaseR* algorithm with footprints identified using the *DGF* algorithm. For the comparison, the same DNaseI-seq data sets were used. As the algorithms use different criteria to define footprints, we expected shifts in footprint edges and searched for footprints that overlapped to different extents (1, 5, 10 bp, exact overlap). We first sorted the footprints, which were in bed-format, with BEDOPS⁴⁹ *sort-bed* and then overlapped using BEDOPS *bedmap*. For *bedmap*, we varied the *--ovr* (overlap) parameter to 1, 5 or 10 or replaced it by *--exact*.

2.2.4 Cell type- and tissue-specific regulatory networks

After evaluating gene expression and detecting DNaseI footprints, we carried on with generating a sample-specific regulatory network and assigning interaction types.

First, we overlapped DNaseI footprint locations with the putative binding motive locations that had been obtained by scanning regulatory regions with FIMO (chapter 2.2.1.3).

Subsequently, we derived the sample-specific regulatory network by “mapping” evidence from gene expression and/or DNaseI footprints onto the reference network and assigning interaction types. Inferring the sample-specific network and assigning interaction types was carried out according to a set of defined rules which will be explained in detail below.

2.2.1.8 Overlapping reference network and DNaseI footprints

The reference file with the genomic locations of binding motifs and the files containing DNaseI footprints were sorted using BEDOPS *sort-bed* and then overlapped using BEDOPS *bedmap*. All putative binding motifs overlapping a DNaseI footprint by at least 3 bp were considered “footprinted”, i.e. bound by their cognate TF.

2.2.1.9 Assigning cell type-/tissue-specific interactions

We now elaborate the rules that we developed to derive sample-specific networks. We assigned different interaction types to the edges in the reference network based on the evidence from RNA-seq and DNaseI-seq experiments. Depending on the strength and consistency of evidence that supported an interaction, we also assigned weights. We will first describe rules for interactions with an expressed TF as a regulator, and then move on to explaining the rules for interactions that have unexpressed TFs as regulators.

a) Interactions with expressed regulators

If the regulating TF of an interaction was expressed, we searched for a “footprinted” binding motif of the regulator-TF in the target TF’s regulatory region. If there was a DNaseI footprint on a binding motif, this meant that the binding motif had been occupied by a DNA-binding protein in situ, likely by its cognate TF. We therefore assigned an interaction with the TF whose binding motif was “footprinted” as the regulator and the TF in whose regulatory region the binding motif

as the target. If there was no footprint on the regulator TF's binding motif, we did not assign an interaction.

We regarded the interaction as positive/activating if the target was expressed and as negative/inhibiting if the target was not expressed. Since these two interaction types, the negative and the positive, are independently supported by the reference network and both sources of experimental evidence (expression and DNaseI footprints), we assigned the highest weight "A" for these interactions. In the following, the two interaction types will be referred to as "A+" or "A-", for activation or inhibition respectively.

A schematic depiction of rules involving expressed regulators can be found in Figure 3 A.

b) Interactions with unexpressed regulators

For edges in the reference network where the regulator was not expressed, we examined all binding motifs in the regulatory regions of the target TF for DNaseI footprints. If none of the binding motifs was "footprinted", this meant that none of the other potential regulators from the reference network had a regulatory influence on the target. In this case, the absence of the regulator could have a regulatory impact on the target.

Depending on the expression of the target we again assigned positive/activating or negative/inhibiting interaction types. If the target was expressed, its expression could have been influenced by the absence of the suppressing regulator, implying that if the regulator had been expressed, the target would not be expressed. The interaction assigned in this case was therefore negative/inhibiting. Likewise, if the target was not expressed, its non-expression could have been influenced by the absence of the regulator if none of the binding motifs in the target's regulatory region was "footprinted". In this case, we inferred that if the regulator had been expressed, the target would have also been expressed. We therefore assigned a positive/activating interaction.

These two types of interaction are supported by less certain evidence than the interactions involving expressed regulators that were described above. They are based only on the predictions of the reference network and the absence of DNaseI footprints on binding motifs in regulatory regions. We therefore assigned a lower weight: "B". These two interaction types will be called "B+" or "B-" in the following, depending on activation or inhibition.

When we examined reference network edges with unexpressed regulators and found any of the binding motifs in the target's regulatory regions "footprinted", the influence of the absent regulator on the target was even less certain. It was unclear whether the unexpressed regulator had an activating, inhibiting or even no regulatory influence at all on the target. In such cases, the interaction was solely supported by the predicted edge in the reference network and we therefore assigned the lowest weight "P" (for "predicted").

For a graphical illustration of the rules for interactions going out from unexpressed regulators, please refer to Figure 3 B and Figure 3 C.

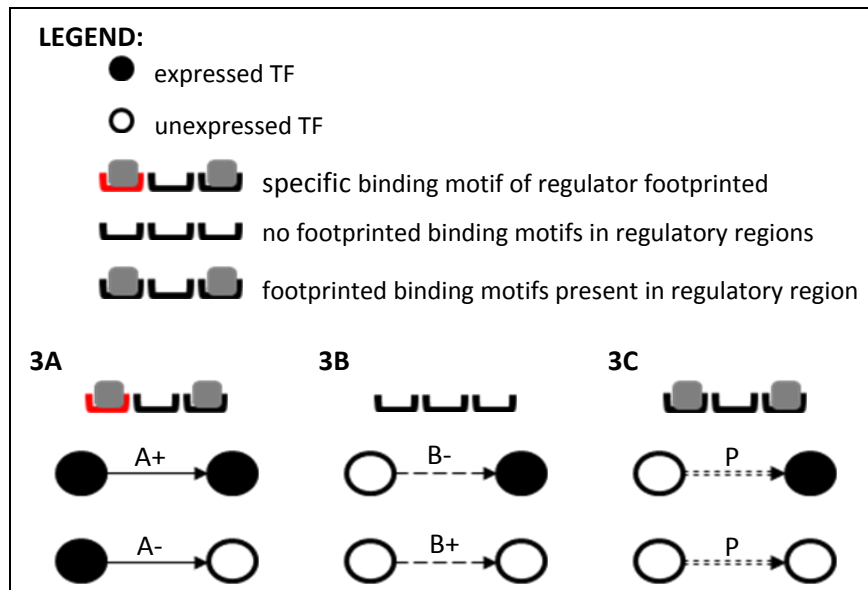


Figure 3: Rules for assigning interactions in the sample-specific network and weights for each category. 3A: Interactions of the highest weight A are assigned when regulators are expressed and their specific TFBS/binding motif in a regulatory region of the targets is footprinted. 3B: Interactions of intermediate weight B are assigned when regulators are unexpressed and there are no footprints present on the TFBS/binding motifs in the regulatory regions of the target. 3C: Interactions of the lowest weight P are assigned when regulators are unexpressed and unspecific TFBS/binding motifs in the regulatory regions of the target are footprinted. The highest weight is A, B is an intermediate weight and P ("predicted") is the lowest weight.

We implemented the rules described above in a perl script (*assign_cell type-specific_interactions.pl*) that used as inputs a list of expressed TF isoforms, the reference network and the overlap of the reference network with DNaseI footprints. The output of the script included an edgelist (list of interactions), all incoming interactions listed for each TF (node outfile) and a file listing expression on the gene-level. For structure of *assign_cell type-specific_interactions.pl* see Box 2; the full code of the script can be found in the supplement.

```
TOP section
open mapping file of TF-isoform/-gene name to Ensembl identifier
open mapping of HGNC symbols to UniProt IDs
open file with genomic locations of binding motifs (reference net)
create SET 1: interactions from reference net
open file listing expressed isoforms
create SET2a: expressed TF
create SET2b: unexpressed TF (complete list - expressed TF)
open overlap of reference network with DNaseI footprints
create: SET3: TF with footprints in regulatory regions (i.e.
    "footprinted TF")
create SET4: "footprinted" binding motifs/IA
MAIN section
foreach interaction in SET1
    if (regulator ∈ SET2a)
        if (target ∈ SET2a)
            if (interaction ∈ SET4)
                interaction = A+
            if (target ∈ SET2b)
                if (interaction ∈ SET4)
                    interaction = A-
        elsif (regulator ∈ SET2b)
            if (target ∈ SET2a)
                if (target ∉ SET3)
```

```

                                interaction = B-
                        else
                                interaction = P
                elseif (target ∈ SET2b)
                        if (target ∉ SET3)
                                interaction = B+
                        else
                                interaction = P

OUTPUT section
        make edgelist outfile
        make node outfile
        make expression outfile

```

Box 2: Pseudocode of `assign_cell type-specific_interactions.pl`. Each interaction from the reference net (SET1) is examined. Specifically, the expression of an interaction's target and regulator are checked, and the presence of DNaseI footprints on binding motifs in the regulatory regions of the target is surveyed. For expressed regulators (SET2a), it is checked whether their specific binding motif inside the target's regulatory region is footprinted (SET4). Based on the expression of the target (SET2a or SET2b), an activating or inhibiting interaction of the highest weight, A, is assigned. For unexpressed regulators (SET2b), the script checks if the target contains any DNaseI footprint in its regulatory regions (SET3). If there is none, activating or inhibiting interactions of intermediate weight B are assigned according to the target's expression (SET2a or SET2b). When there are footprints present, interactions are marked as "predicted" with the lowest weight P, regardless of the target's expression state.

2.2.5 Active subnet

In the previous step, we assigned interaction types and generated a sample-specific network. To obtain the active sub-network inside this network, we devised a set of selection criteria. The rationale behind these criteria is that the active regulatory sub-network is governed by the expressed TFs. We therefore aimed to find criteria that would include all interactions and chains of interactions that were regulated by or that had a regulatory effect on the expressed TFs. In the following, we will present the criteria we developed along with their justification. The criteria were:

a) Exclusion of predicted edges of weight "P"

We first decided that active subnet edges should be supported by experimental data instead of being purely predicted, since both the type of predicted interactions and their validity were uncertain. This would reduce the error of the derived active sub-network.

b) Interaction is regulated by expressed TF

Given that the active regulatory sub-network is controlled by expressed TFs, including all interactions where the regulator was expressed was a self-evident criterion. These interactions, described in chapter 2.2.1.9 above, were supported by gene expression data and DNaseI footprints on the regulator-TF's binding motifs in the target-TF's regulatory regions. Depending on the expression state of the target, these interactions can be either positive/activating or negative/inhibiting, and their weight is "A" (*i.e.* this criterion includes all A+/- edges).

c) Interaction is regulated by unexpressed TFs and targeting expressed TFs

The above criterion (b) also includes regulatory interactions going out from expressed TFs targeting expressed TFs. To capture entirely all regulatory effects on expressed TFs, we included the interactions going out from unexpressed regulator-TFs to expressed target-TFs as targets (*i.e.* all B- edges).

The next criteria are concerned with including regulatory interactions exercised indirectly by or on expressed TFs.

d) Interactions between two “first neighbours”

In criteria (b) and (c), we included interactions in which unexpressed TF participated, either as targets or as regulators. These unexpressed TF were special in that they were direct neighbours of expressed TF in the network. We termed them “first neighbours” for this reason, and included interactions where both regulator and target were from the set of first neighbours in the active sub-network. Interactions between first neighbours either forwarded indirect regulatory interactions from or to expressed TFs or they could be part of a dynamically interesting network motif that included an expressed TF. (*i.e.* this criterion included B+ edges between selected (first neighbour) TF).

e) Interaction is part of a dynamically relevant interaction chain

Next, we tried to find dynamically relevant interactions that were not covered by the criteria we defined so far. All interactions left at this point were B+ interactions, *i.e.* positive / activating interactions between two unexpressed TFs. Since we were looking for interactions that were indirectly regulating or regulated by expressed TFs, we examined chains of interactions starting at the first neighbours of expressed TF. We found that there were three possible scenarios for these chains:

- i) the chain of interactions would start or end with a dead end TF (*i.e.* no other edges going out or coming in into this last TF)
- ii) the chain of interactions would run into a futile, self-contained loop or start inside such a loop
- iii) the chain of interactions that started with a first neighbour would end on a first neighbour at some point

From a dynamic point of view, only the last scenario was interesting. We therefore included only those edges that were part of interaction chains as described in iii).

For a better understanding, we illustrate our approach to finding the active sub-network in Figure 4.

LEGEND:

- expressed TF
- unexpressed TF
- first neighbour TF
(unexpressed direct neighbour
of expressed TF)

- IA of weight A
- - - - -> IA of weight B
-> IA of weight P

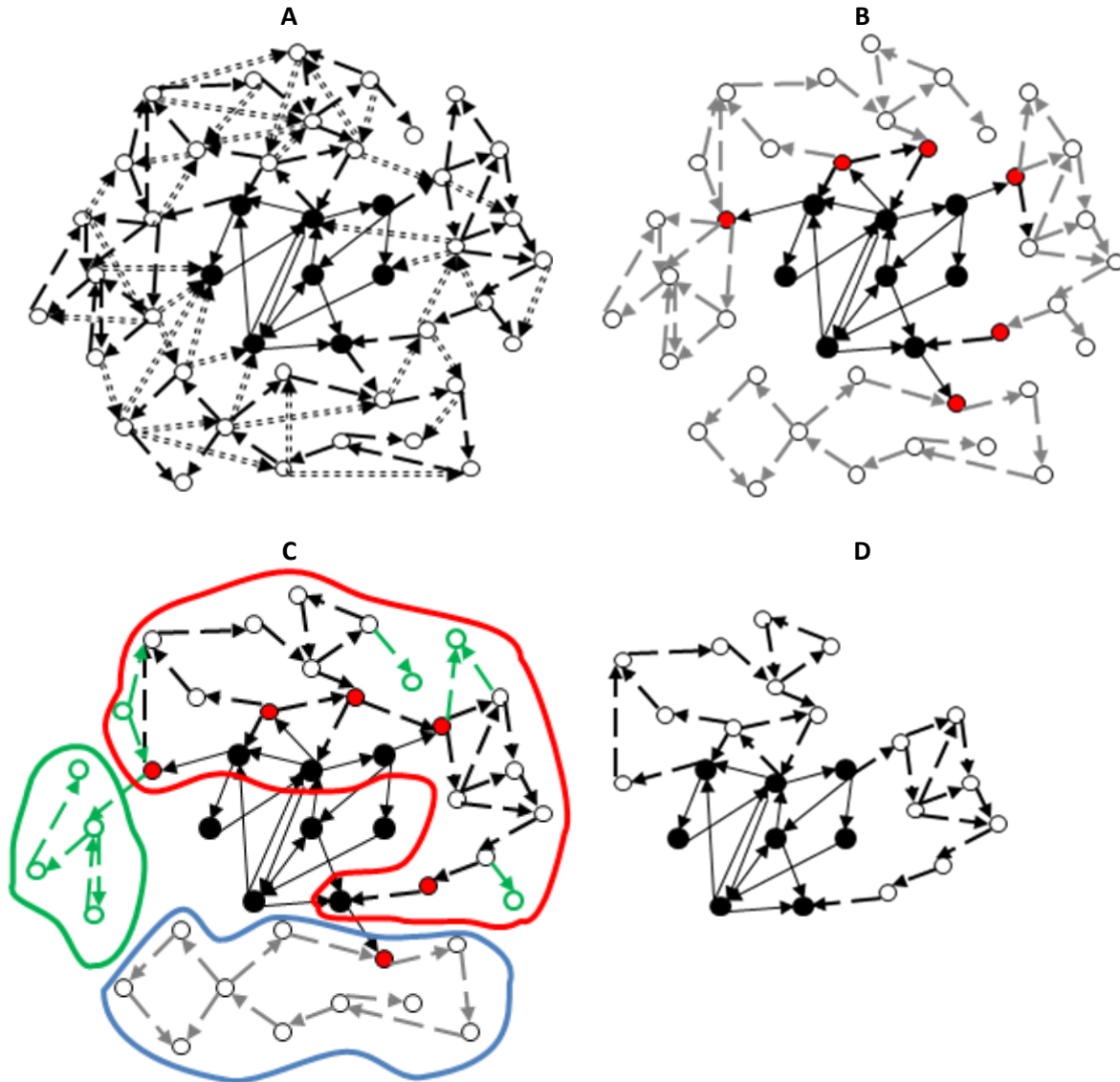


Figure 4: Deriving the active subnet from the sample-specific network. A: A Sample-specific network with expressed TF at the center and unexpressed TF around it, TFs are connected by different interaction types. B: Predicted edges (IA of weight P) have been removed from the sample-specific network. Applying criteria a-d as specified above, we included edges between expressed TF, from between expressed TF and first neighbours (marked red), and edges between first neighbours (coloured in black). A large part of the set of unexpressed TFs has not been explored yet and is coloured grey. C: By building interaction chains starting out from the first neighbours, we find dynamically relevant chains (circled red), futile loops (circled blue) and dead ends (circled/coloured in green) in the set of unexpressed TF. D: The active sub-network contains only interactions that fulfill one of the selection criteria; futile loops, dead ends and their vertices are excluded.

We developed an algorithm based on the criteria described here and implemented it in the script *active_subnet.pl*. The script first creates a set of “core” interactions that contains interactions between expressed TF, interactions between expressed TF and first neighbours, and

interactions between first neighbours. It then iterates through first neighbours and tries to find paths that go over unexpressed TF and lead back to the first neighbours again.

The script takes as an input the edgelist and expression outfile of *assign_cell type-specific_interactions.pl*, and as an output, it generates two files in csv-format, one that list all nodes and one that lists all edges of the active subnet. Furthermore, it generates a pajek net file and a pajek-like file. Pajek is a widely-used file-format that contains two parts. The first part is an indexed list of all vertices/nodes of the network (in our case vertices represent TFs and edges interactions). The second part lists all edges of the network and each edge is denoted with the indexes of the two vertices that are connected by the edge. The pajek-like file generated by the script lists not only TFs that are present in the active subnet, but the complete set of TF from this study as vertices. This file is later used for normalized network degree clustering (see chapter 2.2.8.2). For the structure of *active_subnet.pl*, please refer to Box 3.

```

TOP section
    read edgelist, expression list
    make SET1: expressed TF from expression list
    make SET2: interactions (IA) from edgelist

MAIN section
    get IA between expressed TF
    get IA from expressed TF to unexpressed and vice-versa
        remember IA as "core IA"
        remember unexpressed TF/targets of A- as "outgoing first neighbours"
        remember unexpressed TF/regulators of B- as "ingoing first neighbours"
    get IA between "first neighbours", remember them in the "core IA"
    prepare potential chain-IA (weight B+, IA  $\neq$  IA between first neighbours)
    foreach "outgoing first neighbour"
        SUB: find chains
    combine chain and core IA -> active subnet IA

OUTPUT section
    make nodes.csv
    make edges.csv
    make pajek.net
    make pajek-like.net for degree clustering

SUBROUTINES
find chains for input node {
    STOP if (input node has no targets/ is a "dead end")
    else
        if (all targets have no targets/are "dead ends")
            make input node "dead end"
        foreach target of input node
            if (target is "ingoing first neighbour")
                remember IA (input node to target)
                if (SUB: check if all paths walked = TRUE)
                    make input node "first ingoing neighbour"
            else
                if (target was already seen before)
                    foreach target of target ("t.of.t")
                        if (t.of.t is "ingoing first neighbour")
                            if (SUB: check if all paths walked = TRUE)
                                make target "first ing. Neighbour"
                        else
                            if (all t.of.t seen before)
                                if (SUB: try to break loop = TRUE)
                                    remember loop IAs

```

```

else
    remember loop IA for removal
unless (t.of.t is "dead end")
    remember IA (target -> t.of.t)
    SUB: find chains for t.of.t
else
    unless (t.of.t is "dead end")
        remember IA (target -> t.of.t)
        SUB: find chains for t.of.t
    return (remembered IA, IA for removal, dead_ends, first ingoing neighbours)
}

check all paths walked {
    count outgoing IAs which were already remembered
    count targets that are "dead ends"
    if (sum of the above = total number of targets)
        all paths walked = TRUE
    else
        all paths walked = FALSE
}

try to break loop {
    foreach target of start of loop
        if (seen target)
            SUB: try to break loop(target)
        else
            remember loop IA
    if (remember loop IA != 0)
        return "TRUE"
    else
        return "FALSE"
}

```

Box 3: Pseudocode of `active_subnet.pl`. This script goes through the interactions of the sample-specific net and derives all interactions that are dynamically relevant for expressed TF. It first creates a set of 'core' interactions that have expressed targets or regulators or both, and interactions that connect "first neighbours" - unexpressed regulators or targets of expressed TF. A very important part of the script is a subroutine that searches for chains of interactions that start at first neighbours, go along unexpressed TF and end on a first neighbour again, thereby feeding back to the core of expressed TF. The full code of `active_subnet.pl` can be found in the supplement.

2.2.6 Calculation of standard network measures

Networks were loaded in *GEPHI07 beta*⁵⁰ in pajek net format and standard network measures were computed using the software's functions (Avg. degree, network diameter, graph density, modularity, avg. path length)

2.2.7 Detection of network motifs

We extracted all edge-describing lines from active subnet pajek-files and removed self-edges to prepare the networks for motif detection in MFINDER⁵¹. We then detected all 3-node sub-graphs significantly more frequent than in 1000 randomized networks with a z-score > 2 and a p-value < 10⁻². We selected the option to report all network motif occurrences along with the TFs present inside them (*-omem*).

2.2.8 Comparison of active subnets

2.2.8.1 Jaccard index of networks based on common edges

In order to detect similarity between active sub-networks of different samples, we calculated the Jaccard index (number of edges present in both samples divided by union of edges from both samples). We computed Jaccard indices based on a) all network edges and b) edges between TF that were expressed in both samples. Calculations were done in R⁵² and color-coded tables were generated using the “Conditional Formatting” option “Colour Scale” in Microsoft Excel.

2.2.8.2 Normalized network degree clustering

Normalized network degree clustering is a method to assess structural network similarity, but it has also been used to cluster TF-TF interaction networks by their function.²⁵ Networks are clustered by how similar the degrees of each of their TFs are. For each TF in the network, the degree is normalized by dividing it by the number of edges. Normalized degrees are then ordered in a vector and the Euclidean distance between vectors is computed. Networks are then clustered according to their distance to one another. We carried out this analysis as described in [25Fehler! Textmarke nicht definiert.] using the R packages *igraph*⁵³ and *stats*.

2.2.8.3 Multidimensional scaling of networks

To assess similarity between multiple networks at the level of edges rather than at the level of node degrees, we performed metric multidimensional scaling (MDS). For this analysis, we first created an overall edge set that comprised the union of all edges from all samples. We then observed for each network which edges from the overall set it contained, and removed edges that were present in all samples to improve the resolution of the final plot. From these observations, we created a binary matrix (0 signified the absence and 1 the presence of an edge) and calculated its Euclidean distance. This was then used as an input for the R function *cmdscale()*. We performed MDS in 2D and 3D, making use of the R packages *plyr*⁵⁴ and *plot3D* (available at <http://cran.r-project.org/web/packages/plot3D/>).

2.2.9 Functional enrichment analysis

We carried out functional enrichment analysis using DAVID^{55,56} to investigate a) if the TF present in active subnets reflected cell type or tissue specificity on a functional level and b) if TFs related to cell-type or tissue function were enriched on significant network motifs. We referred to the latter as “marker TFs”. In all analyses, only Gene Ontology (GO) terms⁵⁷ representing Biological Processes were examined since they best described cell type or tissue functions. We reported GO terms as enriched when they matched following cutoff scores: p-value < 0.1, fold change > 1.5 and Benjamini $\leq$ 0.05.

For the background in all functional enrichment analyses we used organism-specific, reviewed transcription factors sets which we obtained by querying UniProt KB. We retrieved the data on 13.2.2014 using the queries (("transcription factor" AND organism:"homo sapiens") AND reviewed:yes) and (("transcription factor" AND organism:"mus musculus") AND reviewed:yes).

2.2.9.1 Functional enrichment in active subnets

When we assessed whether the TFs present in active subnets reflected cell type or tissue function, we first created a set of “specifically expressed” TF. For this purpose, we subtracted collectively expressed TFs from the expressed TFs of a target sample and examined the resulting specific set for functional enrichment. To obtain collectively expressed TFs, we pooled the expressed TFs of all other non-cancer samples without including related samples (*i.e.* when listing Hsmm-specific expressed TFs, we did not include the expressed TF of Lhcnm2 and Lhcnm2diff in the collectively expressed genes because all three are muscle cell types).

For mouse samples, we obtained collectively expressed TFs by pooling expression from all other tissues, since no disease conditions or closely related tissues were present in this sample group.

2.2.9.2 Functional enrichment in network motifs

To investigate if “marker TFs” related to cell type or tissue function were enriched on network motifs, we collected the TFs of each specific network motif which had previously been established as significant. From these motif-specific TF sets, we removed all unexpressed TFs and carried out functional enrichment analysis on the resulting gene lists.

3 RESULTS

In the following, we present results in an analogous order to the steps in the workflow, starting with the reference network, followed by the sample-specific transcriptomes (RNA-seq) and regulatory site occupancy screens (DNase-seq). We then continue with the characterization of sample-specific networks that are generated by combining the reference networks with experimental evidence from RNA and DNase-seq. Finally, we present the active sub-networks that were derived from the sample-specific networks. This chapter thus focuses on the results of the different analyses applied to uncover structural properties of the active sub-networks.

3.1 Reference networks

We use the term “reference network” to describe the union of all predicted interactions in any tissue or developmental stage. Notably, the reference network will become more fine-grained and false-positive predictions will be reduced by adding different evidence types in the next steps.

3.1.1 Human reference network

The human reference network (HRNW) contained 404 human TFs represented by nodes. The total number of edges that represent potentially possible interactions between these TFs was 58,454. This is less than a total predicted from binding motifs (681,952). The edge number is derived from the following considerations.

As previously explained, a direct edge from the regulator to the target was set when a regulator-TF’s binding motif was predicted in the regulatory regions of a target TF. From our definition of regulatory regions (intronic or intergenic regions inside a $\pm$ 5kb interval around the transcription start site, TSS) it follows that a regulatory region belongs to a specific transcript, namely that transcript from whose specific intron/exon structure it was derived. It follows also that the binding motifs and the edges are transcript-specific.

Because of overlapping isoform-specific regulatory regions or shared TSS of transcripts, the detected binding motifs were often redundant, as the same DNA sequence was recognized multiple times as a binding motif. Moreover, predicted binding motifs frequently overlapped one another due to similar binding affinities of their TFs, and intersections ranged from only a few bp to complete overlaps. If entirely overlapping, binding motifs were collapsed into one, reducing the total of detected binding motifs to 312,424.

When the information from binding motifs was projected to the level of interactions, binding motifs that represented interactions of the same regulator TF with the same target TF isoforms were removed. Through this exclusion, TF to TF isoform interactions were made unique as well. Their total, non-redundant number was 234,378.

In the last step, we mapped interactions from isoforms to TFs to enable a characterization of its properties with standard network measures. This mapping reduced the interactions to the above mentioned total of 58,454 edges between TFs. For a better overview, we present the different

binding motif and edge numbers for HRNW in Table 2. Along with HRNW, we show analogous numbers for the mouse reference network (MRNW).

	HRNW	MRNW
Total number of binding motif detections	681,952	387,812
Number of binding motifs after removal of entirely overlapping motifs	312,424	214,930
Number of edges / interactions between TFs and TF isoforms	234,378	128,976
Number of edges / interactions between TFs (= final edge number in reference networks)	58,454	49,202

Table 2: Overview of binding motifs and edge numbers in the human reference network (HRNW) and mouse reference network (MRNW). The raw output from binding motif detection is shown in the top row, the two middle rows show numbers after correcting for different types of redundancy (overlapping binding motifs, binding motifs that describe the same TF to TF isoforms interaction), and at the bottom the final edge numbers in HRNW and MRNW are presented.

Along with the node and edge numbers, we have compiled other standard network measures that characterize HRNW in Table 2. As a reference, we provide the same measures for a pooled network (pooled NW) which we constructed by taking the union of all cell type-specific networks published by Neph et al.²⁵ **Fehler! Textmarke nicht definiert.** For a more similar comparison, we also include values for the “reduced” versions of both networks. In this case, we reduced the number of nodes to 350 TFs that were present in both data sets and removed all edges that were not related to these 350 corresponding TFs.

	HRNW	Pooled NW	Reduced HRNW	Reduced pooled NW	MRNW
Nodes	404	538	350	350	388
Edges	58,454	47,945	31,275	27,142	49,202
Avg. degree	289	178	178	155	254
NW diameter	3	4	3	4	3
Graph density	0.354	0.166	0.256	0.222	0.328
Modularity	-0.004	-0.003	-0.004	-0.005	-0.004
Avg. path length	1.6	1.9	1.6	1.8	1.6

Table 3: Standard network measuresⁱⁱ for: human reference network (HRNW), a pooled network constructed by integrating the edges from the published regulatory networks of 41 different human cell types (pooled NW), the reduced versions of both networks which contain only TFs found in both data sets and the edges between them (reduced HRNW and reduced pooled NW), mouse reference network (MRNW).

As Table 3 shows, HRNW differs from the pooled NW most prominently in the number of nodes and edges, the average degree and graph density. Both latter measures can be attributed to the higher number of edges and the lower number of nodes in HRNW. When the networks are

ⁱⁱ Explanation of network measures: Average degree: The average number of edges a node has; Network diameter: Longest graph distance between any two nodes in the network; Graph density: How close a network is to complete. A complete network as all possible edges and a density of 1; Modularity: Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules (source: [http://en.wikipedia.org/wiki/Modularity_\(networks\)\)](http://en.wikipedia.org/wiki/Modularity_(networks))); Average path length: Average graph distance between all pairs of nodes. Unless specified, definitions for network measures were taken from GEPHI network analysis software.

reduced to corresponding TFs / nodes, all values become more comparable, indicating structural similarity. Notably, the networks exhibit low modularity in all cases which is interesting because biological networks have been shown to exhibit modularity.⁵⁸ To explain this, one could argue that both networks are not functional units. The pooled NW is a union of several cell type-specific networks and HRNW is based only on the prediction of TF binding motifs. To extract functional units, the experimental interaction data were mapped onto the edges of the reference network.

In order to compare our results with the cell-type specific networks generated by Neph et al, it is necessary to explain the methodical differences in generating the two networks. Both studies used a +/-5kb interval around TSS to define regulatory regions. However, Neph's networks were the result of a probabilistic modeling approach combined with binding motif occupancy data from DNaseI-seq. Therefore, edges in their networks represent the presence of DNaseI footprints on predicted binding motifs. On the contrary, edges in HRNW simply represent predicted binding motifs. Furthermore, Neph et al did not distinguish binding motifs in introns and exons, while we restricted the reference network to motifs located in introns or intergenic regions. Finally, the TFBS models from Neph et al were retrieved from the licensed TRANSFAC database, while we used models from publicly available HOCOMOCO. Although 350 TFs in the network were shared, different TFBS models could have lead to the detection of other binding motif locations, resulting in different interactions (Figure 5).

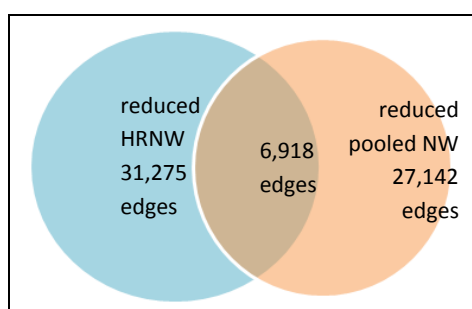


Figure 5: Venn diagram showing common edges in the reduced human reference network (HRNW) and reduced pooled network. 6,918 edges are common to both networks, which is about 22% of the reduced version of HRNW and 25% of the reduced pooled NW. Even though the same TFs make up the nodes of these two networks, different TFBS model databases have been used, resulting in the detection of other binding motif locations / interactions. The differences can also be attributed to the restriction of regulatory regions to introns and intergenic regions in HRNW. Binding motifs within exons were used to predict interactions in the reduced pooled NW, but not in HRNW.

From this analysis we expected the two networks not to be identical. However, as visible in Table 3, the overall topological features are similar for HRNW and the pooled NW.

3.1.2 Mouse reference network

Due to the conservation of genomic sequences between the mouse and the human, we expected to find a comparable number of edges and similar values for the standard network measures in HRNW and the mouse reference network (MRNW). Since the same TFBS models were used for the detection of binding motifs, sequence similarity in homologous regulatory regions would lead to the detection of conserved binding motifs. Weak conservation of sequences in regulatory regions, along with changes of intron/exon structure in the course of evolution would however contribute to differences in the detection of motifs.

Applying the same concept of reference network construction as for HRNW, we found 388 TF nodes and 49,202 interactions for MRNW (Table 3).

To assess the structural similarity between HRNW and MRNW, we first reduced the set of edges from HRNW to those which exclusively contained TFs that were present in MRNW. We thereby obtained 38,614 HRNW edges that corresponded to MRNW. Of these corresponding edges, 22,031 (57 %) were also present in MRNW (Figure 6).

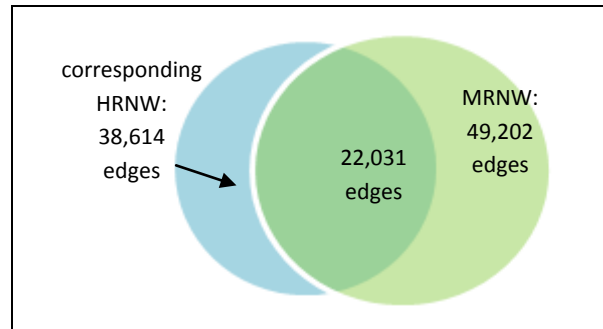


Figure 6: Venn diagram showing common edges in HRNW and MRNW. Prior to overlapping the networks, HRNW was reduced to those edges that corresponded to the TFs present in MRNW, creating “corresponding” HRNW. 22,031 edges were present in both networks (57 % of corresponding HRNW, 45% of MRNW).

As expected, the values of standard network measures and topological features for MRNW and HRNW are similar (Tables 2 and 3).

3.2 Identification of expressed TF through RNA-seq

Ready-made alignments that had been mapped to reference genomes with standard tools (Tophat²⁶, STAR²⁷) were available for all samples. On these alignments, we applied the widely-used Cufflinks.⁴⁰ RNA-seq analysis software. The obtained expression values were normalized with respect to transcript length and the number of mapped reads (as Fragments Per Kilobase of transcript per Million mapped reads, FPKM). We first examined FPKM distribution, seeking to determine an appropriate expression cutoff. We also present the number of genes and TFs expressed above the cutoff and document the expression of housekeeping genes.

3.2.1 FPKM distribution

Since the FPKM distribution followed a similar pattern in all samples, we show the distribution in replicate 1 of the human Hsmm (human skeletal muscle myoblasts) sample as an example (Figure 7).

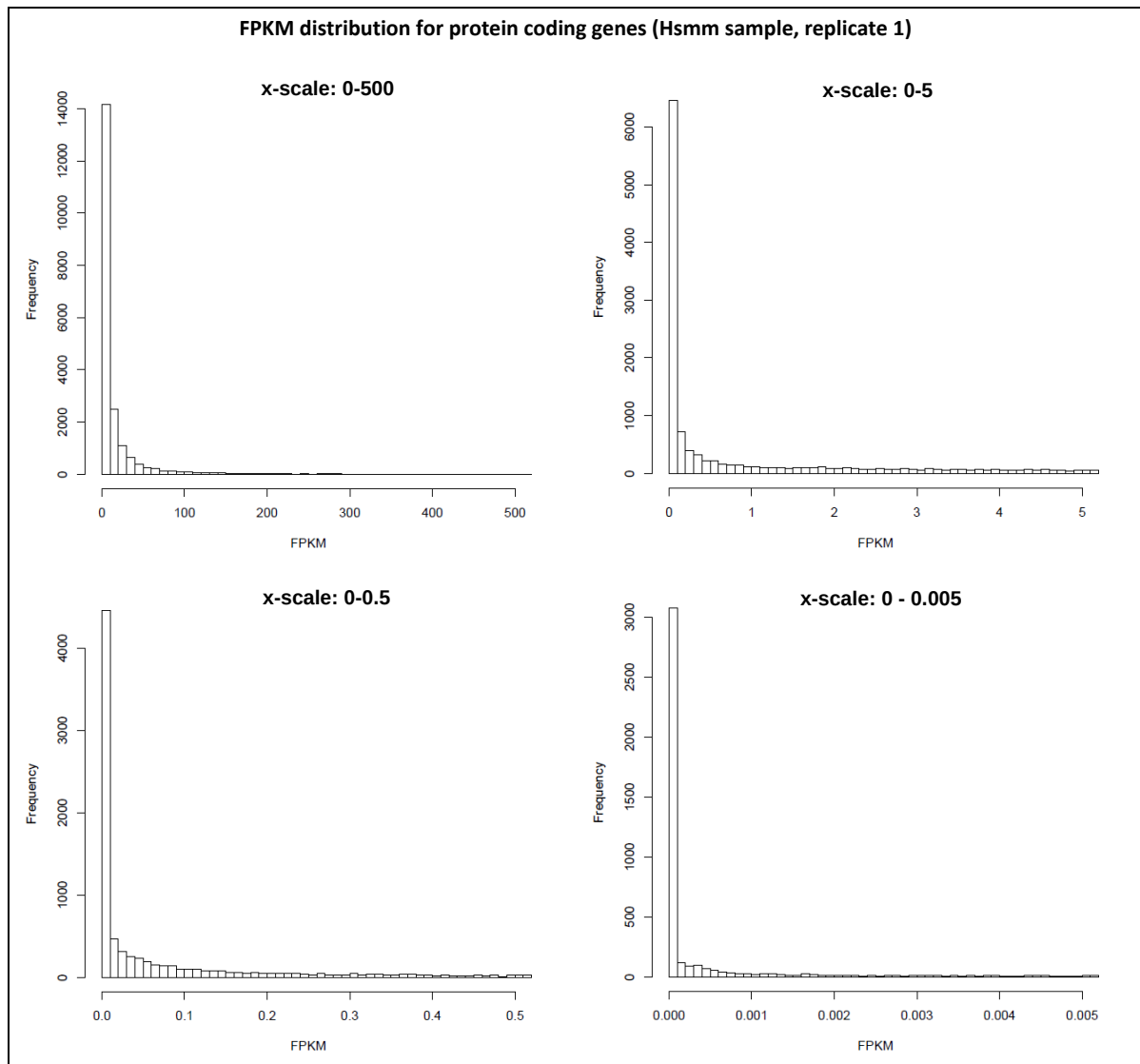


Figure 7: FPKM distribution for protein coding genes in the HSMM sample, replicate 1. The distribution is left-skewed with a single peak at 0 FPKM and a long tail to the right. We examined different levels of magnification and found that this distribution persisted in all of them.

As visible in Figure 7, it was impossible to choose an expression cutoff that would be justified by the distribution of FPKMs. Therefore, we decided to refer to a cutoff from the literature. In accordance with a publication that characterized expressed genes in the developmental transcriptome of *Drosophila melanogaster*, we chose a cutoff of $\text{FPKM} > 1$.⁴⁷

3.2.2 Housekeeping genes are expressed above expression cutoff

To confirm that our chosen cutoff captured expressed genes, we examined the expression of housekeeping genes in each replicate (Table 4 and Table 5). In all cases, housekeeping genes were expressed above the cutoff of $\text{FPKM} > 1$. We referred to published lists of housekeeping genes for human⁵⁹ and mouse⁶⁰.

		Human housekeeping genes										
		RAB7A	SNRPD3	GPI	VPS29	PSMB2	REEP5	CHMP2A	EMC7	C1orf43	PSMB4	VCP
Replicates of human RNA-seq samples	A549 1	103	108	229	101	36	97	110	82	183	216	269
	A549 2	107	80	252	109	39	78	132	111	157	195	206
	Cd20 1	38	44	35	51	10	13	66	12	48	199	55
	Cd20 2	46	93	55	67	11	20	56	11	50	191	51
	HepG2 1	80	121	186	39	23	24	22	46	98	97	84
	HepG2 2	83	85	169	36	24	28	27	61	102	131	96
	HepG2 3	114	62	81	62	26	134	28	64	181	218	89
	HepG2 4	99	60	68	55	22	121	29	56	178	207	100
	Hsmm 1	128	95	143	56	44	44	75	65	107	176	196
	Hsmm 2	155	124	144	48	50	32	84	49	106	190	207
	Hsmm 3	228	267	198	44	63	64	110	41	229	307	169
	Hsmm 4	272	142	88	73	54	51	89	56	164	195	159
	Huvec 1	145	123	139	107	49	69	64	47	77	223	310
	Huvec 2	109	94	121	79	33	48	81	36	71	163	283
	Huvec 3	160	129	72	64	63	98	73	30	171	511	188
	Huvec 4	175	137	87	78	69	89	85	35	172	480	207
	Imr90 1	189	161	170	69	47	45	101	46	122	319	185
	Imr90 2	223	128	175	69	48	75	116	44	125	366	183
	K562 1	137	182	228	97	105	24	79	58	91	265	163
	K562 2	87	75	193	84	60	19	84	33	176	169	98
	K562 3	123	83	53	138	65	46	44	44	151	259	138
	K562 4	89	69	149	107	62	43	47	34	203	248	120
	Lhcnm2	173	92	161	66	46	62	140	57	103	231	223
	Lhcnm2 diff	146	74	118	37	34	74	117	57	120	172	190
	Monocd14 1	132	58	75	122	15	42	123	23	73	174	54
	Monocd14 2	175	89	147	62	24	37	153	17	106	214	45
	Nhlf 1	189	101	164	65	48	41	142	70	91	233	221
	Nhlf 2	165	78	154	68	35	41	108	55	79	171	178
	Nhlf 3	379	127	110	96	47	64	150	61	116	247	186
	Nhlf 4	293	81	98	107	33	62	111	60	114	191	165
	Sknshra 1	146	72	88	48	37	108	132	50	153	337	148
	Sknshra 2	148	56	100	40	35	111	136	44	187	366	171

Table 4: Expression of human housekeeping genes in FPKM, all replicates across all RNA-seq samples are shown. The genes in this compilation had been chosen based on the expression of their exons which have a “geometrical mean expression exceeding RPKM = 50”.⁵⁹ In this study, FPKM was used instead of RPKM as a normalization measure, but the measures are similar. In the majority of samples, the value of 50 FPKM is exceeded. In all cases, the expression cutoff of FPKM > 1 is met.

		Mouse housekeeping genes							
		Puf60	Psmd4	Eif3f	Heatr3	Chmp2a	Leng8	Rpl37	Rpl38
Replicates of mouse RNA-seq samples	Heart 1	34	52	28	11	37	110	126	111
	Heart 2	44	59	40	7	34	147	191	159
	Kidney 1	73	66	46	7	48	230	290	189
	Kidney 2	64	62	51	7	62	111	361	317
	Liver 1	56	75	31	7	42	50	160	134
	Liver 2	60	75	30	7	35	98	151	140
	Lung 1	86	60	56	6	47	599	335	196
	Lung 2	103	56	89	6	39	588	321	222
	Thymus 1	131	86	140	8	27	590	262	198
	Thymus 2	122	95	73	8	31	646	305	212
	Wbrain 1	113	115	36	16	59	191	348	663
	Wbrain 2	84	72	47	13	49	157	283	528

Table 5: Expression of mouse housekeeping genes in FPKM, all replicates across all RNA-seq samples are shown. The genes in this compilation have been chosen based on their expression across 15 different mouse samples.⁶⁰ For all tissues and genes, expression values are higher than the chosen expression cutoff of FPKM > 1.

3.2.3 Total number of expressed genes is comparable across samples

Furthermore, we took notes of the total number of genes expressed above the cutoff (Table 6). We examined all replicates to check if there were outliers with a very different total. Since the sequencing depth was similar for all samples, we expected the number of expressed genes to be relatively constant. For human samples, it varied between 11,000 and 16,000, and for mouse samples between 10,000 and 14,000. Even human samples from different cancer cell types and cell lines (A549, HepG2, K562, Lhcnm2, Sknshra) were not found to have a strongly variable total number of expressed genes.

Human sample, replicate no.	Number of genes above FPKM 1	Human sample, replicate no.	Number of genes above FPKM 1	Mouse sample, replicate no.	Number of genes above FPKM 1
A549 1	13,448	Imr90 1	12,767	Heart 1	12,044
A549 2	13,244	Imr90 2	12,648	Heart 2	12,214
Cd20 1	12,816	K562 1	13,153	Kidney 1	13,075
Cd20 2	13,008	K562 2	14,189	Kidney 2	13,229
HepG2 1	13,417	K562 3	12,359	Liver 1	10,604
HepG2 2	13,199	K562 4	12,529	Liver 2	10,857
HepG2 3	12,828	Lhcnm2	15,584	Lung 1	14,037
HepG2 4	12,876	Lhcnm2diff	16,024	Lung 2	13,604
Hsmm 1	13,294	Monocd14 1	11,934	Thymus 1	13,071
Hsmm 2	13,334	Monocd14 2	11,546	Thymus 2	13,382
Hsmm 3	13,050	Nhlf 1	13,262	Wbrain 1	13,711
Hsmm 4	12,778	Nhlf 2	13,116	Wbrain 2	13,655
Huvec 1	15,998	Nhlf 3	12,902		
Huvec 2	15,757	Nhlf 4	12,757		
Huvec 3	12,761	Sknshra 1	14,575		
Huvec 4	12,947	Sknshra 2	14,591		

Table 6: Total number of genes expressed above the cutoff of FPKM > 1 for each replicate in human and mouse samples. When different replicates of the same sample are examined, the number of expressed genes is similar. Samples within one organism differ, but there are no outliers with dramatically larger or smaller numbers.

3.2.4 Expression calling on transcription factors

We assessed the expression of TFs using the expression cutoff described above and applied it to all protein coding isoforms of a given TF. We carried out the expression calling of the level of isoforms because for many TFs, there were non-protein coding transcript types that had an influence on the expression value calculated for the TF gene. However, our main interest was the expression of functional, DNA-binding TF proteins. For that reason, we defined TFs as expressed when at least one of protein coding isoforms of the given TF was expressed above the cutoff. Considering replicates, we defined isoforms as expressed by decision of majority wherever possible, and by average expression across replicates in all other cases.

Human samples	Number of expressed TFs	Mouse samples	Number of expressed TFs
A549	239	Heart	224
Cd20	202	Kidney	256
Hepg2	224	Liver	201
Hsmm	224	Lung	269
Huvec	231	Thymus	245
K562	205	Whole brain	252
Lhcnm2	215		
Lhcnm2 diff	221		
Monocd14	204		
Nhlf	239		
Sknshra	242		

Table 7: Number of expressed TFs in human and mouse samples. Of the 404 possible human TFs, 200-250 were called as expressed in human samples. For mouse samples, where 388 TFs were assessed, the number of expressed TFs was roughly within the same range.

For both human and mouse samples, the number of expressed TFs lies approximately within a range of 200-250 (Table 7). Among mouse samples, liver has considerably fewer expressed TFs than all of the others. The result of the expression calling forms one of the bases for the identification of sample-specific interactions and is also critical for the functional enrichment analysis which will be carried out on expressed TFs in chapter 3.5.5.

3.3 DNaseI footprinting in DNaseI-seq data

In the following we present the results of DNaseI footprinting carried out in DNaseI-seq data from different human cell types and mouse tissues. As we found strong dependency of results on the algorithm used to derive footprints from DNaseI, we first report a comparison of footprints that have been derived for six human cell types by two different algorithms, *DNaseR*⁴⁸ and *DGF*⁶¹, with *DNaseR* being the algorithm that was used in this study. Next, we present the result of the combination of human reference network (HRNW) and mouse reference network (MRNW) with DNaseI footprints from their corresponding cell types and tissues. Finally, we compare the number of DNaseI footprints in introns/intergenic regions with their number in exons, addressing our model assumption that exons and coding sequences would not contain TFBS relevant for the regulation of transcription.

3.3.1 Comparison of DNaseI footprinting results with published DNaseI footprints

We identified DNaseI footprints across 12 human samples with a p-value cutoff of e^{-3} . In Table 8 we present an overview of footprints identified for the ENCODE DNaseI seq data sets using the *DNaseR* algorithm and the *DGF* algorithm.³⁰ We show the number of footprints identified in each sample and the percentage of footprints identified by both algorithms. Both algorithms were set to detect footprints of a width 6-40 bp. Here, we report footprints detected by both algorithms that have an overlap of at least 1, 5 or 10 bp. This is necessary because the algorithms specify the edges of footprints upon different criteria and completely overlapping footprints are expected to be rare.

		Number of FP derived by <i>DNaseR</i> (p-val e^{-3})	Number of FP derived by <i>DGF</i> (FDR 1%)	% of DGF FP identified by <i>DNaseR</i>			
				1 bp overlap	5 bp overlap	10 bp overlap	exact overlap
Human cell types	Cd20	4,750,733	603,190	6.2	6.0	1.5	0.10
	Hepg2	5,191,674	448,678	6.9	6.8	2.0	0.07
	Hsmm	1,244,576	1,668,243	10.5	9.9	3.7	0.06
	K562	6,997,479	498,683	7.2	7.1	1.7	0.11
	Nhlf	1,002,711	1,567,106	10.6	10.2	4.3	0.05
	Sknshra	3,772,166	498,926	6.2	6.1	1.7	0.06

Table 8: Overview of DNaseI footprints identified in the same data sets by the *DNaseR* and the *DGF* algorithms. The percentage of footprints detected with both algorithms is shown to the right. Since differences in the algorithms lead to shifts in the determination of footprint edges, we report the percentage of overlapping footprints with different overlaps: 1, 5 or 10 bp. With an overlap of 1 to 5 bp 6-11% of the footprints identified by *DGF* area also detected by *DNaseR*. With an overlap of 10 bp, the value drops to 2-4%. Exact matches range from 0-0.1% across samples.

The comparison shows that the overall number of detected footprints is different for the two algorithms (Table 8). The disparate criteria used to define footprints and the statistical methods that control significance (p-value) and multiple testing (false discovery rate, FDR) are likely to contribute to the difference. Remarkably, neither of the algorithms detects more footprints in all samples (*DGF* identifies more footprints than *DNaseR* in Hsmm and Nhlf while *DNaseR* detects more footprints in the other cell types). This could indicate that neither of them is the more sensitive one.

The numbers of footprint detections and their proportion to each other vary. The percentage of footprints identified by both algorithms is within the range of 6-11% for 1 and 5 bp overlaps, and in the range of 2-4% for 10 bp overlaps across all samples. An extremely small fraction of footprints is reported with exactly the same genomic position by the two algorithms, 0-0.1%. For those samples where the larger number of footprints was reported by the *DGF* algorithm, the percentage of footprints that were also identified by *DNaseR* is higher than in other samples.

3.3.2 Combination of DNaseI footprints with reference networks

In Table 9 we show across all human and mouse samples how many of the predicted binding motif locations in the reference networks HRNW and MRNW were matched by DNaseI footprints. As mentioned in chapter 2.1.1, there was a total of 681,952 predicted binding motifs in HRNW, and for MRNW, 387,812 binding motifs were predicted.

Human samples	Number of predicted binding motifs overlapped by FP	% of predicted binding motifs from HRNW overlapped by FP	Mouse samples	Number of predicted binding motifs overlapped by FP	% of predicted binding motifs from MRNW overlapped by FP
A549	53,808	7.9	Heart	23,006	5.9
Cd20	58,652	8.6	Kidney	7,817	2.0
Hepg2	75,592	11.1	Liver	2,265	0.6
Hsmm	32,907	4.8	Lung	13,257	3.4
Huvec	51,024	7.5	Thymus	14,330	3.7
K562	80,550	11.8	Whole brain	10,539	2.7
Lhcnm2	37,300	5.5			
Lhcnm2diff	39,131	5.7			
Monocd14	48,284	7.1			
Nhlf	38,920	5.7			
Sknshra	63,155	9.3			

Table 9: Number of predicted binding motifs from the human and mouse reference network (HRNW and MRNW) that overlapped DNaseI footprints. Both reference networks are reduced markedly through overlapping them with footprints.

When combining the evidence from DNaseI footprinting with reference networks, we considered predicted binding motifs as overlapped and confirmed by DNaseI footprints when they shared at least 3 bp of overlap. This is an established approach which has been previously used to characterize binding motif occupancy.²⁵

Table 9 shows that only a small fraction of the predicted binding motifs coincide with footprints. It is likely that many binding motifs with no overlap are false-positive predictions. However, it is impossible to distinguish unbound true binding motifs from falsely predicted motifs. To approach this problem, we examine the overlap of footprints and predicted binding motifs across 12 human cell-types or 6 mouse tissues. This is a first step towards addressing the problem, and the sample number should be increased in future work.

Human samples	Total number of detected DNaseI FP	Mouse samples	Total number of detected DNaseI FP
A549	4.523.642	Heart	4.000.137
Cd20	4.750.733	Kidney	1.583.736
Hepg2	5.191.674	Liver	650.990
Hsmm	1.244.576	Lung	1.688.380
Huvec	1.971.345	Thymus	1.838.503
K562	6.997.479	Whole brain	1.832.342
Lhcnm2	3.376.895		
Lhcnm2diff	2.968.808		
Monocd14	2.416.100		
Nhlf	1.002.711		
Sknshra	3.772.166		

Table 10: Overall number of detected DNaseI footprints in human and mouse samples. Mouse heart and liver diverge from the other samples in this comparison. When the corresponding counts of footprinted binding motifs are examined (Table 9), the same mouse samples stand out, suggesting that the deviations have their origin in the overall detection count.

From the comparison of Table 10: Overall number of detected DNaseI footprints in human and mouse samples. Mouse heart and liver diverge from the other samples in this comparison. When the corresponding counts of footprinted binding motifs are examined (Table 9), the same mouse samples stand out and Table 9 we can see some correlation of the overall detection count and

the number of predicted binding motifs overlapped by footprints. In human samples, those that have most footprinted motifs also have most detected footprints (Hepg2, K562). However, Lhcnm2diff has the second lowest number of footprinted motifs even though its overall detection number is intermediate. This demonstrates that there is no strict correlation between the two measures.

3.3.3 DNaseI footprints in introns / intergenic regions and exons

In spite of our focus on regulatory regions (excluding exons), we examined the numbers of DNaseI footprints and footprints overlapping predicted binding motifs within introns / intergenic regions as well as within exons. For this purpose we adjusted the program which we used to extract introns and intergenic regions within +/- 5kb around the TSS as regulatory regions (chapter 2.2.1.2) to output exons in the same interval. We then compared across exonic and intronic / intergenic regulatory regions: the number of DNaseI footprints mapping to regulatory regions, their relative number with respect to total regulatory region length, and the number of footprints that overlapped binding motifs predicted in regulatory regions (Table 11).

Human samples	Exonic regulatory regions			Intronic / intergenic regulatory regions		
	FP in regions	Relative number of FP	FP overlapping predicted binding motifs	FP in regions	Relative number of FP	FP overlapping predicted binding motifs
A549	3,834	1	2	19,724	1	7,343
Cd20	4,659	2	2	23,154	1	8,515
Hepg2	5,542	2	2	23,196	1	8,881
Hsmm	2,558	1	0	9,094	0	3,598
Huvec	3,857	1	0	15,566	1	5,844
K562	5,470	2	4	32,101	2	10,934
Lhcnm2	2,629	1	2	14,033	1	4,756
Lhcnm2diff	2,518	1	1	12,991	1	5,087
Monocd14	3,172	1	2	13,039	1	5,177
Nhlf	2,695	1	3	8,471	0	3,576
Sknshra	4,400	2	0	17,375	1	6,956
Total length of exonic regulatory regions: 2,708,979 bp						
Total number of predicted binding motif locations in exonic regulatory regions: 56,233						
Total length of intronic / intergenic regulatory regions: 19,251,465 bp						
Total number of predicted binding motif locations in intronic/intergenic regulatory regions: 312,424						

Table 11: Number of DNaseI footprints and footprints that overlap predicted binding motifs in exonic and intronic / intergenic regulatory regions. The relative number of DNaseI footprints is comparable for the two types of regulatory regions (relative number = FP number divided by total length of regulatory regions, multiplied by 1000). However, footprints that overlap predicted binding motifs occur at a much higher frequency in intronic / intergenic regulatory regions.

We see from this comparison that a higher number of footprints map to intronic / intergenic regulatory regions compared with exonic regions. From the relative number of footprints, it is apparent that this difference is a consequence of the much higher total length of the intronic / intergenic regulatory regions. (Relative number of FP = FP number divided by total length of regulatory regions, multiplied by 1000).

Considering only those DNaseI footprints that overlap predicted binding motifs, a difference between exons and introns / intergenic regions is maintained. Remarkably, the number of

footprints that overlap binding motifs in introns / intergenic regions is on average 1000 fold that in exonic regulatory regions. This difference persisted when the total number of predicted binding motif locations in each of the groups was taken into account.

We regarded DNaseI footprints that overlapped predicted binding motifs as markers of binding events involved in transcriptional regulation. The data reveals very few actual binding events on the predicted binding motifs in exonic regulatory regions.

3.4 Sample-specific networks

For the sample-specific networks, the elements that we described up to this point were combined: the reference network, the expression calls from RNA-seq and the binding motif occupancy profiles from DNaseI-seq. By integrating these different data types we confirmed predicted edges from the reference network, assigned interaction types and provided weights, specifically for each cell type / tissue. Our approach made use of the previously applied strategy to verify predicted binding motifs by overlapping them with DNaseI footprints²⁵

Fehler!
Textmarke nicht definiert., and furthermore introduced gene expression to enable differentiation between activating and inhibiting interactions, applying a set of logical rules. We developed these rules on the basis of different scenarios of transcriptional regulation and implemented them algorithmically.

Since our methodology is new, we first recapitulate how interaction types and weights were assigned and then present their distribution in the sample-specific networks. As a validation of our networks, we confirmed that the cell type-specific network generated for human skeletal muscle myoblasts (Hsmm) contains TFs typical for muscle-specific transcription regulation. For this purpose, we compared the Hsmm-specific network with a small, well-studied sub-network from the literature.

3.4.1 Distribution of weighted interaction types

From the combination of sample-specific experimental evidence with reference networks, we generated specific networks for 12 human cell types and 6 mouse tissues. We developed rules to classify interactions into three different weight groups based on the evidence that supported them:

A – highest weight: supported by expression, DNaseI footprints and binding motif prediction

B – intermediate weight: supported by expression, absence of DNaseI footprints and binding motif prediction

P – lowest weight: supported by prediction from the reference network.

For all weights and interactions, the model assumption was that if binding motif occupancy is confirmed through a DNaseI footprint, the TF related to this binding motif actively regulates the TF in whose regulatory region the binding motif is located. From this assumption, following conclusions/ rules followed:

A) Weight A: For expressed TFs that were regulators in predicted edges, interactions were assigned when a binding motif of the regulator in the regulatory region of the target was “footprinted” (*i.e.* overlapped a DNaseI footprint). If the target was expressed, the interaction was considered activating and termed +A, if not, the interaction was regarded as an inhibition and termed -A.

B) Weight B: We expected binding motifs belonging to unexpressed, inactive regulators to be devoid of DNaseI footprints. When none of the binding motifs in the entire regulatory region of their target TF were footprinted, we attributed the regulatory effect to the absence of the aforementioned inactive regulator. If the target was expressed, this was regarded as an inhibitory interaction and termed -B. Conversely, if both regulator and target were unexpressed, the regulator’s action was considered activating and the interaction was called +B.

C) Weight P: When a regulator-TF was not expressed and any binding motif in the target’s regulatory region was footprinted, we assigned an interaction of weight P. In this case it is not clear whether there is a regulatory effect or not. The binding motif of the regulator could simply be a false-positive prediction, but it could also be true. For true motifs, the sign of the interaction is uncertain, because its effect is overshadowed by a different, expressed regulator whose action is confirmed by DNaseI footprints.

		Distribution of weighted interaction types in NW					Total number of interactions in NW
		+A	-A	+B	-B	P	
Human Cell types	A549	4,526	2,735	446	0	21,341	29,048
	Cd20	4,029	3,255	83	42	26,665	34,074
	Hepg2	4,903	3,660	50	0	24,227	32,840
	Hsmm	2,966	1,201	1,670	142	22,863	28,842
	Huvec	4,354	1,684	227	0	24,274	30,539
	K562	5,485	3,420	252	0	27,033	36,190
	Lhcnm2	2,823	1,658	1,504	329	24,182	30,496
	Lhcnm2 diff	3,467	1,606	634	274	23,863	29,844
	Monocd14	3,410	2,127	468	0	26,246	32,251
	Nhlf	3,195	1,079	1,659	209	21,221	27,363
	Sknshra	4,138	2,821	133	226	22,836	30,154
Mouse tissues	Heart	2,860	1,559	594	677	18,153	23,843
	Kidney	1,506	479	1,633	3,023	10,938	17,579
	Liver	359	248	5,649	6,018	10,402	22,676
	Lung	2,509	479	1,290	1,719	11,043	17,040
	Thymus	2,272	869	1,607	1,827	13,407	19,982
	Whole brain	1,387	736	1,493	2,611	12,701	18,928

Table 12: Distribution of weighted interaction types in human and mouse sample-specific networks. In both mouse and human samples, by far the most edges are classified as predicted interactions of the lowest weight P. In all human samples, the next most abundant are interactions of weight A, which is the highest. Interactions of intermediate weight B are the least abundant. Conversely, in mouse samples weight B interactions are represented more prominently. In all but one sample they are more abundant than weight A interactions, and their counts are very high in the liver-specific network.

We see in Table 12 that the weighted interaction types follow a distinct distribution in human and mouse sample-specific networks. Weight P is the most frequent by far in both, but in human

samples, weight B is the least abundant and counts of weight A are intermediate. This means that fewer interactions are regulated by unexpressed TFs than by expressed ones in these networks. For mouse samples, this relationship is reverted. Only in the network specific for mouse heart, there are more weight A interactions than weight B interactions. What is more, the mouse liver-specific network contains an exceedingly high number of both + and -B interactions compared to other mouse samples. This means that a very large proportion of interactions has unexpressed regulators.

The comparison in Table 11 suggests that the mouse heart- and liver-specific networks are different from all other mouse networks. The aberrations in the liver network can be explained when different inputs to its generation are examined: the low numbers of expressed TFs (Table 7), overall detected DNaseI footprints (Table 10) and footprinted predicted binding motifs overlapped (Table 9). Moreover, the small number of occupied / footprinted binding motifs could actually have been caused by the low number of expressed TFs, but also other reasons are possible. For example, the cause could lie on another level of transcriptional regulation, *i.e.* an interference with a chromatin remodeling pathway, leading to a smaller proportion of active chromatin and a decreased accessibility to DNaseI.

When DNaseI footprints for the mouse heart sample are examined, another deviation from all other mouse samples is visible. The high overall detection count (Table 10) and the large number of footprinted binding motifs (Table 9) possibly caused the low number of weight B interactions in this case.

When examining the proportion of activating and inhibiting interaction types within one weight group, +A interactions are always more abundant than -A. For weight B, a difference between human and mouse networks arises: While +B interactions occur more frequently than -B for all but one human cell type-specific network, in mouse tissue-specific networks this relationship is inverse. This difference is interesting because it could give rise to structural dissimilarity between mouse and human active sub-networks (or tissue- and cell type active sub-networks respectively) which are derived from the sample-specific networks presented here.

Another interesting result in Table 11 is the complete absence of -B interactions in some human networks. Following the definition of B-, this means that in these samples all expressed TFs that were targeted by unexpressed TFs had footprinted binding motifs in their regulatory regions occupied by other TFs. Accordingly, all interactions with unexpressed regulators and expressed targets were thus classified as interactions of weight P. When we cross-examine how many predicted binding motifs were in total overlapped by DNaseI footprints in the samples where -B interactions are missing (Table 9), we would expect high overlap counts, but this is not true for all samples, e.g. Monocd14.

When the total number of interactions in sample-specific networks is compared with the number of predicted edges in the reference networks (58,454 for HRNW and 49,202 for MRNW, Table 3), a large difference becomes apparent. The edges constituting this difference are interactions with expressed regulators that have not been confirmed through DNaseI footprints on predicted binding motifs. For both human and mouse samples, only around 50% of the edges in the reference network form the specific networks. What is more, since interactions of weight P are not considered for the generation of active sub-networks, only 10-25% of the edges that

were initially predicted are used in the next step. This considerable reduction shows that false-positive binding motif predictions can be excluded effectively through the integration of experimental data. It is possible however, that not all of the excluded interactions are false-positively predicted, some of them could be functional in other cell types or tissues.

3.4.2 Comparison of Hsmm cell type-specific network with muscle regulatory network

As a validation for the generated sample-specific networks, we examined the network specific to human skeletal muscle myoblasts (Hsmm) in greater detail. For this purpose, we used the muscle regulatory network presented in [25] for comparison (Figure 8). In this reference network, four muscle-specific TFs, MEF2A, MYF6, MYOG and MYOD interact with each other. All four were present and expressed in the Hsmm-specific network, but MYOD was termed MYOD1 (an official synonym according to UniProt KB⁶²). Yet none of the eight muscle-specific interactions in the Figure 8 were present in our network.

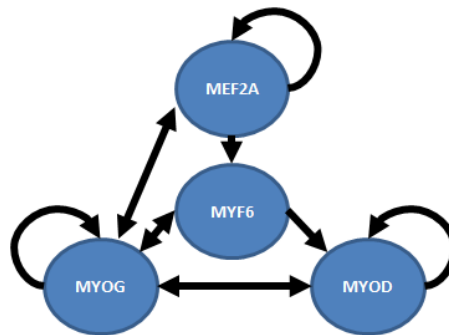


Figure 8: Muscle regulatory network redrawn from [7]. All four muscle-specific TFs in this network were present and expressed in the human skeletal muscle myoblast (Hsmm) -specific network, but no interactions were recapitulated.

When the interactions of each of these TF were investigated separately, it became apparent that MYF6 had no interactions in which it was the regulator in the Hsmm-specific network. We found this true also when we stepped back one level and examined the predicted interactions in HRNW. MYF6 was only part of interactions as a target, never as a regulator. This observation suggested that the TFBS model used to predict MYF6 binding sites in regulatory regions had a very low binding affinity to a diverse set of nucleotide sequences, which made it peculiar.

Also for the other muscle-specific TFs, MEF2A, MYOG and MYOD1, it was possible that the properties of their TFBS models were the cause of the missing interactions. Only two auto-regulatory feedback loops (for MEF2A and MYOD1) that are present in the reference muscle regulatory network were also predicted in HRNW, indicating that the reason for the absent interactions lied within the prediction of binding motifs.

We also investigated if including exons in regulatory regions would lead to the detection of the missing muscle-specific binding motifs / interactions. Thereby, we did find the interaction between MYOG and MEF2A, but this approach also introduced a spurious interaction between MYOD1 and MEF2A which is not present in the muscle regulatory network (Figure 8).

To further investigate the cell type- and tissue-specificity of the networks we generate, we performed functional enrichment analysis by Gene Ontology and examined the similarity of networks for related cell-types using different methods. These analyses have been carried out on the level of active sub-networks however, and the results can be found in the corresponding

chapter (3.5.5).

3.5 Active sub-networks

The aim of this step was to reduce sample-specific networks to subnets that cover all TFs / nodes and interactions that are dynamically connected with active / expressed TFs. Here, we provide standard network measures for the obtained sub-networks and present the distribution of interaction types inside them.

In subsequent sections we present results of the analyses carried out in active sub-networks. We first document enrichment of network motifs and enrichment patterns across samples. Next, human active sub-networks are compared in both one to one and one to all manner, to explore their diversity and to determine if shared interactions coincide with relatedness among cell-types. Finally, we address functional enrichment analyses in which we asked whether GO-terms related to the biological functions of specific samples were enriched in active subnets, and whether a functional enrichment of such GO-terms could be seen on enriched network motifs.

3.5.1 Characterization of active sub-networks using standard network measures

	A549	Cd20	Hepg2	Hsmm	Huvec	K562	Lhcnm2	Lhcnm2 diff	Monocd14	Nhlf	Sknshra
Nodes	398	401	401	395	395	395	394	401	388	393	403
Edges	7.589	7.326	8.563	5.646	6.129	8.905	5.867	5.855	5.539	5.791	7.263
Avg. degree	38	37	43	29	31	45	30	29	29	29	36
NW diameter	6	6	5	7	5	5	7	8	5	9	7
Graph density	0.048	0.046	0.053	0.036	0.039	0.057	0.038	0.037	0.037	0.038	0.045
Modularity	-0.003	-0.002	-0.003	0.28	-0.006	-0.004	0.246	0.151	-0.003	0.295	-0.003
Avg. path length	2,4	2,7	2,3	3,2	2,5	2,1	3,0	3,1	2,5	3,1	2,7

Table 13: Standard network measures for human active sub-networks. In the points of nodes and edges, average degree, network diameter, graph density and average path length, all networks exhibit similar numbers. Examining modularity however, the networks for human skeletal muscle myoblasts (Hsmm), the immortalized cell line of skeletal myoblasts (Lhcnm2), the differentiated skeletal myoblast cell line (Lhcnm2diff) and lung fibroblasts (Nhlf) show distinctly higher values than the other networks.

Across almost all network measures, the generated active sub-networks for human cell-types appear similar (Table 13). The number of nodes in the networks is comparable to the node number in HRNW which means that besides active / expressed TFs, also a considerable proportion of inactive / unexpressed TFs whose interactions were selected as dynamically relevant are present in the networks (for the number of expressed TFs per sample see Table 7).

Remarkably, the networks for human cell-types differ in their modularity, despite similarities in other measures. Even though modularity is not considered meaningful until it reaches at least 0.4, networks divide into two distinct groups when modularity is examined. For one group, modularity is around zero, while for the other it reaches values between 0.15 and 0.3. We refer to this difference again in the section where we discuss the distribution of interaction types (chapter 2.7.2).

	Heart	Kidney	Liver	Lung	Thymus	Whole brain
Nodes	384	376	371	379	381	383
Edges	5.649	6.296	11.609	5.794	6.392	6.086
Avg. degree	29	33	63	31	34	32
NW diameter	6	8	9	7	8	7
Graph density	0.038	0.045	0.085	0.040	0.044	0.042
Modularity	0.214	0.263	0.049	0.265	0.333	0.264
Avg. path length	2,8	2,8	2,7	2,9	2,8	2,9

Table 14: Standard network measures for mouse active sub-networks. Unlike the networks for human cell types, almost all mouse active subnets all exhibit a similar modularity. The subnet constructed for liver stands out from the others with a markedly higher edge number and a higher average degree as a result of this. Modularity for the liver active sub-network is lower than for all other mouse networks.

For mouse active sub-networks, standard network measures, including modularity, are similar across all but one sample, the liver (Table 14). However, the differences between liver and the other samples can be attributed to the distribution of interaction types in the liver-specific network which diverges from the distribution in other mouse tissue-specific networks (see Table 12).

3.5.2 Distribution of weighted interaction types

We examined the distribution of interaction types in active sub-networks, under consideration of their respective selection criteria. The aim of selecting interactions from the sample-specific networks was to obtain dynamically relevant interactions of active TFs. We defined “dynamically relevant” paths of interactions that involved active TFs, both directly and indirectly over edges with inactive TFs. As explained in Methods, in chapter 2.2.5 and Figure 4, we chose interactions that had active TFs as targets or regulators, which applies to +A, -A and -B edges. For +B edges, which connect two unexpressed TFs, we considered interactions dynamically relevant in two cases: 1) when they connected two unexpressed TFs that had other interactions with active TFs and 2) when they were part of chains of interactions which started and ended on active TFs. Hence, the counts of interaction types differ between active sub-networks and sample-specific networks only for +B edges. Of course, another important difference between active and sample-specific networks is the exclusion of weight P edges. They were not considered for the selection of dynamically relevant interactions since the experimental evidence that supports them is spurious.

		Distribution of weighted interaction types				Total number of interactions in NW
		+A	-A	+B	-B	
Human Cell types	A549	4.526	2.735	328 (446)	0	7.261
	Cd20	4.029	3.255	0 (83)	42	7.326
	Hepg2	4.903	3.660	0 (50)	0	8.563
	Hsmm	2.966	1.201	1.377 (1.670)	142	4.309
	Huvec	4.354	1.684	91 (227)	0	6.038
	K562	5.485	3.420	0 (252)	0	8.905
	Lhcnm2	2.823	1.658	1.057 (1.504)	329	4.810
	Lhcnm2 diff	3.467	1.606	508 (634)	274	5.347
	Monocd14	3.410	2.127	2 (468)	0	5.537
	Nhlf	3.195	1.079	1.308 (1.659)	209	4.483
	Sknshra	4.138	2.821	78 (133)	226	7.185
Mouse tissues	Heart	2.860	1.559	553 (594)	677	5.096
	Kidney	1.506	479	1.288 (1.633)	3.023	5.008
	Liver	359	248	4.984 (5.649)	6.018	6.625
	Lung	2.509	479	1.087 (1.290)	1.719	4.707
	Thymus	2.272	869	1.424 (1.607)	1.827	4.968
	Whole brain	1.387	736	1.352 (1.493)	2.611	4.734

Table 15: Distribution of weighted interaction types in human cell type and mouse tissue active sub-networks. The numbers of +A, -A and -B interactions are the same as in sample-specific networks, for +B interactions the counts from sample-specific networks are shown in brackets.

For human cell type active sub-networks, the largest fraction of interactions has weight A, which means most edges have regulators that are expressed TFs (Tab. 15). The proportion of weight B interactions, which are regulated by unexpressed TFs, differs strongly between human samples. While some networks have few or no edges of this weight, in other samples this group makes up a considerable proportion of the network. Interestingly, there is a correlation between a large fraction of weight B edges and a degree of modularity (Tab. 13). This correlation applies to all human active sub-networks with high numbers of both +B and -B edges, namely Hsmm, Lhcnm2, Lhcnm2diff and Nhlf. A likely reason for this could be the high number of connections between unexpressed TFs on the one side (represented by +B edges) and expressed TFs on the other (represented by +A edges), possibly creating a module of unexpressed TFs and one of expressed TFs.

The same explanation could apply to modularity in mouse tissue active sub-networks, where +A and +B interaction types are abundant across almost all samples. The comparably low modularity of the liver sub-network is paired with the low number of +A interactions in this sample.

In general, we did not expect marked modularity in the active sub-networks, because they constitute specific modules themselves, and our results are in line with this hypothesis.

The method applied to assess the modularity of networks focuses on finding so-called communities. Communities are groups of nodes in which nodes are more connected with each other than with nodes outside the group. Note, that communities with many nodes are not the only level at which modularity can be implemented in biological networks. Network motifs, *i.e.* small, recurring sub-graphs of connected nodes, have been implicated in functional network dynamics and can also be seen as having the hallmarks of modules, namely the connectedness and functionality. In the next section, we therefore move on to elaborate the results of network motif detection in active sub-networks.

3.5.3 Network motifs in active sub-networks

In the following, we report significantly enriched network motifs in active sub-networks of human cell types and mouse tissues. The criteria for significant enrichment were $p\text{-value} < 10^{-2}$ and $z\text{-score} > 2$, and networks were compared to 1000 randomized versions. It should be noted that a significant enrichment of network motifs does not imply functionality. Motifs can be enriched also because they arise easily and have no detrimental effect.

		Network motif ID												
														

Table 16: Significantly enriched network motifs in human cell type- and mouse tissue active sub-networks. The network motif with the highest z-score in each sample is marked with an asterix. While a rather clear pattern of enrichment arises for mouse tissues (motif IDs 38, 46, 108, 110, 238), in human samples the picture is more diverse. For each sample group, one sample shows a distinctly different enrichment pattern: Sknshra (Neuroblastoma cell line, differentiated with retinoic acid) in human cell types and heart in mouse tissues.

The above overview shows that some sub-graphs are enriched in multiple human samples: motifs 38, 46, 102, 108, 110 and 238 (Table 16). For mouse tissue active subnets, a distinct enrichment pattern consists of motifs 38, 46, 108, 110 and 238. These motifs are enriched also in

human cell type networks, but less consistently across cell types and the pattern is met entirely only in lung fibroblasts (Nhlf).

Two of the samples exhibit very different network motifs than all others: Sknshra (Neuroblastoma cell line, differentiated with retinoic acid) from the group of cell-type samples and heart from the group of tissue samples. The distribution of interaction types in Sknshra and heart is slightly different from the general trend in their sample groups (Table 15), but since the group is small, it is difficult to state with certainty if this aberration is significant. The mouse heart sample has already diverged from other samples in previous analyses but for Sknshra, deviations have not been observed so far. , The deviation in Sknshra network motif enrichment pattern could be indicative of deregulation, as Sknshra is a cancer sample.

If cancer samples (A549, Hepg2, K562, Sknshra) are not considered, the motifs enriched most frequently in human and mouse samples are 46, 238 and 110, followed by 38 and 108. Interestingly, these five motifs are also enriched in mouse samples. When we examine motifs with strongest enrichment (highest z-score) across all samples, the top three motifs are 46, 108 and 38 (excluding cancer samples again).

In summary, this analysis provides a small number of candidate network motifs that could be central in the regulation of traits. Strikingly, a combination of enriched motifs has been observed in the active sub-networks of tissues and to a lesser extent, in the networks of cell-types. This combination is consistent with that found in studies of Neph et al on network motifs in human cell type regulatory networks.²⁵

3.5.4 Comparison of active sub-networks

We performed different types of comparisons to assess how diverse the generated active-sub-networks were. One of the aims was to ascertain whether the consistency of enriched motifs in mouse samples arose because the networks were generally similar. Such a finding would explain the pattern, but it would also mean that the networks are in fact not tissue-specific. Our previous analyses have shown distinct distributions of interaction types, but we have so far not assessed what proportion of edges is shared. We expected networks of related samples, e.g. different muscle cell types, to be more similar and to share more interactions than unrelated samples, due to the underlying developmental regulation.

Networks were compared in pair wise manner by computing the Jaccard index of two networks on the basis of common edges. Moreover, we also carried out one to all comparisons, clustering networks based on their normalized network degree and performing multidimensional scaling based on common edges.

3.5.4.1 Jaccard index of active sub-networks

The Jaccard index presented here measures similarity between pairs of networks, based on the presence of common interactions.

17a	Human cell type active sub-networks										
	A549	Cd20	Hepg2	Hsmm	Huvec	K562	Lhcnm2	Lhcnm2 diff	Monocd14	Nhlf	Sknskra
A549	1,000	0,207	0,213	0,163	0,206	0,222	0,161	0,185	0,188	0,158	0,213
Cd20	-	1,000	0,202	0,150	0,203	0,217	0,155	0,190	0,220	0,147	0,202
Hepg2	-	-	1,000	0,162	0,222	0,253	0,150	0,169	0,191	0,158	0,256
Hsmm	-	-	-	1,000	0,202	0,168	0,169	0,171	0,165	0,231	0,157
Huvec	-	-	-	-	1,000	0,238	0,166	0,190	0,214	0,204	0,220
K562	-	-	-	-	-	1,000	0,160	0,180	0,209	0,159	0,226
Lhcnm2	-	-	-	-	-	-	1,000	0,192	0,155	0,155	0,152
Lhcnm2diff	-	-	-	-	-	-	-	1,000	0,191	0,162	0,168
Monocd14	-	-	-	-	-	-	-	-	1,000	0,174	0,189
Nhlf	-	-	-	-	-	-	-	-	-	1,000	0,158
Sknskra	-	-	-	-	-	-	-	-	-	-	1,000

17b	Mouse tissue active sub-networks					
	Heart	Kidney	Liver	Lung	Thymus	Whole brain
Heart	1,000	0,154	0,099	0,216	0,179	0,142
Kidney	-	1,000	0,245	0,213	0,164	0,150
Liver	-	-	1,000	0,152	0,147	0,142
Lung	-	-	-	1,000	0,225	0,178
Thymus	-	-	-	-	1,000	0,162
Whole brain	-	-	-	-	-	1,000

Table 17a,b: Color-coded tables with Jaccard indices of common edges (16a: human cell type active subnetworks, 16b: mouse tissue active sub-networks). Jaccard indices were computed based on shared edges / interactions. Red color represents high similarity (with 1 meaning identity) and yellow dissimilarity (with 0 meaning complete disparity). It is evident that the generated networks for human cell types or mouse tissues are highly dissimilar. Black borders were put around cells in which an increased similarity would be expected due to relatedness.

From the one to one comparisons it is obvious that the generated networks are very distinct in their interaction repertoire (Table 17a,b). The Jaccard index ranges from 0 (dissimilarity) to 1 (identity). Here, Jaccard indices are low in all network pairs, as signified by the orange color. This result supports the cell type and tissue-specificity and excludes that the pattern of network motif enrichment presented in 2.7.4 is an artifact caused by highly similar networks.

However, similarities between related cell types (e.g. Hsmm – human skeletal muscle myoblasts, Lhcnm2 – skeletal myoblast cell line, Lhcnm2diff – differentiated skeletal myoblast cell line) are not visible in this comparison. The highest Jaccard indices are observed for cancer samples: Hepg2 and K562, and Hepg2 and Sknskra, but the values differ only very subtly from the rest.

3.5.4.2 Normalized network degree clustering

This analysis visualizes the similarity of networks on a structural level. Networks are clustered based on how similar the degree of each of their nodes is. The use of this clustering technique for network comparison assumes that regulators have similar numbers of interactions in functionally related cell types, which is rather uncertain.

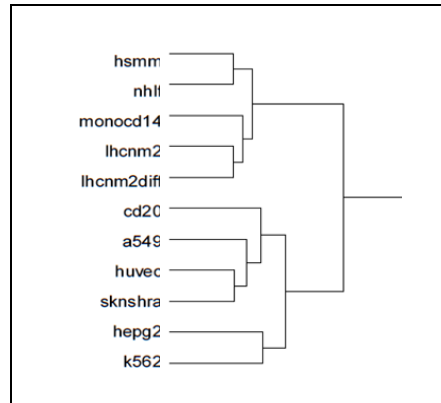


Figure 9: Normalized network degree clustering of human cell type active sub-networks. The clustering divides samples into two distinct groups that only partly correlate with functional relatedness. Skeletal myoblast Lhcnm2 is clustered together with Lhcnm2diff which is the same cell type that underwent differentiation. The third muscle cell type Hsmm is however clustered farther away from them.

While few related cell types cluster together in this analysis, many other relationships are not displayed (Figure 1). For example, it is reasonable that the three muscle cell types Hsmm (human skeletal muscle myoblasts), Lhcnm2 (skeletal myoblast cell line) and Lhcnm2diff (differentiated Lhcnm2) are present in the same group. The closeness between cancer cell types (A549 – epithelial cells from lung carcinoma, Hepg2 – hepatocellular carcinoma, K562 – cell line from myelogenous leukemia, Sknshra – neuroblastoma cell line) is interesting because they all represent deregulation, even though their origins are diverse. However, the blood cell types Cd20 (B-cells) and Monocd14 (monocytes) cluster very far apart. Additionally, the proximity the two Lhcnm2 muscle cell types (skeletal muscle myoblast cell line, undifferentiated and differentiated) to Monocd14 (monocytes) is unexpected, especially considering that another muscle cell type, Hsmm (human skeletal muscle myoblasts), is located farther away.

Next, we apply yet another method of network comparison to test the robustness of the findings.

3.5.4.3 Multidimensional scaling

Turning the focus from the overall structure to the level of interactions, we performed metric multidimensional scaling (MDS) of the human cell type active subnets based on the presence / absence profiles of all possible edges in the human samples. The result of the scaling is presented in a plot in which networks with similar interaction profiles lie close and networks with little overlap lie distant. We carried out this analysis in two- and three-dimensional space to obtain the best resolution.

We focus on human cell types here, as no considerable similarities were expected between mouse samples.

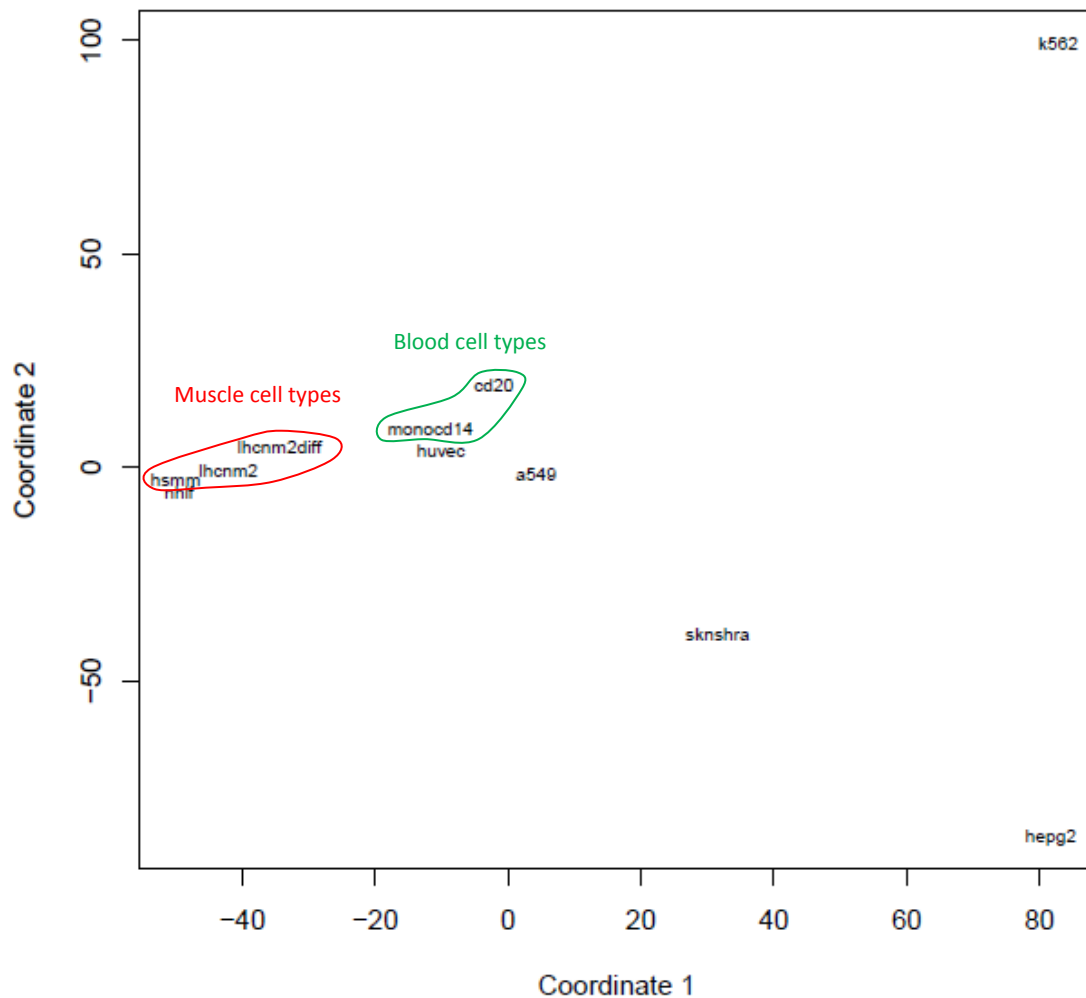


Figure 10: Two-dimensional scaling of human cell type active sub-networks. Muscle cell types are within close proximity (circled red), indicating that the corresponding networks are more similar to each other than to the rest of the analyzed nets. Similarly, Cd20+ B-cells and CD14+ monocytes, both from the hematopoietic cell lineage are located near each other (circled green). Cancer cell types Sknshra (Neuroblastoma), K562 (myelogenous leukemia) and Hepg2 (hepatocellular carcinoma) are located far from the other samples, suggesting strong deviations.

The result of the two-dimensional scaling correlates with cell type function to a large extent (Figure 10). On the left, the three muscle cell types Hsmm, Lhcnm2 and Lhcnm2diff (human skeletal muscle myoblasts, skeletal myoblast cell line, differentiated skeletal myoblast cell line) are located within close proximity, while blood cell types Monocd14 and Cd20 (Monocytes and B-cells) form a pair closer to the middle. The large distance between healthy cell types and some of the cancer cell types indicates that these two groups are very different at the level of interactions.

However, there are also some cases in which proximity does not match functional and developmental relatedness. For instance, Hsmm (Human skeletal muscle myoblasts) is in very close proximity to Nhlf (lung fibroblasts) and Monocd14 (monocytes) is close to Huvec (umbilical vein endothelial cells), even though there is no obvious functional similarity between them. Located nearby Monocd14 and Huvec is also the fourth cancer sample, A549 (epithelial cells

from lung carcinoma), demonstrating that it shares more interactions with the networks of healthy samples than with all other cancer cell types that were examined.

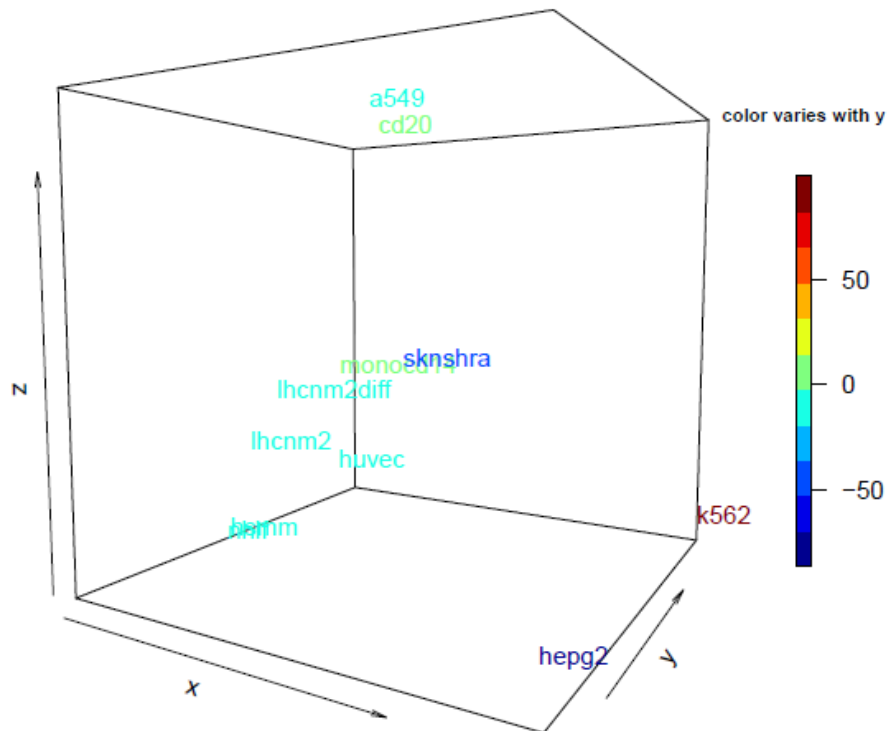


Figure 11: Three-dimensional scaling of human cell type active sub-networks. The color-scale represents depth along the y-axis. Similar to the 2D plot, functionally similar and related cell types are located close to each other. Biologically meaningful correlations persist (Lhcnm2, Skeletal myoblast cell line and Lhcnm2diff, differentiated skeletal myoblast cell line; Cd20, B cells and Monocd14, Monocytes), but also some that appear less plausible are still visible (e.g. Hsmm, human skeletal muscle myoblasts and Nhlf, lung fibroblasts).

When the 3D-scaling plot of the human cell type active subnets is examined, the distances and proximities between certain samples become even more apparent (Figure 11). Cancer cell types are now exclusively located at the borders of the plot (Hepg2 – liver carcinoma, K562 – myelogenous leukemia, Sknshra – neuroblastoma cell line), but there is still proximity between the cancer sample A549 (epithelium from lung carcinoma) and a healthy sample, Cd20 (B-cells). This is a rather unexpected result because the two cell types are unrelated, but still more similar in their regulatory repertoire than to other samples.

Hsmm, Lhcnm2 and Lhcnm2diff (human skeletal muscle myoblasts, skeletal myoblast cell line, differentiated skeletal myoblast cell line) still co-locate, but they are in the vicinity of less similar cell types like Nhlf (lung fibroblasts) and Huvec (umbilical vein endothelial cells). Remarkably, there is almost a complete overlap between Hsmm and Nhlf despite their different function. Meanwhile, the relatedness of the blood cell types Cd20 and Monocd14 (Monocytes) is highlighted by their colouring (green).

In summary, separation and clustering of samples in the MDS plots recapitulate unrelatedness and relatedness between the cell types rather accurately. This result confirms our hypothesis that common interactions in regulatory networks are an indicator for relatedness among samples.

3.5.5 Analysis of functional specificity

When the 3D-scaling plot of the human cell type active subnets is examined, the distances and proximities between certain samples become even more apparent (Figure 11). Cancer cell types are now exclusively located at the borders of the plot, but there is still proximity between the cancer sample A549 (epithelium from lung carcinoma) and a healthy sample, Cd20 (B-cells). Hsmm, Lhcnm2 and Lhcnm2diff (human skeletal muscle myoblasts, skeletal myoblast cell line, differentiated skeletal myoblast cell line) still co-locate, but they are in the vicinity of less similar cell types like Nhlf (lung fibroblasts) and Huvec (umbilical vein endothelial cells). Remarkably, there is almost a complete overlap between Hsmm and Nhlf despite their different function. Meanwhile, the relatedness of the blood cell types Cd20 and Monocd14 (Monocytes) is highlighted by their colouring (green).

In summary, separation and clustering of samples in the MDS plots recapitulate unrelatedness and relatedness between the cell types rather accurately. This result confirms our hypothesis that common interactions in regulatory networks are an indicator for relatedness among samples.

3.5.4.4 Functional enrichment in active sub-networks

		Number of genes analyzed	Enrich. score	Biological Process GO terms in annotation cluster	Rank of corresponding annotation cluster
Human cell types	A549 (epithelium from lung carcinoma)	103	1.66	Gland development, tube development, tissue morphogenesis, tube morphogenesis, epithelium development (B 0.068)	7
	Cd20** (B-cells)	57**	0.58**	Positive regulation of cytokine biosynthetic process, regulation of cytokine biosynthetic process, regulation of cytokine production **	9**
	Hepg2 (liver carcinoma)	89	2.22	endocrine system development (B 0.11), endocrine pancreas development (B 0.14), pancreas development (B 0.17), liver development (B 0.69)	3
	Hsmm (skeletal muscle myoblasts)	80	1.59	muscle organ development, striated muscle tissue development (B 0.29), muscle tissue development (B 0.33), muscle organ development (B 0.68), muscle tissue development (B 0.68)	4
	Huvec (umbilical vein endothelium)	91	1.26	blood vessel development (B 0.54), vascular development (B 0.56), blood vessel morphogenesis (B 0.76)	6
	K562 (myelogenous leukemia)	52	1.52	regulation of myeloid cell differentiation (B 0.18), regulation of erythrocyte differentiation (B 0.77)	2
	Lhcnm2 (skeletal myoblasts)	74	1.12	muscle organ development (B 0.27), striated muscle tissue development (B 0.64), muscle tissue development (B 0.69)	4
	Lhcnm2diff (diff. skeletal myoblasts)	79	1.42	muscle organ development (B 0.12), striated muscle tissue development (B 0.64), skeletal muscle organ development (B 0.65), skeletal muscle tissue development (B 0.65), muscle tissue development (B 0.68)	6
	Monocd14 (monocytes)	57	1.03	regulation of myeloid cell differentiation (B 0.21), regulation of myeloid leukocyte differentiation (B 0.52)	8
	Nhlf (lung fibroblasts)	98	2.20	tissue morphogenesis, embryonic morphogenesis, tube development, branching morphogenesis of a tube, morphogenesis of a branching structure, gland development (B 0.11), epithelium development (B 0.11), lung development (B 0.11), respiratory tube development (B 0.11) etc.	4
	Sknsbra (neuroblastoma)	105**	0.97**	gland development (B 0.07), forebrain development (B 0.23), pituitary gland development (B 0.36), diencephalon development (B 0.52), regulation of ossification (B 0.56)**	11**

Table 18: Enrichment of annotation clusters specific to examined human cell-types. The analysis was restricted to Biological Process GO terms since they best represent sample function. For all samples except Cd20 (B cells), an annotation cluster related to sample-specific functions was found enriched. As for GO terms, we report those matching p-value < 0.1, fold change > 1.5 and Benjamini correction ≤ 0.05 , unless specified otherwise (e.g. B 0.068 for Benjamini correction).

**not significantly enriched

The enrichment score (significant at ≥ 1) provided in Table 18 is the geometric mean of all p-values in the annotation cluster. P-values were calculated using a modified Fisher's exact test and significance was established at a cutoff of p-value < 0.1. When reporting GO terms, we also considered fold-change > 1.5 and Benjamini correction for multiple testing ≤ 0.05 . For the latter however, we also reported GO terms exceeding this cutoff, because Benjamini correction is a conservative approach and relying too strictly on it could reduce sensitivity.⁵⁶

As visible in Table 18, annotation clusters that list sample-specific biological processes were found enriched for “specifically” expressed TFs in all but two human cell types, namely Cd20 (B cells) and Sknshra (neuroblastoma cell line).

It should be noted that for Cd20 and two other samples, a rather small set of genes reduced the power of the analysis. Highest enrichment scores were found for Hepg2 and Nhlh. Interestingly, Hepg2 still shows an enrichment of liver-specific GO terms even though the sample represents a state of disease where interferences with the normal regulation of gene expression could be expected.

Lower enrichment scores co-occur with a smaller number of TFs used for the analysis, indicating that the data becomes more difficult to interpret when fewer input genes are used. An exception from this is Sknshra, where a low enrichment score was calculated although the largest gene list was used as an input. This may be attributed to a cancerous deregulation of gene expression in the sample.

		Number of genes analyzed	Enrich. score	Biological Process GO terms in annotation cluster	Rank of corresponding annotation cluster
Mouse tissues	Heart	75	1.68	regulation of cardiac muscle growth (B 0.33), regulation of cardiac muscle tissue development (B 0.33), regulation of cardiac muscle cell proliferation (B 0.33), regulation of organ growth (B 0.35)	2
	Kidney	108	2.00	urogenital system development, kidney development (B 0.12), metanephros development (B 0.58)	4
	Liver	52	2.12	endocrine pancreas development, pancreas development (B 0.22), liver development (B 0.34), endocrine system development (B 0.52)	2
	Lung	121	2.21	tube development (B 0.25), respiratory system development (B 0.3), lung development (B 0.34), respiratory tube development (B 0.34)	2
	Thymus	96	1.61	leukocyte differentiation, hemopoiesis (B 0.25), hemopoietic of lymphoid organ development (B 0.27), immune system development (B 0.33), cell activation (B 0.56), lymphocyte differentiation (B 0.57) etc.	3
	Whole brain	104	3.81	forebrain development, endocrine system development, pituitary gland development, diencephalon development (B 0.052), gland development (B 0.16)	4

Table 19: Enrichment of annotation clusters specific to examined mouse tissues. As with human samples, the analysis was restricted to Biological Process GO terms. For all mouse tissues, a corresponding annotation cluster covering sample-specific functions was found enriched. We report those GO terms matching p-value < 0.1, fold change > 1.5 and Benjamini corrections ≤ 0.05, unless specified otherwise (e.g. B 0.33 for Benjamini correction).

Compared with the results for human a trend towards higher enrichment scores is visible for mouse samples (Table 19). This picture persists when the ranks of sample-specific annotation clusters are examined. While ranks are rather diverse for human cell types, for mouse tissues the corresponding annotation clusters are within the Top 4 for all samples.

As evident from the ranks of the specific annotation clusters, for all samples there were also other GO term clusters that were more enriched. In many samples, especially those with high ranks, these higher-ranking clusters referred to transcriptional regulation, a function that

corresponded to all genes entering the analysis because they were TFs. Conversely, Lower-ranking annotation clusters, such as the one for Sknshra, were preceded by rather non-specific functions (embryonic development, muscle development, reproductive organ development).

3.5.4.5 Functional enrichment in significant network motifs

Human sample	Network motif ID	Number of genes analyzed	Enrich. Score	Biological Process GO terms in annotation cluster	Rank of corresponding annotation cluster
A549	38 	236	1.75	tube development (B 0.12), respiratory system development (B 0.46), respiratory tube development (B 0.52), lung development (B 0.52)	13
	108 	168	1.49	tube development (B 0.37), lung development (B 0.47), respiratory tube development (B 0.47), respiratory system development (B 0.58)	11
Hepg2	46 	202	1.40	pancreas development (B 0.54), endocrine system development (B 0.56), endocrine pancreas development (B 0.63) - cluster chosen based on liver marker genes (HNF1A, HNF1B, HNF6) ^{63,64}	10
Huvec	46 	209	1.75	angiogenesis (B 0.18), blood vessel morphogenesis (B 0.41), blood vessel development (B 0.41), vasculature development (B 0.46)	10
K562	46 	197	2.59	regulation of myeloid cell differentiation, positive regulation of myeloid cell differentiation, positive regulation of cell differentiation (B 0.22), etc.	4
	238 	72	1.38	regulation of erythrocyte differentiation (B 0.27), regulation of myeloid cell differentiation (B 0.34), positive regulation of erythrocyte differentiation (B 0.42), etc.	3
Monocd14	38 	201	1.57	positive regulation of myeloid cell differentiation (B 0.15), regulation of myeloid leukocyte differentiation (B 0.47), positive regulation of myeloid leukocyte differentiation (B 0.77)	11
	46 	175	2.62	lymphocyte differentiation, leukocyte differentiation, lymphocyte activation, cell activation, leukocyte activation, hemopoiesis (B 0.08), myeloid cell differentiation (B 0.32), etc.	6
Nhlf	38 	231	1.59	tube development (B 0.22), respiratory tube development (B 0.46), lung development (B 0.46), respiratory system development (B 0.6)	12
	46 	194	1.74	tube development (B 0.11), respiratory tube development (B 0.44), lung development (B 0.44), respiratory system development (B 0.57)	13

Table 20: Enrichment of cell type function related annotation clusters on significant network motifs (human samples). For 6 of 12 human cell types, we find TFs related to cell type function enriched on certain significant network motifs. Notably, motifs 38 (feed-forward loop) and 46 are enriched for marker TFs in diverse cell-types. We report GO terms that match following criteria: p-value < 0.1, fold change > 1.5 and Benjamini correction ≤ 0.05 , unless specified otherwise (e.g. B 0.12 for Benjamini correction).

This analysis reveals that in various human samples, the marker TFs which characterize the development, maintenance and identity of cell types are enriched on significant network motifs (Table 20). In five of the cell types examined in this study, this enrichment was not found (Cd20 –

B-cells, Hsmm – Human skeletal muscle myoblast, Lhcnm2 – Skeletal muscle myoblast cell line, Lhcnm2diff – differentiated Lhcnm2, Sknshra – Neuroblastoma cell line), but in the other six, especially motifs 38 and 46 were enriched for marker TFs, indicating that their dynamics could play a role in the cell types developmental regulation. For motif 38, the feed-forward loop, it has already been shown that it can implement functional dynamics. The dynamics of motif 46 however remain to be investigated.

Mouse sample	Network motif ID	Number of genes analyzed	Enrich. Score	Biological Process GO terms in annotation cluster	Rank of corresponding annotation cluster
Heart	6 	218	1.44	cardiac muscle cell differentiation (B 0.18), cardiac cell differentiation (B 0.24), positive regulation of cardioblast regulation (B 0.41), etc.	7
	12 	221	1.45	cardiac muscle cell differentiation (B 0.2), cardiac cell differentiation (B 0.26), positive regulation of cardioblast regulation (B 0.43), etc.	7
	36 	221	1.45	cardiac muscle cell differentiation (B 0.2), cardiac cell differentiation (B 0.26), positive regulation of cardioblast regulation (B 0.43), etc.	7
Kidney	38 	228	1.31	urogenital system development (B 0.16), kidney development (B 0.32), ureteric bud development (B 0.57), metanephros development (B 0.59) etc.	14
	46 	135	1.01	ureteric bud development (B 0.87), urogenital system development (B 0.87), metanephros development (B 0.88), kidney development (B 0.87)	7
Liver	38 	155	1.02	endocrine pancreas development (B 0.74) - cluster chosen based on liver marker genes (HNF1A, HNF1B, HNF6) ^{63,64}	10
Lung	38 	259	2.86	respiratory system development (B 0.09), tube development (B 0.09), lung development (B 0.11), respiratory tube development (B 0.11)	2
	46 	173	3.45	respiratory system development, lung development, respiratory tube development, tube development (B 0.19)	2
	110 	23	1.56	respiratory system development (B 0.12), lung development (B 0.23)	2
Thymus	38 	224	2.25	leukocyte differentiation, lymphocyte differentiation (B 0.06), cell activation (B 0.21), immune system development (B 0.2), hemopoiesis (B 0.2), etc.	2
	46 	150	1.08	leukocyte differentiation (B 0.56), hemopoiesis (B 0.59), lymphocyte differentiation (B 0.6), leukocyte activation (B 0.67), immune system development (B 0.71) etc.	7
Whole brain	38 	217	1.80	forebrain development (B 0.25), pituitary gland development (B 0.31), endocrine system development (B 0.54), diencephalon development (B 0.71), gland development (B 0.82)	5

Table 21: Enrichment of tissue function related annotation clusters on significant network motifs (mouse samples). We find that TFs related to tissue function are enriched on certain significant network motifs in all examined mouse samples. In all but one sample, we find motif 38 (feed-forward loop) to be enriched for such “marker TFs”. In the majority of samples, this is also true for motif 46. This pattern of enrichment resembles the one found in human samples (Table 20). GO terms that match p-value < 0.1, fold change > 1.5 and Benjamini correction ≤ 0.05 are reported, unless specified otherwise (e.g. B 0.12 for Benjamini correction).

For each mouse sample, we find at least one network motif to be enriched for tissue “marker” TFs (Table 21). Annotation clusters related to tissue function were found only marginally enriched for some motifs (kidney motif 46, liver motif 38, thymus motif 46), but in these cases a smaller number of genes was analyzed, possibly lowering the statistical power of the analysis. Conversely, for lung motif 110 an average enrichment score was found even though only 23 genes were analyzed.

While most of the mouse samples follow a pattern of enrichment with motif 38 and motif 46, the heart sample shows a distinct enrichment of network motifs and marker genes inside them. In no other sample of this study, motifs 6, 12 and 36 were significant and enriched for tissue-specific genes. The reason for this deviation could lie in the distribution of interaction types in the heart active sub-network, which is quite different from other mouse tissues (Table 15).

Interestingly, we discovered an enrichment of tissue markers on FFLs (motif 38) in all mouse samples where the FFL was called significant. Together with marker gene enrichments on FFLs in human samples, these findings emphasize that feed-forward loop dynamics could be involved in determining tissue development and identity.

4 DISCUSSION

The following sections are devoted to summarizing our findings regarding trait-specific functional enrichment on network motifs, discussing results related to the regulatory networks we constructed and presenting the importance of our results in the context of other studies. Since a large part of our study was devoted to method development, and results critically depend on the choice of methods, we also discuss the considerations and decisions that underlie our methodology and draw parallels to other studies.

Key achievements of our work comprise:

- pilot study in network reconstruction from multiple complementary data types
- establishment of link between topological motifs, dynamics and trait development
- establishment of functional trait interpretation based on regulatory interaction repertoire
- binning of interaction types and first steps towards parameterization of regulatory networks

Furthermore, our study demonstrates a need for standards in computational algorithms for DNaseI footprinting, improvements in TFBS model prediction and a systematic collection of experimental data in related conditions (e.g. cell lineages, homologous cell types).

4.1 Active sub-networks

In active sub-networks, we integrated all possible influences of and on the active regulators in the network, capturing the union of regulatory effects going out from or imposed on TFs.

4.1.1 Establishing connection between network motifs and functional interpretation of regulatory networks

Within active sub-networks, we studied the enrichment of network motifs, and their importance for the regulation of traits. Our analyses show that certain motifs are frequently enriched in regulatory networks of different human cell types and mouse tissues (Table 16). There is especially strong consistency between tissue samples, of which most exhibit an enrichment of motifs 38, 46, 108, 110 and 238 (Figure 12). In cell type samples, these motifs are also often enriched.

Motif 38 (Figure 12) was termed “feed-forward loop” in the literature and already has a strong reputation as a functional network motif. It has been studied in dynamic network analysis and one of its subtypes, the coherent FFL, has been characterized as a noise filter that passes on impulses to targets only upon prolonged input signals.⁶⁵ While enrichment may indicate function, it should be noted that not all network motifs have been examined for potential functions, and some have been considered to show enrichment only because they arise easily.

Interestingly, feed-forward loops in many samples show an enrichment of developmental regulators (marker genes) specific to the trait under study, e.g. muscle TFs in a muscle cell type (Table 20 and 21).

Network motif 46 is also enriched for marker genes in many samples. This motif can be seen as a precursor or successor of the FFL, since FFLs can be generated from motif 46 by removing only one interaction. Because of sequence-specific binding of TFs, point mutations easily cause a loss of an interaction. Furthermore, motif 46 itself can arise easily. Starting with an interaction in which the regulator has an auto-regulatory ability, a simple duplication of the gene that encodes the regulator would suffice to produce this motif. This indicates that motif 46 could be generated as a non-functional by-product, but an analysis of its potential dynamics would be required to shed light on its functionality. This could be implemented in a model of ordinary differential equations (ODE).

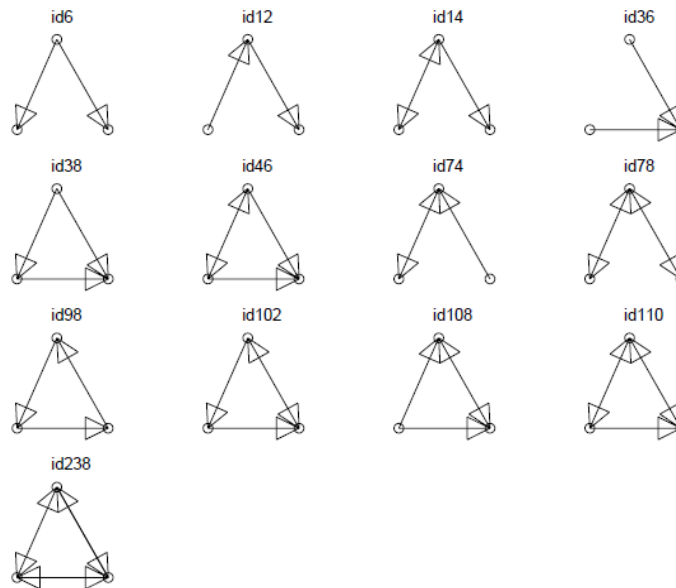


Figure 12: The 13 possible 3-node sub-graphs that were examined for significant enrichment in active sub-networks. The specified IDs correspond to IDs used in Table 15 and in the text.

Another enriched network motif with a confirmed function is motif 108. It has been shown that it can support individualization of two targets regulated by a common signal (S. Widder, unpublished work) which is an interesting ability considering cell lineage decisions and developmental differentiation in general. However, marker gene enrichment on this motif has been found only in one sample, and it was not particularly strong (Table 20).

For network motifs 6, 12, 36, 110 and 238 marker gene enrichments have been found in single samples. Just as for motif 46, their dynamics are unknown so far.

It is noteworthy that in all cancer samples except Sknshra (neuroblastoma cell line), marker gene enrichment was found on network motifs (Table 20). This is an interesting result because cancerous cells represent deregulation, *i.e.* less specific states of cells (perhaps similar to non-differentiated ones, or partially so). Despite their deregulated state, regulators related to their original function are still located within network motifs. This invites speculation that changes in regulation in cancer cells pertain to other aspects of regulation, potentially to peripheral regulatory networks, while the core aspect remains intact. Such interpretation is plausible, as cancer cells are invasive, and thus their regulatory network, even if deregulated, is certainly highly functional.

4.1.2 Establishing specificity and functional relatedness based on network interactions

To show that the constructed active sub-networks are specific with regard to cell type or tissue function, we examined them using a number of different comparative measures (chapter 3 to 3.5.4.4). Analyses were carried out on both edge and node level.

A calculation of Jaccard indices based on common edges showed high diversity among samples in the interaction repertoire (Table 17). This finding confirms that our method is free of biases which would make networks generally similar. However, the comparison of Jaccard indices shows no similarities between networks of functionally related samples, *i.e.* various muscle cell types. We therefore applied further measures that capture the functional aspects of the network, which revealed a clustering of related samples.

With multidimensional scaling (MDS) we analyzed particularly the interaction structure between the TFs and thus used it as a measure to assess functional relatedness between regulatory networks. MDS in both two- and three-dimensional space display a more plausible picture of the relationships between samples (Figure 10 and 11). Developmentally related cell types appear in proximity and cancer samples are located far from healthy ones. Many of cancer samples are also immortalized cell lines which are known to exhibit altered gene expression. We did not find a different overall number of expressed genes in cancer / cell line samples (Table 6), but for many of these samples, the interaction repertoire reflects altered expression and regulation. Interestingly, this is not true for the muscle cell line Lhncm2 (skeletal muscle myoblasts) which is located closely to Hsmm (human skeletal muscle myoblasts) in the MDS plot.

In MDS, the presence / absence patterns of edges is used to construct a distance matrix of all samples. Thereby, the information of the union of regulatory interactions found within networks is exploited to assess similarity.

It is likely that relatedness is more conserved in regulatory interactions than in the degree of nodes (chapter 2.2.8.3 and 2.2.8.2). The reasons for this are two-fold. Firstly, the degree does not reflect actual regulatory effects exercised by TFs and can be similar even when regulatory effects are different. This means that in NND clustering, the cell types in which regulators perform different functions can be erroneously clustered together. Secondly, when trait evolution and individualization is considered, derived traits such as related cell types are likely to possess much of the regulatory network of their ancestor because they arise through modifications of developmental regulation. Another argument in favor of this concept is differentiation in cell lineages, which produces related cell types. In cell lineages, differentiation occurs in a step-wise manner, with TF being deactivated or additional TFs being activated at each step, relative to the preceding stage. As gene expression changes gradually, also regulation is modified gradually, and differentiated cells share parts of their regulatory networks with precursors. The closer cell types are in lineages, the more regulation they share.

For these reasons, we argue that MDS is a suitable measure to assess sample-specificity and especially useful in investigating relationships between samples.

Lastly, we examined active sub-networks at the level of nodes. Functional enrichment analysis of specifically expressed TFs demonstrated that all but two networks were enriched for marker genes associated with cell type or tissue function.

In summary, our results underpin the specificity of the generated cell type- and tissue-specific active sub-networks and emphasize the validity of our methods.

4.2 Sample-specific networks

At the level of sample-specific networks, we integrate data from complementary methods to model cell type- and tissue-specific regulatory interactions. Sample-specific networks form the basis for active sub-network construction.

4.2.1 Distribution of interaction types implies advantages of species-specific TFBS models

The analysis of human and mouse sample-specific networks, shown in Table 12, revealed interesting differences between the two sample groups. More precisely, the distribution of interaction types has different trends. Human networks tend to contain more weight A interactions which require footprinted binding motifs, while in mouse networks, the proportion of weight B interactions is larger. In contrast to weight A, weight B interactions require binding motifs in regulatory regions to be devoid of DNaseI footprints, because they are assigned to unexpressed regulators. Considering that regulatory regions were derived using the same technique for both species, it follows that the likely cause for the differing distributions lies within DNaseI footprints or binding motifs.

The DNase-seq footprint counts are comparable between mouse and human samples, in spite of a tendency in mouse towards lower overall numbers (Table 10). When the number of binding motifs overlapped by footprints is examined (Table 9), it becomes apparent that a smaller percentage of predicted motifs are footprinted in mouse samples. This is unexpected. It demonstrates that many predicted binding motifs do not correspond to binding events. Since there is no reason to expect fewer binding events in the mouse samples, this result could indicate that the TFBS models used to predict binding motifs do not fully reflect the binding behavior of mouse TFs. In a broader context, this finding suggests that a common practice in databases, the use of TFBS models for mapping binding sites of homolog TFs in related species, should be reconsidered. Even though TF proteins in related species share structural and sequence similarities, their binding affinities can differ, and TFBS models could be refined to accommodate these differences.

4.2.2 Interaction types and weights pave the way for studies of network-wide dynamics

A crucial feature of this work is the introduction of activating and inhibiting interaction types and weight groups. This feature makes our approach stand out from other studies that have constructed cell type-specific networks or active sub-networks, yet have, only established direction of interactions, but not their activating or inhibiting nature. The nature of interaction is crucial for many follow-up analyses. For instance, having +/- information is a prerequisite for studying a network's dynamics, and thereby its functional repertoire. Using this evidence, it can be traced how information is transmitted in a network and disruptions of transmission can be simulated. The weights furthermore enable us to discriminate the likely pathways of information transmission from the unlikely ones. Thereby, our methodology provides first steps towards

parameterization and quantitative modeling of gene regulatory networks. This could contribute to elucidating deregulation in pathological states, help develop interference strategies, or point towards potential drug targets. In the present study we do not fully use of the activation/inhibition information, but it has many potential purposes in future.

4.3 Methodic choices and considerations

4.3.1 Reference network

Our method to generate reference networks has its basis in the approach of Nepf et al in [25] in two major aspects: 1) probabilistic modeling of TFBS and 2) definition of proximal regulatory regions as a +/-5kb interval surrounding the TSS of target TFs. However, we confined regulatory regions to introns and intergenic sequences and constructed reference networks not only for human, but also for mouse.

A drawback of the probabilistic modeling approach is the high rate of false-positive TFBS. Since binding motifs are short (typically 6-12 bp), they are detected very often. This challenge can be addressed by using a strict p-value cutoff, but the cutoff alone cannot control this issue properly. For instance, a very strict cutoff might exclude true-positive TFBS with a more unusual nucleotide composition. What is more, while TF might have a strong likelihood to bind to a sequence, the binding may have no regulatory function *in vivo*.¹⁴

Methods to discriminate regulatory TFBS from false-positives have therefore focused on properties of true regulatory regions. They include phylogenetic footprinting which makes use of the conservation of regulatory regions across different species. Another method is the detection of TFBS-clusters. So called cis-acting regulatory modules (CRM) have been found to contain clusters of binding sites which reflect the cooperative action of TFs.⁵⁵ Predicted TFBS occurring in clusters therefore have a higher probability of being true regulatory TFBS instances, and binding motif detection methods have made use of this property.

We addressed the high false-positive rate of probabilistic modeling by three different measures: 1) the introduction of DNaseI footprints to identify binding sites occupied *in vivo*, 2) the use of transcriptome data to classify expressed TFs and their interactions, and 3) by focusing on a restricted genomic regions in which to detect binding motifs. In this last measure, we only considered introns and intergenic regions inside the +/-5kb interval. The assumption here is that exons and coding sequences do not contain TFBS relevant for the regulation of transcription. This premise is in line with a study by Long et al in which second exon sequences were used to establish false-positive cutoff-scores for PWM-based motif prediction, stating that such sequences are “believed to have few intrinsic biological relevant cis-regulatory elements”.⁶⁶

In this context, we investigated the distribution of DNaseI footprints in intronic / intergenic and exonic regulatory regions to examine possible regulatory TF binding events in exons. From the overview of footprinted binding motifs in introns and exons in Table 11 we can see that there are very few binding events in exons. Possibly, the absence of TF binding in exonic regions still has a regulatory impact, but it seems rather unlikely that this applies to all of the predicted binding motifs. Moreover, this would suggest an unlikely bias in the mechanisms of transcriptional regulation: Within introns / intergenic regions binding of active TF would be the

more pronounced mechanism while within exons, the absence of active TF binding would be prevalent. For these reasons it is plausible that transcriptional regulation through direct binding of TF to exon DNA has only a small, negligible impact. In summary, our results support the exclusion of exons from regulatory regions as a model assumption for this study.

Notably, it has long been believed that coding sequences, which make up a large part of exons, do not contain TFBS. Such TFBS would be subject to two different, possibly even competing constraints: one to preserve the amino acid sequence and one to preserve the binding sites affinity.⁶⁷ In contrast, recent studies in various species suggest that TF binding can occur inside exons and that exons and coding sequences can contain transcriptional regulators/enhancers.^{68,69} The frequency and impact of TFs binding in exons and coding sequences is unknown, as most of these studies were not carried out on a genome-wide. Even so, these findings should be considered in further work.

It should be noted that the regulation of transcription in eukaryotes is a complex process that integrates many different mechanisms, beyond the one-dimensional models of transcriptional regulation we and others have focused on. Besides TF binding in proximal regulatory regions which is used here to model gene regulatory networks, it also includes chromatin modification, chromosome conformation and distal regulatory regions, TF co-regulators and direct interactions between TF proteins. Current efforts concentrate on developing techniques that include the regulatory effect of distal regulatory regions and DNA folding, and transcriptional regulation is increasingly understood as a process in three-dimensional space.^{70,71}

4.3.2 Expression calling on RNA-seq data

RNA-seq is a wide-spread tool for transcriptome profiling, however one should keep its weaknesses in mind. Many choices are made in conducting the RNASeq analysis, such as choice of sequencing depth, read length, the choice of mapping and expression analysis tools, the availability and quality of reference genomes and annotations, the choice of gene models, expression cutoff; and all of these choices can affect the results.

In spite of these caveats, the reference genomes and annotations for mouse and human that we used here are some of the best studied. To ensure comparability between samples, we used pre-processed and aligned data from the ENCODE project which has adopted standards and best practices for its RNA-seq experiments (e.g. sequencing depth and read length).⁷² Since alternative splicing creates a high diversity of transcripts in eukaryotic transcriptomes, tools that can align reads to splice junctions had been applied during mapping. Finally, the expression cutoff we used has been applied previously to characterize actively expressed genes.

4.3.3 DNaseI footprinting

DNaseI-seq and subsequent footprinting promise the identification of occupied regulatory binding-sites in a cell-type or even cell-state specific manner. Yet, DNaseI footprinting holds experimental and computational challenges. For example, it requires very high sequencing depths to detect true DNaseI footprints. The human DNaseI-seq samples in this study applied an average sequencing depth of 273 million mapped reads.

The software for DNaseI footprinting in large genomes may impose a limitation to the method. We used the R-package *DNaseR*, as the only publicly available software to our knowledge at the time. *DNaseR* relies on the fact that cleavage counts are higher in the regions flanking footprints than within footprints. It uses the correlation of two Poisson distributions corresponding to the two strand-specific cleavage counts.⁷³ An alternative approach, “digital genomic footprinting” (*DGF*), developed originally for *Saccharomyces cerevisiae*⁶¹ and later adapted for human cells³⁰, unfortunately isn’t available in the form of open software.

At the time the best practice for the validation of footprinting algorithms is the comparison with ChIP-seq results from the same biological setting (e.g. the same cell-type). Such comparisons have been done for certain *DGF*-derived footprint data sets, using footprints that matched a binding motif models from TRANSFAC or JASPAR. For high-confidence footprints from this subset, correlations were found with evolutionary conservation patterns.³⁰ For a sample of four transcription factors, high-confidence footprints were shown to correlate with strong ChIP-seq signal intensity. This validation is limited however, as ChIP-seq data sets are specific for single DNA-binding proteins while DNaseI-seq includes the all DNA-binding events in a sample.

Since a validation of the *DNaseR* algorithm was outside the scope of this project, we confined ourselves to comparing *DNaseR*-derived with *DGF*-derived footprints for all samples where this was possible (Table 8). This comparison shows *DNaseR* reproduces only 6-11% of the footprints identified by *DGF* and accentuates the need for standardizing computational methods in DNaseI footprinting.

4.3.4 Sample-specific networks

The generation of sample-specific networks is a crucial step in the workflow. Firstly, the reference network of predicted interactions is confined to those interactions that are supported by experimental evidence. When edge numbers in the generated networks are compared with the reference network, it becomes evident that a large proportion of the potential predicted edges likely bear no biological meaning. In addition to the exclusion of false-positively predicted interactions, the networks gain cell type- and tissue-specificity. This is a vital aspect considering that the aim of this study is to elucidate the importance of network structure for the regulation of traits, such as cell types and tissues. While certain regulatory interactions might be present in various tissues, others can be specific to a certain context, *i.e.* cell type or tissue. Similarly, some edges that appear falsely predicted in one context can be functional in another.

Our approach is not the first to model cell type- and tissue-specific regulatory networks. For example, specific networks have recently been constructed by Nepf et al²⁵ for 41 human cell types and Li et al²³ for different human tissues. However, the methodology developed in our study has some decisive advantages over other approaches. The combined use of probabilistic modeling to predict TF-TF interactions, gene expression profiles from RNA-seq and binding motif occupancy from DNaseI-seq makes it possible not only to assign evidence-based interactions, but also interaction types and weights. Providing signs for interactions to represent their activating or inhibiting nature and weights that prioritize edges with strong experimental evidence over less supported edges is an important step towards parameterization and the elucidation of network-wide dynamics.

The method assumes that the binding of a TF to the DNA has a regulatory effect, and that TFs are never just “parked” without an impact on regulation. Consequently, the binding motif occupancy profiles derived from DNaseI footprinting show events of actual regulation. Even though TF binding is used to predict gene expression⁷⁴, it is possible that TF binding has no function. Since we only examine proximal regulatory regions close to TSS which coincide with promoters, we consider the binding events we identified through DNaseI footprinting as reliable.

The rules developed here to assign interaction types extend the previously applied approaches. For interactions of weight A, *i.e.* interactions with expressed regulators, we followed the idea of Nepf et al in [25], confirming edges from the reference networks when a footprint overlapped the binding motif of the regulator in a target’s regulatory region. We advanced this concept by taking into account the expression state of the target to give a “+” or “-” sign to the interaction.

For interactions of weight B, we considered the consequence of unexpressed TFs. The absence of a TF can have a regulatory effect on gene expression by neutralizing an activation or a repression. However, estimating the effect of a TF’s absence is difficult. In most cases, it is overshadowed by other active TFs that bind in regulatory regions and leave footprints. In these circumstances, we assigned interactions of weight P and treated the edges as false-positive predictions. The only clear regulatory case is when there is no footprinted binding motif, and thereby no evidence of regulation through active TFs.

This methodology does include the possibility of using false-positively predicted binding motifs to support weight B interactions, since the binding motifs of their unexpressed regulators are not confirmed through DNaseI footprints. But in this case, it is possible that the reason for their target’s expression/non-expression lies precisely within the absence of a regulator. Unlike weight P edges, there are no active TFs that cause the target’s expression state. Ideally, one would examine weight B interactions across a large number of samples, but 12 human and 6 mouse samples do not allow for a rigorous inspection. With many different biological conditions, some interactions could be confirmed, namely in conditions where the former inactive TF would be expressed and its binding motif in the target’s regulatory regions was footprinted. Other interactions would remain spurious and could eventually be excluded as mere predictions.

With regard to weight P interactions, it cannot be excluded that they (*i.e.* the absence of their regulators) have a regulatory effect. However, it is unclear whether weight P interactions are activating, inhibiting or even false predictions, and the latter is likely from the current point of view. Clearly, the targets of weight P interactions are regulated by active TFs, and these regulatory interactions are supported by experimental evidence from DNaseI footprinting. It follows that using weight P edges would introduce an unknown amount of error, and we therefore excluded them from the next step, the generation of active sub-networks.

Finally, the binding of TFs to DNA is not the only mechanism through which TFs regulate transcription. Indirect binding through protein-protein interactions between TFs and the recruitment of co-factors are known to affect gene expression. However, including these levels of regulation requires more data and would moreover make the model too complex and vulnerable.

4.3.5 Active sub-network

The idea of using gene expression to identify parts of regulatory networks that are active in specific conditions is not entirely new. Kim et al²⁴, have used differential gene expression to detect active sub-networks in distinct areas of *Drosophila melanogaster* embryos across different developmental stages, and investigated “spatiotemporal” network motifs within them. In contrast to their work, gene expression calls (presence / absence calls) were used in the present study, because distinct cell types and tissues were examined instead of a time course of observations of one single sample and its development.

As our aim is to elucidate structural patterns in regulatory networks that control traits, *i.e.* their development, maintenance and identity, the method developed in this project considers interactions between active / expressed TFs, as well as those involving unexpressed TFs. Since developmental decisions, *e.g.* differentiation across cell lineages, frequently rely on the repression of regulators that would push development towards an alternative cell fate, including inactivated / unexpressed TFs enables us to see the complete picture. Furthermore, their interactions, or rather the repression thereof, may be necessary to activate cell type- or tissue-specific regulators.

Another important difference from study of Kim et al is our use of DNaseI footprints. This type of evidence enabled us to distinguish realized predicted binding site from false-positives and thereby reduced the number of spurious edges in active sub-networks considerably. Conversely, Kim et al’s regulatory network relied on the manually curated TRANSFAC database.

Consequently, the active sub-networks generated here are different from the active sub-networks constructed in Kim et al’s study, even though our aims are certainly related.

4.3.6 Network motif detection

The method we applied to detect network motifs required a removal of self-edges, *i.e.* edges for which one node is both regulator and target, from the network. This is inconvenient because self-edges convey auto-regulatory effects which are known important mechanisms in developmental processes that can filter signal from noise.⁷⁵ Still, the exclusion of self-edges is necessary because all network motif detection methods rely on the comparison with randomized networks and the established randomization technique for the majority of methods is edge-switching. In this technique, the degree (*i.e.* the number of connections) of each node is kept constant while interaction partners are switched. Self-edges do not fit into this paradigm because switching a self-edge from one node to another would change their degrees. Due to this problem, most methods cannot detect network motifs that contain nodes with self-edges. Only recently, a new randomization algorithm that can deal with self-edges has been presented.⁷⁶ But it has not yet been implemented in a detection software with comparable functionality to MFINDER, the one used in this study.

4.3.7 Functional enrichment analysis

The results of functional enrichment analysis can vary greatly depending on the background that is chosen to detect significantly enriched functional annotation terms. Usually, gene lists should be compared against the background set of genes from which they were selected.⁵⁶ In our case,

this would have been the union of human TFs or mouse TFs applied in this study. Since these sets contain only around 400 genes however, using them as a background did not give enough power to detect enriched annotation terms. Applying a species-specific genome-wide background of all genes was one of the possible alternatives, and having such a large set of genes as a background would have given more significant enrichment scores. We followed a more conservative approach instead and used all transcription factor genes of a species as a background. Their functional annotations were reliable, because we included only on reviewed transcription factors.

A further limitation was the number of genes, which were analyzed for functional enrichment. Using short lists reduces the statistical power of the test, yet the exclusion of certain genes was necessary to ensure that the test was meaningful. For example, when examining functional enrichment in active sub-networks, we excluded “collectively” expressed TFs because they likely reflect unspecific housekeeping functions. Furthermore, we restricted the enrichment analysis on network motifs to expressed TFs because our question was related to active, cell type- and tissue-specific regulators and their use of network motif dynamics.

Finally, the choice of enrichment cutoffs was critical. On the one hand, statistical significance has to be ensured, but strict cutoffs, for example for multiple testing correction methods, can also limit sensitivity. Because we worked with small gene sets for many of the samples, we reported also genes above the Benjamini correction cutoff of ≤ 0.05 .

4.4 Scope of the study

In the previous chapters, we have detailed the influence of method choices on the results. From this point on, we focus on potential for improving the methodology, and the potential for extending the study to address related interesting questions.

Due to the limited availability of high-quality TFBS models in public databases, and because each TF in the network has to be represented by at least one TFBS in the process of modeling the reference network, our networks are composed of a relative small number nodes (~400 TFs). In contrast, the number of confirmed human TFs in the protein reference database UniProt KB currently exceeds 1,700. Even though other studies that constructed cell type- and tissue-specific regulatory networks used TF sets with comparable sizes^{25,23}, it will be interesting to expand networks and acquire a more complete picture of the transcriptional regulation governed by TFs in the future. The generation of reliable TFBS models describing the potentially diverse repertoire of binding sites that a TF recognizes is challenging. While traditional experimental techniques produce models that are less likely to detect more degenerate binding sites far from the consensus sequence, the new high-throughput techniques typically output longer sequences which may contain binding sites of other TFs as well.³⁸ HOCOMOCO, the TFBS model database applied in this study, aims at resolving this problem by integrating binding sites obtained from both low- and high-throughput experiments.

Another potentially important addition to the network would be to incorporate miRNAs and their binding interactions. This would be especially interesting in the context of our study, as the cell type- and tissue- specific activity of miRNAs has been demonstrated.

In general, the tools of regulatory network characterization and comparison developed in this work present interesting potential for numerous biological questions across varying range of samples.

Among these are questions about the underlying network changes during cell type ontological differentiation as well as evolutionary divergence; normal versus pathological state, different environmental conditions as well as drug treatments.

Examining additional tissues and cell types, particularly in a systematic manner along developmental stages or cell lineages would enable an investigation of local sub-networks around trait-specific developmental and differentiation regulators. Such studies could shed more light on how network motifs are applied in decisions of developmental fate. Comparisons of homologous cell types and tissues across greater number of species would enable testing the hypothesis that the developmental programs underlying homologous characters / traits are regulated by highly conserved core gene regulatory networks, so-called Character Identity Networks (ChINs).⁹ It would be very interesting to see if network motifs can be detected in these small core networks as well. At present however, our study was limited by the availability of complementing RNA-seq and DNase-seq data sets, preventing us from addressing homologous characters.

The comparison healthy vs. disease states of the same trait (e.g. an organ) is another application that could inform about the pathways involved.

Large collections of sample-specific networks can also be regarded as different conditions of the same general regulatory apparatus and could be used as experimental evidence to improve networks. Our current methodology proposes three weight groups, which depend on the strength of evidence that underlies an interaction. While interactions with lower weight allow us to keep track of indirect effects caused by the absence of regulators, they also leave open the possibility of assigning interactions based on falsely-predicted binding sites. This is problematic, because false positive predictions occur at a high rate and are common in the probabilistic modeling approach applied in this study. Even though we restricted false positives by limiting the regulatory regions in which we scanned for putative binding sites, our networks could benefit from a more extensive validation of low-weight edges. Monitoring of interactions across different conditions would provide more evidence for some interactions, while others could be identified as false more reliably. What is more, this investigation could reveal heterogeneity of regulation across cell types or tissues.

Our study also points towards the aspects where improvement may be most needed. We have shown in chapter 3.3.1 that there is currently little agreement between different algorithms that produce genome-wide profiles of binding-protein occupancy from DNase-seq data. This highlights the need for a gold standard in DNase footprinting methods. With an established standard, future studies building on our work will benefit from more accurate and reliable

footprint data generated by improved algorithms.

4.5 The common structure of the dynamical core of traits

One of the main questions of our study was whether common patterns in the core regulatory topologies can be detected among phenotypically different samples. The core regulatory network is assumed to determine the trait identity, *i.e.* define its essential characteristics and distinguish it from other similar traits. The mechanisms mediating the central importance of this network for the trait are development (cell differentiation) and maintenance. At the level of cell types and cell lineages, organs and body plans, single genes or small groups of genes from the core networks have been determined as marker genes of trait-specific development, e.g. PAX5 in B-cell differentiation⁷⁷ or PDX1, PTF1A, GATA4 and GATA6 in pancreas development⁷⁸, etc. Using methods such as co-expression and knock-out experiments, reporter gene assays and chromatin immunoprecipitation (ChIP), the interactions between marker genes have been characterized to varying extents, revealing in some cases small gene networks that are necessary for the development of a trait.

Several design principles of the wiring of regulatory interactions that underlie development have already been identified. For example, in cell lineage decisions, regulators have been reported to inhibit one another's expression, thereby driving development into mutually-exclusive cell fates (so-called cross-antagonism in binary cell-fate decisions⁷⁹), and to be involved in feed-forward cascades that activate further cell type-specific regulators.⁸⁰

Individual studies of regulatory circuits that determine the development of organs and body parts have also brought about interesting findings. According to a study of developmental regulation in the early pancreatic organ niche, a feed-forward loop module consisting of transcription factors (TFs) SOX9, FGFR2b and FGF10 is necessary to direct multipotent progenitors towards a pancreatic cell fate.⁸¹ Two of these TFs (SOX9, FGF10) are indeed annotated as involved in development of pancreatic cells (query on <http://amigo.geneontology.org/amigo/>).

A study that examined the transcriptional regulation underlying sex comb development in *Drosophila melanogaster* found a positive feedback loop between two regulators.⁸² This finding is interesting also in an evolutionary context. The positive feedback loop not only stabilizes the expression of regulators in a noisy environment and thereby contributes to robustness, it also promotes the evolvability of the sex comb by amplifying the effect of mutations and making them visible in the phenotype, allowing for selection to act.

The positive feedback loop has also been shown to implement irreversible commitment during *Xenopus* oocyte maturation, a model of cell fate induction.⁸³ In this study, the positive feedback loop was characterized as a “memory module” which is able to induce permanent changes in the cell based on a transient signal. Even though this module acts on the level of cell signaling, this finding highlights the functionality of small circuits in developmental decisions.

The aforementioned design principles in lineage differentiation decisions and the evidences of feed-forward and feedback loops in the regulation of individual characters imply that specific

topological patterns could generally be involved in trait regulation. In the present study, we have addressed this hypothesis by investigating the role of small, recurring sub-graphs (network motifs). Our results show that cell type and tissue marker genes are enriched on network motifs that occur in the core active regulatory sub-networks and indicate that network motif dynamics could be exploited by the regulatory processes that underlie character identity. Since marker genes enrich on feed-forward loops across many different cell types and tissues and in two different species, it is very likely that its dynamics is involved in trait regulation.

5 CONCLUSION & OUTLOOK

Network motifs are common in the dynamical core of phenotypically diverse traits

In this study, we present a new method to generate cell type- and tissue-specific gene regulatory networks (GRNs) and investigate the role of small, recurring sub-graphs (network motifs) in the regulation of traits such as cell types and tissues. We applied our workflow to 18 trait-specific GRNs from two different species, human and mouse, and for both we find that marker genes, *i.e.* TFs that have been characterized as important trait-specific regulators, are enriched on certain network motifs, especially the feed-forward loop. These results implicate that dynamics implemented by network motifs are used in the regulation that underlies trait development, maintenance and identity. As our findings stem from traits of different complexity, namely cell-types and tissues, they also support the hypothesis of common topological patterns involved in trait regulation across different hierarchical levels (e.g. cells, organs, body parts). Drawing on this result, it would be interesting to investigate network motifs in the regulatory cores of related samples, e.g. along cell lineages. This would aid to clarify if and how network motifs contribute to the development of new traits.

GRN interaction repertoire recapitulates functional relationships of cell types and tissues

We systematically compared the GRN of different samples using standard network measures, a structural clustering technique (normalized network degree clustering) and two methods based on the interaction repertoire, *i.e.* the union of interactions found in a GRN. Our results show that the networks we generated are diverse and exhibit a functional enrichment related to their corresponding trait. What is more, multidimensional scaling revealed that the GRN of related samples are more alike in their interaction repertoire than unrelated ones. Based on this result, we propose multidimensional scaling as a method of functional comparison of GRN.

The integration of complementary data types enables first steps towards GRN parameterization

In order to obtain trait-specific GRN, we combined a reference network obtained by computational prediction with experimental data. The use of data from gene expression assays and binding motif occupancy profiles allowed us to focus on the active regulatory core of each respective trait and to limit false-positive prediction of interactions. Moreover, through the integration of these complementary data types, we were able to infer the activating or inhibitory nature of interactions. In future work, it will be necessary to systematically compare inferred interaction types to database knowledge. Furthermore, systematic screens of interactions in large numbers of samples could uncover regulatory diversity.

Our study points towards where methodic improvements are needed

In various areas, our study highlighted the shortcomings of presently used methods. Firstly, interaction type distributions and binding motif occupancy profiles in human and mouse GRN demonstrated that predicted binding motifs are found markedly less often occupied in mouse GRN. This suggests that the predicted mouse binding sites are inaccurate and implies that GRN

modeling could benefit from species-specific TF binding site models. In addition, our project uncovered, that DNaseI footprints obtained using two present algorithms differ significantly, which emphasizes the need for standard methods in this area. Finally, this work pointed out the importance of systematic data acquisition for questions of trait development and evolution.

Abbreviations

A549 – Epithelial cell line derived from a lung carcinoma tissue
Cd20 – B cells
CRM – cis-acting regulatory module
ChIP – chromatin immunoprecipitation
DBP – DNA-binding protein
DGF – digital genomic footprinting
FDR – false discovery rate
FFL – feed-forward loop
FP – footprint
GRN – gene regulatory network
Hepg2 – hepatocellular carcinoma
Hsmm – human skeletal muscle myoblast
Huvec – umbilical vein endothelial cells
HRNW – human reference network
IA – interaction
K562- cell line from myelogenous leukemia
Lhcnm2 – skeletal myoblast cell line
Lhcnm2diff – differentiated skeletal myoblast cell line
MDS – multidimensional scaling
Monocd14 – monocytes
MRNW – mouse reference network
NGS – next generation sequencing
Nhlf – lung fibroblasts
NND – normalized network degree
NW – network
ODE – ordinary differential equation
TF – transcription factor
TFBS – transcription factor binding site
Sknshra – neuroblastoma cell line

REFERENCES

- ¹ Lawrence, E. Henderson's Dictionary of Biology. Pearson, Prentice Hall (2005)
- ² Wagner, G. P. Homology, Genes, and Evolutionary Innovation. (Princeton University Press, 2014).
- ³ Wagner, G. P. & Lynch, V. J. Evolutionary novelties. *Current Biology* 20, R48–R52 (2010).
- ⁴ Sanetra, M., Begemann, G., Becker, M.-B. & Meyer, A. Conservation and co-option in developmental programmes: the importance of homology relationships. *Frontiers in Zoology* 2, 15 (2005).
- ⁵ Arendt, D. The evolution of cell types in animals: emerging principles from molecular studies. *Nat. Rev. Genet.* 9, 868–882 (2008).
- ⁶ Arendt, D. Evolution of eyes and photoreceptor cell types. *Int. J. Dev. Biol.* 47, 563–571 (2003).
- ⁷ Shubin, N., Tabin, C. & Carroll, S. Deep homology and the origins of evolutionary novelty. *Nature* 457, 818–823 (2009).
- ⁸ McCune, A. R. & Schimenti, J. C. Using Genetic Networks and Homology to Understand the Evolution of Phenotypic Traits. *Curr Genomics* 13, 74–84 (2012).
- ⁹ Wagner, G. P. The developmental genetics of homology. *Nat Rev Genet* 8, 473–479 (2007).
- ¹⁰ Maston, G. A., Evans, S. K. & Green, M. R. Transcriptional regulatory elements in the human genome. *Annu Rev Genomics Hum Genet* 7, 29–59 (2006).
- ¹¹ Cooper, G. M. *The Cell*. (Sinauer Associates, 2000).
- ¹² Reményi, A., Schöler, H. R. & Wilmanns, M. Combinatorial control of gene expression. *Nature Structural & Molecular Biology* 11, 812–815 (2004).
- ¹³ Bulger, M. & Groudine, M. Enhancers: The abundance and function of regulatory sequences beyond promoters. *Developmental Biology* 339, 250–257 (2010).
- ¹⁴ Wasserman, W. W. & Sandelin, A. Applied bioinformatics for the identification of regulatory elements. *Nat Rev Genet* 5, 276–287 (2004).
- ¹⁵ Hardison, R. C. & Taylor, J. Genomic approaches towards finding cis-regulatory modules in animals. *Nat. Rev. Genet.* 13, 469–483 (2012).
- ¹⁶ Dermitzakis, E. T. & Clark, A. G. Evolution of transcription factor binding sites in Mammalian gene regulatory regions: conservation and turnover. *Mol. Biol. Evol.* 19, 1114–1121 (2002).
- ¹⁷ Bradley, R. K. et al. Binding Site Turnover Produces Pervasive Quantitative Changes in Transcription Factor Binding between Closely Related *Drosophila* Species. *PLoS Biol* 8, e1000343 (2010).
- ¹⁸ Wray, G. A. The evolutionary significance of cis-regulatory mutations. *Nat Rev Genet* 8, 206–216 (2007).
- ¹⁹ Arvey, A., Agius, P., Noble, W. S. & Leslie, C. Sequence and chromatin determinants of cell-type-specific transcription factor binding. *Genome Res.* 22, 1723–1734 (2012).
- ²⁰ Ellis, J. D. et al. Tissue-Specific Alternative Splicing Remodels Protein-Protein Interaction Networks. *Molecular Cell* 46, 884–892 (2012).
- ²¹ Grove, C. A. & Walhout, A. J. M. Transcription factor functionality and transcription regulatory networks. *Mol. Biosyst.* 4, 309–314 (2008).
- ²² Sood, P., Krek, A., Zavolan, M., Macino, G. & Rajewsky, N. Cell-type-specific signatures of microRNAs on target mRNA expression. *Proceedings of the National Academy of Sciences* **103**, 2746–2751 (2006).
- ²³ Li, J., Hua, X., Haubrock, M., Wang, J. & Wingender, E. The architecture of the gene regulatory networks of different tissues. *Bioinformatics* 28, i509–i514 (2012).
- ²⁴ Kim, M.-S., Kim, J.-R., Kim, D., Lander, A. D. & Cho, K.-H. Spatiotemporal network motif reveals the biological traits of developmental gene regulatory networks in *Drosophila melanogaster*. *BMC Systems Biology* 6, 31 (2012).
- ²⁵ Neph, S. et al. Circuitry and Dynamics of Human Transcription Factor Regulatory Networks. *Cell* **150**, 1274–1286 (2012).
- ²⁶ Trapnell, C., Pachter, L. & Salzberg, S. L. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* 25, 1105–1111 (2009).
- ²⁷ Dobin, A. et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* 29, 15–21 (2013).
- ²⁸ Tilgner, H., Grubert, F., Sharon, D. & Snyder, M. P. Defining a personal, allele-specific, and single-molecule long-read transcriptome. *PNAS* 201400447 (2014). doi:10.1073/pnas.1400447111
- ²⁹ Galas, D. J. & Schmitz, A. DNase footprinting: a simple method for the detection of protein-DNA binding specificity. *Nucleic Acids Res* **5**, 3157–3170 (1978).
- ³⁰ Neph, S. et al. An expansive human regulatory lexicon encoded in transcription factor footprints. *Nature* 489, 83–90 (2012).

- ³¹ Barabási, A.-L. & Oltvai, Z. N. Network biology: understanding the cell's functional organization. *Nat Rev Genet* **5**, 101–113 (2004).
- ³² Rodríguez-Caso, C., Corominas-Murtra, B. & Solé, R. V. On the basic computational structure of gene regulatory networks. *Mol Biosyst* **5**, 1617–1629 (2009).
- ³³ Milo, R. Network Motifs: Simple Building Blocks of Complex Networks. *Science* **298**, 824–827 (2002).
- ³⁴ Widder, S., Solé, R. & Macía, J. Evolvability of feed-forward loop architecture biases its abundance in transcription networks. *BMC Syst Biol* **6**, 7 (2012).
- ³⁵ Wagner, G. P. & Stadler, P. F. Quasi-independence, homology and the unity of type: a topological theory of characters. *J. Theor. Biol.* **220**, 505–527 (2003).
- ³⁶ Kersey, P. J. *et al.* Ensembl Genomes: an integrative resource for genome-scale data from non-vertebrate species. *Nucleic Acids Research* **40**, D91–D97 (2011).
- ³⁷ Harrow, J. *et al.* GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Res.* **22**, 1760–1774 (2012).
- ³⁸ Kulakovskiy, I. V. *et al.* HOCOMOCO: a comprehensive collection of human transcription factor binding sites models. *Nucleic Acids Research* **41**, D195–D202 (2012).
- ³⁹ The ENCODE Project Consortium. A User's Guide to the Encyclopedia of DNA Elements (ENCODE). *PLoS Biol* **9**, e1001046 (2011).
- ⁴⁰ Trapnell, C. *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotech* **28**, 511–515 (2010).
- ⁴¹ Kinsella, R. J. *et al.* Ensembl BioMart: a hub for data retrieval across taxonomic space. *Database* **2011**, bar030–bar030 (2011).
- ⁴² Magrane, M. & Consortium, U. UniProt Knowledgebase: a hub of integrated protein data. *Database* **2011**, bar009–bar009 (2011).
- ⁴³ Stajich, J. E. *et al.* The Bioperl Toolkit: Perl Modules for the Life Sciences. *Genome Res.* **12**, 1611–1618 (2002).
- ⁴⁴ Grant, C. E., Bailey, T. L. & Noble, W. S. FIMO: scanning for occurrences of a given motif. *Bioinformatics* **27**, 1017–1018 (2011).
- ⁴⁵ Swindell, W. R. *et al.* Meta-Profiles of Gene Expression during Aging: Limited Similarities between Mouse and Human and an Unexpectedly Decreased Inflammatory Signature. *PLoS ONE* **7**, e33204 (2012).
- ⁴⁶ Hart, T., Komori, H. K., LaMere, S., Podshivalova, K. & Salomon, D. R. Finding the active genes in deep RNA-seq gene expression studies. *BMC Genomics* **14**, 778 (2013).
- ⁴⁷ Graveley, B. R. *et al.* The developmental transcriptome of *Drosophila melanogaster*. *Nature* **471**, 473–479 (2011).
- ⁴⁸ Pedro Madrigal (2013). DnaseR: Dnase I footprinting analysis of Dnase-seq data. R package version 1.0.0.
- ⁴⁹ Neph, S. *et al.* BEDOPS: high-performance genomic feature operations. *Bioinformatics* **28**, 1919–1920 (2012).
- ⁵⁰ Bastian M., Heymann S., Jacomy M. (2009). Gephi: an open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media.
- ⁵¹ Milo, R. *et al.* Superfamilies of Evolved and Designed Networks. *Science* **303**, 1538–1542 (2004).
- ⁵² R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- ⁵³ Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006. <http://igraph.sf.net>
- ⁵⁴ Hadley Wickham (2011). The Split-Apply-Combine Strategy for Data Analysis. *Journal of Statistical Software*, 40(1), 1-29. URL <http://www.jstatsoft.org/v40/i01/>
- ⁵⁵ Huang, D. W., Sherman, B. T. & Lempicki, R. A. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucl. Acids Res.* **37**, 1–13 (2009).
- ⁵⁶ Huang, D. W., Sherman, B. T. & Lempicki, R. A. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat Protoc* **4**, 44–57 (2009).
- ⁵⁷ Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
- ⁵⁸ Hartwell, L. H., Hopfield, J. J., Leibler, S. & Murray, A. W. From molecular to modular cell biology. *Nature* **C47** (1999). doi:10.1038/35011540
- ⁵⁹ Eisenberg, E. & Levanon, E. Y. Human housekeeping genes, revisited. *Trends Genet.* **29**, 569–574 (2013).

-
- ⁶⁰ Kouadjo, K. E., Nishida, Y., Cadrin-Girard, J. F., Yoshioka, M. & St-Amand, J. Housekeeping and tissue-specific genes in mouse tissues. *BMC Genomics* **8**, 127 (2007).
- ⁶¹ Hesselberth, J. R. et al. Global mapping of protein-DNA interactions in vivo by digital genomic footprinting. *Nat Meth* **6**, 283–289 (2009).
- ⁶² <http://www.uniprot.org/uniprot/P15172>
- ⁶³ Odom, D. T. et al. Control of pancreas and liver gene expression by HNF transcription factors. *Science* **303**, 1378–1381 (2004).
- ⁶⁴ Costa, R. H., Kalinichenko, V. V., Holterman, A.-X. L. & Wang, X. Transcription factors in liver development, differentiation, and regeneration. *Hepatology* **38**, 1331–1347 (2003).
- ⁶⁵ Mangan, S., Zaslaver, A. & Alon, U. The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks. *J. Mol. Biol.* **334**, 197–204 (2003).
- ⁶⁶ Long, F. et al. Genome-wide prediction and analysis of function-specific transcription factor binding sites. In *Silico Biol. (Gedruckt)* **4**, 395–410 (2004).
- ⁶⁷ Tümpel, S., Cambrero, F., Sims, C., Krumlauf, R. & Wiedemann, L. M. A regulatory module embedded in the coding region of *Hoxa2* controls expression in rhombomere 2. *PNAS* **105**, 20077–20082 (2008).
- ⁶⁸ Birnbaum, R. Y. et al. Coding exons function as tissue-specific enhancers of nearby genes. *Genome Res.* **22**, 1059–1068 (2012).
- ⁶⁹ Ritter, D. I., Dong, Z., Guo, S. & Chuang, J. H. Transcriptional Enhancers in Protein-Coding Exons of Vertebrate Developmental Genes. *PLoS ONE* **7**, e35202 (2012).
- ⁷⁰ Lan, X., Farnham, P. J. & Jin, V. X. Uncovering Transcription Factor Modules Using One- and Three-dimensional Analyses. *J Biol Chem* **287**, 30914–30921 (2012).
- ⁷¹ Splinter, E. & de Laat, W. The complex transcription regulatory landscape of our genome: control in three dimensions. *EMBO J* **30**, 4345–4355 (2011).
- ⁷² https://genome.ucsc.edu/ENCODE/protocols/dataStandards/RNA_standards_v1_2011_May.pdf
- ⁷³ <http://www.bioconductor.org/packages/devel/bioc/vignettes/DNaseR/inst/doc/DNaseR.pdf>
- ⁷⁴ Cheng, C. & Gerstein, M. Modeling the relative relationship of transcription factor binding and histone modifications to gene expression levels in mouse embryonic stem cells. *Nucleic Acids Res* **40**, 553–568 (2012).
- ⁷⁵ Stelling, J., Sauer, U., Szallasi, Z., Doyle, F. J., 3rd & Doyle, J. Robustness of cellular functions. *Cell* **118**, 675–685 (2004).
- ⁷⁶ Megraw, M., Mukherjee, S. & Ohler, U. Sustained-input switches for transcription factors and microRNAs are central building blocks of eukaryotic gene circuits. *Genome Biology* **14**, R85 (2013).
- ⁷⁷ O'Brien, P., Morin, P., Jr, Ouellette, R. J. & Robichaud, G. A. The Pax-5 gene: a pluripotent regulator of B-cell differentiation and cancer disease. *Cancer Res.* **71**, 7345–7350 (2011).
- ⁷⁸ Xuan, S. et al. Pancreas-specific deletion of mouse *Gata4* and *Gata6* causes pancreatic agenesis. *Journal of Clinical Investigation* **122**, 3516–3528 (2012).
- ⁷⁹ Graf, T. & Enver, T. Forcing cells to change lineages. *Nature* **462**, 587–594 (2009).
- ⁸⁰ Holmberg, J. & Perlmann, T. Maintaining differentiated cellular identity. *Nat Rev Genet* **13**, 429–439 (2012).
- ⁸¹ Seymour, P. A. et al. A *Sox9*/*Fgf* feed-forward loop maintains pancreatic organ identity. *Development* **139**, 3363–3372 (2012).
- ⁸² Kopp, A. *Drosophila* Sex Combs as a Model of Evolutionary Innovations. *Evol Dev* **13**, 504–522 (2011).
- ⁸³ Xiong, W. & Ferrell, J. E. A positive-feedback-based bistable ‘memory module’ that governs a cell fate decision. *Nature* **426**, 460–465 (2003).

SUPPLEMENTAL MATERIAL

```
extract_reg_regions.pl

#!/usr/bin/perl

use strict;
use warnings;
use files;

use Bio::SeqIO;
use Bio::Seq;
use Bio::PrimarySeq;

my $open = "<";
my $write = ">";
my $add = ">>";

# # top section # #
# specify parameters, name of fasta-genome and cutoff for regulatory regions
my $fasta_genome = "Homo_sapiens.GRCh37.73.dna.primary_assembly.fa";
my @excluded_chromosomes = ("MT"); # do you want to have tf "SRX" in or not? specify "Y" to exclude chrY
my $exon_flag = $ARGV[1];
my $shard_cutoff = 5000;

#create outfile for fasta sequences of reg regions and log file. different name according to exon flag
#if exon flag 1: first exon included in regulatory regions, if exon flag = 0 not included
my $out_file_regions = "reg_regions/batch" . $ARGV[0];
my $log_file_name = $out_file_regions;
if ($exon_flag == 1) {
    $out_file_regions .= ".exon.out";
    $log_file_name .= ".exon.log";
}
else {
    $out_file_regions .= ".out";
    $log_file_name .= ".log";
}
my $log_fh = files::write_or_open($log_file_name, $write);
my $out_regions_fh = files::write_or_open($out_file_regions, $write);

# # # FASTA GENOME - make a hash with chromosomes as keys and sequences as values
my %chromosome_seqs;
my $genome_seqio = Bio::SeqIO->new(-file => $fasta_genome, -format => 'Fasta');

while(my $seq = $genome_seqio->next_seq) {
    my $id = $seq->display_id;
    my $description = $seq->description;
    my $length = $seq->length;

# #
# initialize $check_if_excluded_chromosome. this is a control variable that will exclude specified chromosomes; 1 = exclude, 0 = don't exclude
# set control variable for exclusion to 1, if current sequence represents excluded chromosome
my $check_if_excluded_chromosome = 0;
foreach my $excluded (@excluded_chromosomes) {
    if ($id eq $excluded){
        $check_if_excluded_chromosome = 1;
    }
}
# take only sequences that contain the whole chromosome (marked "dna:chromosome"; no supercontigs etc)
# take only chromosomes that have not been excluded (see top section)
if ($description =~ /dna:chromosome/) {
    if($check_if_excluded_chromosome == 0) {
        my $chr_name = "chr" . $id;
        $chromosome_seqs{$chr_name} = $seq;
    }
}
}

#open annotation of isoforms of tfoi
```

```

my $annotation_basepath = $ARGV[0]
my $transcript_annotation = "/scratch/todtova/transcriptome_pipeline/annotations/genecode.v18.annotation.gtf.isoforms.batch". $ARGV[0]; # $annotation_basepath .
"all_tfoi_isoforms_only"; ## /scratch/todtova/transcriptome_pipeline/annotations/test_annotation.isoforms_only"; #
my $transcript_fh = files::write_or_open($transcript_annotation, $open);
my @transcripts = <$transcript_fh>;

# # # open complete annotation
my $genecode_file = "/scratch/todtova/transcriptome_pipeline/annotations/genecode.v18.annotation.gtf";
my $genecode_fh = files::write_or_open($genecode_file, $open);
my @genecode = <$genecode_fh>;
my $genecode_ref = \@genecode;

# # # generate a hash with the transcripts and some descriptive information in the key
my %transcript_hash;
foreach my $line (@transcripts) {
    if ($line =~ /^/) { # operate only on annotation lines, not on empty ones (; is delimiter!)
        my @split_line = split(/\s/, $line);
        my $iso_id = $split_line[23];
        my $chr = $split_line[0];
        my $strand = $split_line[6];
        my $ENST_tag = $split_line[11];
        my $transcript_biotype = $split_line[19];
        my $knowledge = $split_line[21];
        my $transcript_key = $iso_id . $chr . "." . $strand . "." . $ENST_tag . $transcript_biotype . $knowledge;
        $transcript_hash{$transcript_key} = "";
    }
}

# # # end of top section # #

# # # main # #
foreach my $transcript (sort keys %transcript_hash) {
    my ($tss, $lower_boundary, $upper_boundary, $downstream_overhang, $up_or_down_reg_regions, $gene_end);
    my (@start, @end);
    my (@introns, @introns_comp);
    print $log_fh "\n$transcript\n";
    my @split = split(/\./, $transcript);
    chop $split[0];
    my $isoform_id;
    if ($split[0] =~ /^[^\"]+\."$/ ) {
        $isoform_id = $1;
    }
    my $ENST_tag = $split[3];
    my $strand_tf = $split[2];
    my $chr_name = $split[1];
    my $biotype = $split[4];
    # print $transcript, "\n";
    if ($biotype eq "\"protein_coding\"") {
        # put all start and end coordinates of a transcript (identified by ENST tag) that are tagged as exons into arrays
        foreach my $annotation_line (@transcripts) {
            if ($annotation_line =~ /^$ENST_tag/) {
                my @annotation_split = split(/\t/, $annotation_line);
                if ($annotation_split[2] eq "exon") { # if the line is tagged as exon, not as CDS etc, put starts and ends into arrays
                    push(@start, $annotation_split[3]);
                    push(@end, $annotation_split[4]);
                }
            }
        }
    }

    # # #
    # sort the start and end coordinates in ascending order and find coordinates within the cutoff;
    # store start -> end in a hash (depending on strandedness @introns or @introns_comp)
    my @starts = sort {$a<=>$b} @start;
    my @ends = sort {$a<=>$b} @end;
    my $scalar_of_ends = scalar(@ends);
    my $index_of_last_end = $scalar_of_ends - 1;

    # # #
    # go through the sorted arrays and find intron coordinates (connect exon ends with the start of the next closest exon)
    for my $count (0..($index_of_last_end-1)) {
        if ($strand_tf eq "+") { # if on plus strand

```

```

$ss = $starts[0];
$gene_end = $ends[$index_of_last_end];
$upper_boundary = $ss + $hard_cutoff;
$lower_boundary = $ss - $hard_cutoff;
unless ($starts[$count+1]-1 > $upper_boundary) {
    if ($exon_flag != 1) {
        $introns{($ends[$count]+1)} = ($starts[$count+1]-1); # end coord of exon -> start coord of next exon in hash if they are both smaller
    }
    else {
        if ($count == 0) {
            $introns{$ss} = ($starts[$count+1]-1);
        }
        else {
            $introns{($ends[$count]+1)} = ($starts[$count+1]-1);
        }
    }
}
if ($starts[$count+1]-1 > $upper_boundary) {
    if ($exon_flag != 1) {
        if ($ends[$count]+1 < $upper_boundary) {
            $introns{($ends[$count]+1)} = $upper_boundary; # end coord of exon-> upper boundary in hash if start coord of next exon
        }
    }
    else {
        if ($count == 0) {
            $introns{$ss} = $upper_boundary;
        }
        else {
            if ($ends[$count]+1 < $upper_boundary) {
                $introns{($ends[$count]+1)} = $upper_boundary;
            }
        }
    }
}
}
if ($strand eq "-") { # if on minus strand
    $gene_end = $starts[0];
    $ss = $ends[$index_of_last_end];
    $upper_boundary = $ss + $hard_cutoff;
    $lower_boundary = $ss - $hard_cutoff;
    unless ($ends[$count]+1 < $lower_boundary) {
        if ($exon_flag != 1) {
            $introns_comp{($ends[$count]+1)} = ($starts[$count+1]-1);
        }
        else {
            if ($count == ($index_of_last_end - 1)) {
                $introns_comp{($ends[$count]+1)} = $ss;
            }
            else {
                $introns_comp{($ends[$count]+1)} = ($starts[$count+1]-1);
            }
        }
    }
    if ($ends[$count]+1 < $lower_boundary) {
        if ($exon_flag != 1) {
            if ($starts[$count+1]-1 > $lower_boundary) {
                $introns_comp{$lower_boundary} = ($starts[$count+1]-1);
            }
        }
        else {
            if ($count == ($index_of_last_end - 1)) {
                $introns_comp{$lower_boundary} = $ss;
            }
            else {
                if ($starts[$count+1]-1 > $lower_boundary) {
                    $introns_comp{$lower_boundary} = ($starts[$count+1]-1);
                }
            }
        }
    }
}

```

```

    }
  }
}

# # #
# in case that $index_of_last_end = 0, the transcript will not go through the loops above, because there are no
# introns to get. upper and lower boundary have to be initiated seperately in this case.
if ($index_of_last_end == 0) {
  if ($strand_tf eq "+") {
    $tss = $starts[0];
    $gene_end = $ends[$index_of_last_end];
  }
  else {
    $tss = $ends[$index_of_last_end];
    $gene_end = $starts[0];
  }
  $upper_boundary = $tss + $hard_cutoff;
  $lower_boundary = $tss - $hard_cutoff;
}

# # #
# now, once the boundaries are initiated for all transcripts, print to the log fh.
# also print the intronic sequences that were retrieved in the loop 2 steps above.
print $log_fh "tss & limits: $lower_boundary..$tss..$upper_boundary\ntf_end: $gene_end\n";
unless (!%introns) {
  print $log_fh "inside gene +\n";
  print_from_hash_ref_to_fh(\%introns, \ $log_fh);
}
unless (!%introns_comp) {
  print $log_fh "inside gene -\n";
  print_from_hash_ref_to_fh(\%introns_comp, \ $log_fh);
}

my ($introns_down_overhang_ref, $introns_up_overhang_ref, $upstream_introns_ref, $downstream_introns_ref);
# this part gets regulatory regions in the overhanging regions, if a gene ends before the boundaries
if ($strand_tf eq "+" && $upper_boundary > $ends[$index_of_last_end]) {
  my $upordown = "down";
  print $log_fh "overhang +\n";
  $introns_down_overhang_ref = up_or_down_reg_regions($upper_boundary, $ends[$index_of_last_end], $chr_name, $upordown, $gencode_ref, \ $log_fh);
  print_from_hash_ref_to_fh($introns_down_overhang_ref, \ $log_fh);
}
if ($strand_tf eq "-" && $lower_boundary < $starts[0]) {
  my $upordown = "up";
  print $log_fh "overhang -\n";
  $introns_up_overhang_ref = up_or_down_reg_regions($lower_boundary, $starts[0], $chr_name, $upordown, $gencode_ref, \ $log_fh);
  print_from_hash_ref_to_fh($introns_up_overhang_ref, \ $log_fh);
}
# get regulatory regions up- or downstram from the gene of interest, depending on + or - strand
if ($strand_tf eq "+") {
  my $upordown = "up";
  print $log_fh "upstream +\n";
  $upstream_introns_ref = up_or_down_reg_regions($lower_boundary, $tss, $chr_name, $upordown, $gencode_ref, \ $log_fh);
  print_from_hash_ref_to_fh($upstream_introns_ref, \ $log_fh);
}
if ($strand_tf eq "-") {
  my $upordown = "down";
  print $log_fh "downstream -\n";
  $downstream_introns_ref = up_or_down_reg_regions($upper_boundary, $tss, $chr_name, $upordown, $gencode_ref, \ $log_fh);
  print_from_hash_ref_to_fh($downstream_introns_ref, \ $log_fh);
}
# get sequences of regulatory regions from fasta genome and print them to the output.
my @all_intron_hash_refs = (\%introns, \%introns_comp, $introns_down_overhang_ref, $introns_up_overhang_ref, $upstream_introns_ref, $downstream_introns_ref);
foreach my $ref (@all_intron_hash_refs) {
  unless (!$ref) {
    my %int_hash = %{$ref};
    my $fasta_hash_ref = get_sequences_in_hash($ref, $strand_tf, $chr_name, $isoform_id, \%chromosome_seqs);
    to_fasta($fasta_hash_ref, \ $out_regions_fh);
  }
}
}

```

```

    }
}
## end of main ##

## ## subroutines ##
sub get_sequences_in_hash {
    my ($intron_hash_ref, $strand, $chr_name, $isoform_name, $chr_seqs_ref) = @_;
    my %introns = %{$intron_hash_ref};
    my %chr_seqs = %{$chr_seqs_ref};
    my %output_fasta_hash;
    my $fasta_complete_chr = $chr_seqs{$chr_name};
    foreach my $intron_st (keys %introns) {
        my $name_and_reg_id = $isoform_name . " " . $chr_name . " " . $intron_st . "-" . $introns{$intron_st};
        if ($strand eq "+") {
            my $fasta_subseq = $fasta_complete_chr->subseq($intron_st, $introns{$intron_st});
            $output_fasta_hash{$name_and_reg_id} = $fasta_subseq;
        }
        if ($strand eq "-") {
            my $fasta_trunc = $fasta_complete_chr->trunc($intron_st, $introns{$intron_st});
            my $reverse_complement = $fasta_trunc->revcom;
            my $fasta_subseq = $reverse_complement->seq;
            $output_fasta_hash{$name_and_reg_id} = $fasta_subseq;
        }
    }
    return \%output_fasta_hash;
}

sub to_fasta {
    my ($id_seq_hash_ref, $fh_ref) = @_;
    my %id_seq_hash_ref = %{$id_seq_hash_ref};
    my $fh = ${$fh_ref};
    foreach my $id (keys %id_seq_hash_ref) {
        print $fh ">$id\n";
        while (my $chunk = substr($id_seq_hash_ref{$id}, 0, 80, "")) {
            my $formatted_seq;
            print $fh $formatted_seq .= "$chunk\n";
        }
    }
}

sub print_from_hash_ref_to_fh {
    my ($reg_regions_ref, $fh_ref) = @_;
    my $fh = ${$fh_ref};
    my %reg_regions = %{$reg_regions_ref};
    foreach my $reg_region_start (sort keys %reg_regions) {
        print $fh "$reg_region_start .. $reg_regions{$reg_region_start}\n";
    }
}

sub up_or_down_reg_regions {
    my ($boundary, $end_of_tfoi, $chr, $supordown, $annotation_ref, $log_ref) = @_;
    my $exons_ref = get_exon_coordinates($boundary, $end_of_tfoi, $chr, $supordown, $annotation_ref, $log_ref);
    my %exons = %{$exons_ref};
    my $stretches_covered_by_exons_ref = make_exon_stretches($boundary, $end_of_tfoi, $supordown, $exons_ref, $log_ref);
    my %stretches_covered_by_exons = %{$stretches_covered_by_exons_ref};
    my $introns_ref = get_introns($boundary, $end_of_tfoi, $supordown, $stretches_covered_by_exons_ref, $log_ref);
    return %introns_ref;
}

sub get_exon_coordinates {
    my ($boundary, $end_of_tfoi, $chr, $supordown, $annotation_ref, $log_ref) = @_;
    my @annotation = @{$annotation_ref};
    my %selected_exons;
    foreach my $line (@annotation) {
        if ($line =~ /$chr/) {
            my @split_line = split(/\s/, $line);
            if ($split_line[2] eq "exon") {
                my $current_exon_start = $split_line[3];
                my $current_exon_end = $split_line[4];
            }
        }
    }
}

```

```

#
# find all exons, regardless if from another gene or another isoform,
# whose exon starts or ends falls between the boundary and tf of interest
if ($upordown eq "up") {
    if ($current_exon_end > $boundary) {
        if ($current_exon_end < $end_of_tfoi) {
            $selected_exons{$current_exon_start} = $current_exon_end;
        }
    }
    if ($current_exon_start > $boundary) {
        if ($current_exon_start < $end_of_tfoi) {
            $selected_exons{$current_exon_start} = $current_exon_end;
        }
    }
}
if ($upordown eq "down") {
    if ($current_exon_end < $boundary) { #
        if ($current_exon_end > $end_of_tfoi) { #
            $selected_exons{$current_exon_start} = $current_exon_end;
        }
    }
    if ($current_exon_start < $boundary) { #
        if ($current_exon_start > $end_of_tfoi) { #
            $selected_exons{$current_exon_start} = $current_exon_end;
        }
    }
}
}
}
return \%selected_exons;
}

sub make_exon_stretches {
    my ($boundary, $end_of_tfoi, $upordown, $exons_ref, $log_ref) = @_;
    my $log = ${$log_ref};
    my %exon_stretches;
    my %exons = ${$exons_ref};
    unless (!%exons) {
        my $number_of_exons = keys %exons;
        my $count = 1;
        my ($current_longest_end, $current_shortest_start);
        foreach my $exon_start (sort keys %exons) {
            my ($start, $end);
            if ($upordown eq "up") {
                if ($exons{$exon_start} < $end_of_tfoi) {
                    $end = $exons{$exon_start};
                }
                else {
                    $end = $end_of_tfoi;
                }
                if ($exon_start < $boundary) {
                    $start = $boundary;
                }
                else {
                    $start = $exon_start;
                }
            }
            if ($upordown eq "down") {
                if ($exon_start > $end_of_tfoi) {#
                    $start = $exon_start;#
                }
                else {
                    $start = $end_of_tfoi;#
                }
                if ($exons{$exon_start} < $boundary) {#
                    $end = $exons{$exon_start};#
                }
                else {
                    $end = $boundary;#
                }
            }
        }
    }
}

```

```

    }
    print $log_fh "mex: $start..$end\n";
    if ($count == 1) {
        $current_longest_end = $end;
        $current_shortest_start = $start;
        if ($number_of_exons == 1) {
            $exon_stretches{$current_shortest_start} = $current_longest_end;
        }
    }
    else {
        if ($start <= $current_longest_end) {
            if ($end > $current_longest_end) {
                $current_longest_end = $end; # old end is assigned new value, exonic sequence is extended
            }
            if ($count == $number_of_exons) {
                $exon_stretches{$current_shortest_start} = $current_longest_end;
            }
        }
        else {
            $exon_stretches{$current_shortest_start} = $current_longest_end;
            $current_shortest_start = $start;
            $current_longest_end = $end;
            if ($count == $number_of_exons) {
                $exon_stretches{$current_shortest_start} = $current_longest_end;
            }
        }
    }
    $count++;
}
return \%exon_stretches;
}

sub get_introns {
    my ($boundary, $end_of_tfoi, $supordown, $exon_stretches_ref, $log_ref) = @_;
    my $log = ${$log_ref};
    if ($supordown eq "down") {
        my $old_boundary = $boundary;
        $boundary = $end_of_tfoi;
        $end_of_tfoi = $old_boundary;
    }
    my %stretches_covered_by_exons = %{ $exon_stretches_ref };
    my ($intron_start, $intron_end);
    my $counter = 1;
    my %intron_stretches;
    my $number_of_exonic_stretches = keys %stretches_covered_by_exons;
    print $log_fh "exonic stretches: $number_of_exonic_stretches\n";
    foreach my $exonic_stretch_start (sort keys %stretches_covered_by_exons) {
        my $exonic_stretch_end = $stretches_covered_by_exons{$exonic_stretch_start};
        print $log_fh "stretches: $exonic_stretch_start..$exonic_stretch_end\n";
        if ($counter == 1) {
            if ($exonic_stretch_start == $boundary) {
                unless ($exonic_stretch_end == $end_of_tfoi) {
                    $intron_start = $exonic_stretch_end + 1; # assign value to intron start for next round
                    if ($counter == $number_of_exonic_stretches) {
                        if ($supordown eq "up") {
                            $intron_stretches{$intron_start} = $end_of_tfoi - 1;
                        }
                        if ($supordown eq "down") {
                            $intron_stretches{$intron_start} = $end_of_tfoi;
                        }
                    }
                }
            }
            else {
                $counter++;
            }
        }
    }
    }
    else {

```

```

        if ($upordown eq "up") {
            $intron_start = $boundary;
        }
        if ($upordown eq "down") {
            $intron_start = $boundary + 1;
        }
        $intron_end = $exonic_stretch_start - 1;
        $intron_stretches{$intron_start} = $intron_end;
        $intron_start = $exonic_stretch_end + 1; # assign value to intron start for next round
        if ($counter == $number_of_exonic_stretches) {
            unless ($exonic_stretch_end == $end_of_tfoi) {
                if ($upordown eq "up") {
                    $intron_stretches{$intron_start} = $end_of_tfoi - 1;
                }
                if ($upordown eq "down") {
                    $intron_stretches{$intron_start} = $end_of_tfoi;
                }
            }
        }
        $counter++;
    }
}
else {
    $intron_end = $exonic_stretch_start - 1;
    $intron_stretches{$intron_start} = $intron_end; # $intron_start got its value during last round.
    unless ($exonic_stretch_end == $end_of_tfoi) {
        $intron_start = $exonic_stretch_end + 1; # assign new value to intron_start for next round;
    }
    if ($counter == $number_of_exonic_stretches) {
        unless ($exonic_stretch_end == $end_of_tfoi) {
            if ($upordown eq "up") {
                $intron_stretches{$intron_start} = $end_of_tfoi - 1;
            }
            if ($upordown eq "down") {
                $intron_stretches{$intron_start} = $end_of_tfoi;
            }
        }
    }
    $counter++;
}
}
if (!%stretches_covered_by_exons) {
    $intron_stretches{$boundary} = $end_of_tfoi - 1;
}
return \%intron_stretches;
}
# end of subroutines # #

```

```

assign_cell-type-specific_interactions.pl

#!/usr/bin/perl

use strict;
use warnings;
use files;

=head
Purpose: This script assigns interactions and interaction types between TF based on the results of expression analysis and DNase seq footprint analysis.
Input: List of expressed TF with ENST/ENSG identifiers (in /scratch/todtova/transcriptome_pipeline/cufflinks/<cell-type>/cell-type_ENSTENSG_expressed.
Output: File listing each TF-TF interaction and its type; File listing each tf and its incoming regulators + interaction type.
=cut

my $open = "<";
my $write = ">";
my $add = ">>";

my $cell_type = $ARGV[0];

#TOP section
#1) Open file that contains mapping of ENST and ENSG tags to isoform names. Only protein coding isoforms (and the ENSG tags of all tfoi mapped to "TF-000") are stored in this file.
my $enst_tag_conversion_list = "reg_regions/collection_ENST_protiso_ENSG_all.txt";
my $enst_conv_fh = files::write_or_open($enst_tag_conversion_list, $open);
my @enst_conv = <$enst_conv_fh>;
my %conv_hash; # will contain all isoforms enst tags and their name
foreach my $line (@enst_conv) {
    my @split_line = split(/\t/, $line);
    $conv_hash{$split_line[0]} = $split_line[1];
}

# 2) Read a file containing HGNC gene symbol to uniprot ID mapping (for conversion of uni tf names in reference net and in the footprint files).
# HGNC2uniprot.txt was created using biomaRt, and adding 3 factor manually by hand, namely ZNF238, NR2E3 and FOXD1.
my $hgnc = "../transcriptome_pipeline/HGNC2uniprot.txt";
my $hgnc_fh = files::write_or_open($hgnc, $open);
my @hgnc_uni = <$hgnc_fh>;
my %hgnc_mapping;

foreach my $hgnc_line (@hgnc_uni) {
    unless ($hgnc_line =~ /^#/){
        my @split_hgnc = split(/\t/, $hgnc_line);
        my $hgnc_symbol = $split_hgnc[0];
        my @split2 = split(/\t/, $split_hgnc[1]);
        my $uni_ID = $split2[0];
        $hgnc_mapping{$uni_ID} = $hgnc_symbol;
    }
}

# 3) Create Set 1 (Reference net (only motif-interactions))
my $reference_from_only_motifs = "/scratch/todtova/tf_regions/motif_overlaps/motif_locations_protiso.bed.uniq.bedsorted.noChrY";
my $ref_fh = files::write_or_open($reference_from_only_motifs, $open);
my @reference_int = <$ref_fh>;
my ($ref_net_ref, $ref_net_regulated_ref) = get_interactions_from_file_with_fimo_motifs(@reference_int, \%hgnc_mapping, 0);
my %ref_net = %{ $ref_net_ref };

# 4) Create Set 2a - the set of expressed nodes: Open file with expressed isoform enst-/ensg-tags and get the names of the corresponding isoforms from %conv_hash.
my $expr_isoforms = "/scratch/todtova/transcriptome_pipeline/cufflinks/expression_analysis/" . $cell_type . "_expressed.txt";
my $expr_iso_fh = files::write_or_open($expr_isoforms, $open);
my @expr_isos = <$expr_iso_fh>;

my %expr_iso_list; # will contain only expressed isoform enst tags and their name
foreach my $expr_enst (@expr_isos) {
    chomp $expr_enst;
    my $iso_name = $conv_hash{$expr_enst};
    chomp $iso_name;
    $expr_iso_list{$iso_name} = $expr_enst;
}

my %expr_tf_to_iso_names; # associate a tf with all its expressed isoforms (union of expr isoforms = node (for expressed tf)

```

```

foreach my $transcript (keys %expr_iso_list) {
    chomp $transcript;
    my $tf;
    my @spl = split(/\-\\d\\d\\d/, $transcript); # all isoforms must be in form "TF-001"
    $tf = $spl[0];
    $expr_tf_to_iso_names{$tf} .= " $transcript"; #TF1 -> TF1-001 TF1-002 TF1-003
}

#Create Set 2b - a set of not expressed tfs.
my @excluded = ("NR2E3"); # not recognized or LOWDATA in cufflinks, for mouse: ("Foxc2", "Foxf1", "Hoxd4", "Irf8", "Sox2");
my %excluded_genes;
foreach my $ex (@excluded) {
    $excluded_genes{$ex} = "";
}
my $tf_list = "HGNC symbols humantf.sorted.txt";
my $tf_list_fh = files::write_or_open($tf_list, $open);
my @all_tfs = <$tf_list_fh>;
my %nexpr_tfs;
my ($count, $count2) = 0;
foreach my $tf_gene (@all_tfs) {
    chomp $tf_gene;
    unless (exists $excluded_genes{$tf_gene}) {
        unless (exists $expr_tf_to_iso_names{$tf_gene}) {
            $nexpr_tfs{$tf_gene} = "";
        }
    }
}

# 5) Create Set 3 (all footprinted nodes) and Set 4 (reg interactions on motifs overlapped by footprints)
my $fps = "/scratch/todtova/tf_regions/motif_overlaps/cell-specific_fps/" . $cell_type . "/" . $cell_type . ".bedmap.neph.onlymapped"; # $ARGV[1]; ##SK-N-SH_RA/SK-N-SH_RA.bedmap.neph.onlymapped";
my $fps_fh = files::write_or_open($fps, $open);
my @footprints = <$fps_fh>;

#first put all footprints in a hash
my %resolve_fp;
foreach my $fp_line (@footprints) {
    my @blocks = split(/;/, $fp_line); # first block is motif location, second to ... are footprints that overlap motif
    my $last_index_of_overlapped_fp = scalar(@blocks) - 1;
    for my $i (1..$last_index_of_overlapped_fp) { # some motif-lines are overlapped by more than one fp
        my @fp_tabs = split(/\t/, $blocks[$i]); # split each footprint record
        my $footprint = $fp_tabs[0] . "\t" . $fp_tabs[1] . "\t" . $fp_tabs[2]; #remeber footprints as chr \t start \t end
        $resolve_fp{$footprint} = "";
    }
}

#now associate each fp in the hash with a hash of the motifs that it overlaps. Originally, motifs are in the form tf->tf-iso. Now, through splicing away the isoform id from the target, and
through
#storing in a hash, pseudo-unresolvedness is removed (meaning when one fp overlaps multiple motifs in which the same regulator regulates different isoforms of the same target tf,
#this is counted as one motif for regulator-> regulated tf).
foreach my $fp (keys %resolve_fp) {
    my %motifs_in_fp;
    foreach my $line (@footprints) {
        if ($line =~ $fp) {
            my @tabs = split(/\t/, $line);
            my $iso_motif = $tabs[3];
            my @split_iso_from_motif = split(/-\\d\\d\\d/, $iso_motif);
            my $motif = $split_iso_from_motif[0];
            $motifs_in_fp{$motif} = ""; #multiple motifs with the same regulator but different regulated isoforms of the same target tf are collapsed by putting them into hash
        }
    }
    $resolve_fp{$fp} = \%motifs_in_fp;
}

#if the hash with the regulator-> regulated tf motifs has more than one entry, the footprint is unresolved.
foreach my $footpr (keys %resolve_fp) {
    my $number_of_overlapped_motifs = keys %{ $resolve_fp{$footpr} };
    if ($number_of_overlapped_motifs > 1) {

```

```

        $resolve_fp{$footpr} = "nrs";
    }
    else {
        $resolve_fp{$footpr} = "res";
    }
}

#now create set 3 (footprinted nodes) and set 4 (regulatory interactions on motifs overlapped by footprints.
my ($fp_ref, $footprinted_tf_ref) = get_interactions_from_file_with_fimo_motifs(\@footprints, \%hgnc_mapping, \%resolve_fp);
my %fp = %{ $fp_ref };
my %footprinted_tf = %{ $footprinted_tf_ref };

# MAIN section
# 6) Go through interactions and find interactions of a) expressed b) not-expressed tf.
my %tf_interactions;
foreach my $tf_int (keys %ref_net) {
    my @split_int = split(/\t/, $tf_int);
    my ($interaction_regulator, $interaction_regulated) = ($split_int[0], $split_int[1]);
    #a) go through all expressed isoforms and get their interactions
    if (exists $expr_tf_to_iso_names{$interaction_regulator}) { # check if regulator in int is expressed
        chomp $interaction_regulated;
        my $expr_tf = $interaction_regulator;
        my $first_neighbour_tf = $interaction_regulated;
        if (exists $expr_tf_to_iso_names{$first_neighbour_tf}) { # check if first neighbour (= regulated) is among expressed
            if (exists $fp{$tf_int}) {
                my $ia = $expr_tf . "\t" . $first_neighbour_tf . "\t+d" . "\t$fp{$tf_int}";
                $tf_interactions{$ia} = "";
            }
        }
        #if the target is not expressed, and it has a footprinted motif, it is also an inhibition.
        if (exists $nexpr_tfs{$first_neighbour_tf}) {
            if (exists $fp{$tf_int}) {
                my $ia = $expr_tf . "\t" . $first_neighbour_tf . "\t-d" . "\t$fp{$tf_int}";
                $tf_interactions{$ia} = "";
            }
        }
    }
    #b) go through all not expressed tfs and get their interactions
    elsif (exists $nexpr_tfs{$interaction_regulator}) { # check if regulator in int is not expressed
        my $nexpr = $interaction_regulator;
        chomp $interaction_regulated;
        my $first_neighbour_tf = $interaction_regulated;
        if (exists $expr_tf_to_iso_names{$first_neighbour_tf}) { # check if first neighbour (= regulated) is among expressed
            unless (exists $footprinted_tf{$first_neighbour_tf}) {
                my $ia = $nexpr . "\t" . $first_neighbour_tf . "\t-i" . "\trnc";
                $tf_interactions{$ia} = "";
            }
            #if however the target has a footprint in its regulatory region, the regulatory effect of the non-expression of the current tf is uncertain.
            #an uncertain edge is attributed.
            if (exists $footprinted_tf{$first_neighbour_tf}) {
                my $ia = $nexpr . "\t" . $first_neighbour_tf . "\t0u" . "\trnc";
                $tf_interactions{$ia} = "";
            }
        }
        if (exists $nexpr_tfs{$first_neighbour_tf}) {
            unless (exists $footprinted_tf{$first_neighbour_tf}) {
                my $ia = $nexpr . "\t" . $first_neighbour_tf . "\t+i" . "\trnc";
                $tf_interactions{$ia} = "";
            }
            if (exists $footprinted_tf{$first_neighbour_tf}) {
                my $ia = $nexpr . "\t" . $first_neighbour_tf . "\t0u" . "\trnc";
                $tf_interactions{$ia} = "";
            }
        }
    }
}

}

# OUTPUT section

```

```

# #7) Prepare node-hash for node-outfile. This file will list all interactions in the following form: regulated tf/node, interaction type, resolvedness of footprint, combinedness, regulating
factor (separator: " ")
# First, all regulated nodes are stored in a hash.
my %regulated_node_hash;
foreach my $ia (sort keys %tf_interactions) {
    my $ia_to_split = $ia;
    my @split_outgoing = split(/\t/, $ia_to_split);
    my $receiving_tf = $split_outgoing[1];
    $regulated_node_hash{$receiving_tf} = "";
}

# Open node incomings outfile and expressed tf outfile
my $outprefix = "inputs_to_active_net/" . $Cell_type;
my $node_file = $outprefix . "_node_incomings_list";
my $node_fh = files::write_or_open($node_file, $write);

my $expressed_tf_out = $outprefix . "_expressed_list";
my $expr_out_fh = files::write_or_open($expressed_tf_out, $write);

# Then, the interactions are parsed for each regulated node. Each regulated node is associated with all its regulators and their type of interaction and the resolvedness of the pf.
foreach my $regulated_node (keys %regulated_node_hash) {
    my @regulators_list;
    foreach my $ia (keys %tf_interactions) {
        my @split_outgoing = split(/\t/, $ia);
        my $receiving_tf = $split_outgoing[1];
        my $regulating_tf = $split_outgoing[0];
        my $type_of_ia = $split_outgoing[2];
        my $resolvedness = $split_outgoing[3];
        if ($receiving_tf eq $regulated_node) {
            my $regulator_int = $type_of_ia . "_" . $resolvedness . "_" . $regulating_tf;
            push (@regulators_list, $regulator_int);
        }
    }
    $regulated_node_hash{$regulated_node} = \@regulators_list;
}

# Now, all regulated nodes are parsed again and the number of direct regulations is counted. If there is more than one direct interaction, all direct interactions are termed 'com' (for
# combined effect. If there is only one direct interaction, 'sep' is assigned (separate effect). Indirect and uncertain interactions are termed 'cnc' (combinedness not considered), as they are
# all combined.
foreach my $target_node (keys %regulated_node_hash) {
    my $count_if_combined_effect = 0;
    foreach my $regulation1 (@{$regulated_node_hash{$target_node}}) {
        if ($regulation1 =~ "d") {
            $count_if_combined_effect++;
        }
    }
    if ($count_if_combined_effect > 1) {
        foreach my $regulation2 (@{$regulated_node_hash{$target_node}}) {
            my @spl_reg = split(/_/, $regulation2);
            if ($regulation2 =~ "d") {
                $regulation2 = $spl_reg[0] . "_" . $spl_reg[1] . "_com" . "_" . $spl_reg[2];
            }
            else { # for indirect and uncertain interactions, the effect is always combined
                $regulation2 = $spl_reg[0] . "_" . $spl_reg[1] . "_com" . "_" . $spl_reg[2];
            }
        }
    }
    else {
        foreach my $regulation3 (@{$regulated_node_hash{$target_node}}) {
            my @spl_reg = split(/_/, $regulation3);
            if ($regulation3 =~ "d") {
                $regulation3 = $spl_reg[0] . "_" . $spl_reg[1] . "_sep" . "_" . $spl_reg[2];
            }
            else {
                $regulation3 = $spl_reg[0] . "_" . $spl_reg[1] . "_com" . "_" . $spl_reg[2];
            }
        }
    }
}
}

```

```

# printing to node- and expressed-outfile, preparing hash for edge-outfile
my %regulator_node_hash;
foreach my $node (sort keys %regulated_node_hash) {
    my $state;
    if (exists $expr_tf_to_iso_names{$node}) {
        $state = 1;
    }
    if (exists $nexpr_tfs{$node}) {
        $state = 0;
    }
    print $expr_out_fh $node, "\t $state\n";
    foreach my $regulation (sort @{$regulated_node_hash{$node}}) {
        print $node_fh $node, "_", $regulation, "\n";
    }
}

# as a last thing, the node_incomings_output is rearranged and sorted by regulating tf in bash, creating a list of edges (regulating_tf->target descriptions) (tab-delimited file!)
my $edge_file = $outprefix . "_edge_type_list";
`sed 's/_/\t/g' $node_file | awk '{print \$5 "\t\t" \$1 "\t\t" \$2 "\t\t" \$3 "\t\t" \$4}' | sort -k1 > $edge_file`;

# this function receives 1) a fimo output file produced by 'make_genomic_coord_from_fimo.pl' - this can be either 1.1) the overlap of this reference net with footprints or 1.2) the reference
# net of potential interactions.
# 2) a hash to convert uniprot-names to hgnc names
# 3) a variable holding either a hash reference with the resolvedness of fp (for 1.1) or a variable saying '0' (for 1.2)
sub get_interactions_from_file_with_fimo_motifs {
    my ($file_ref, $hgnc_mapping_ref, $fp_resolvedness) = @_;
    my @file = @{$file_ref};
    my %hgnc_mapping = %{$hgnc_mapping_ref};
    my %resolve_fp;
    #-for 1.1
    unless ($fp_resolvedness == 0) {
        %resolve_fp = %{$fp_resolvedness};
    }
    #-----
    my (%interactions, %regulated);
    foreach my $line (@file) {
        my @spl = split(/\t/, $line);
        my $motif = $spl[3];
        my @split_interaction = split(/\>/, $motif);
        my $fp_regulator = $split_interaction[0];
        my $fp_regulated_hgnc = $split_interaction[1]; # regulated isoform is already in hgnc format
        my @splice_away_isoid = split(/-\d\d\d\d/, $fp_regulated_hgnc);
        my $regulated_tf = $splice_away_isoid[0];
        $regulated{$regulated_tf} = ""; # regulated tf is stored as footprinted
        my ($fp_regulator_hgnc, $interaction, $resolvedness);

        #-for 1.1: for each line, determine if the motif is resolved (based on whether is is overlapped by an "exclusive" footprint
        unless ($fp_resolvedness == 0) {
            my @blocks = split(/:/, $line); # each line has the form: motif-block; fp-block1; ...; fp_block$j
            my $number_of_ovlp_fp = scalar(@blocks) - 1;
            for my $j (1..$number_of_ovlp_fp) { # for each fp-block
                my @tabs = split(/\t/, $blocks[$j]);
                my $footprint = $tabs[0] . "\t" . $tabs[1] . "\t" . $tabs[2]; # fp = chr start end
                if ($number_of_ovlp_fp == 1) { # if there is only one fp-block
                    $resolvedness = $resolve_fp{$footprint};
                }
                else { # if the motif is overlapped by more than one fp, all 'resolvednesses' are collected
                    $resolvedness .= $resolve_fp{$footprint};
                }
            }
            # in this step, resolvedness is searched for an entry 'nrs' (not resolved) and is changed to 'nrs' if it holds any fp that is not resolved (from
            concatenation/collection in previous loop!)
            # this is done because of the possibility of motifs that are overlapped by multiple fp - if one of the fp is unresolved, the motif is termed unresolved
            if ($resolvedness =~ "nrs") {
                $resolvedness = "nrs";
            }
            else {

```

```

        $resolvedness = "res";
    }
}
#-----
if ($fp_regulator =~ /\+/) { # for combined motif-models. eg. GABP1+GABP2
    my @combined_tf_model = split /\+/, $fp_regulator;
    foreach my $tf (@combined_tf_model) {
        $fp_regulator_hgnc = $hgnc_mapping{$tf};
        $interaction = $fp_regulator_hgnc . "\t" . $regulated_tf;
        $interactions{$interaction} = "";
        #for 1.1: interactions get their resolvedness from motifs. if multiple motifs with different resolvedness exist for one interaction, it is unresolved with
        at least one unresolved motif.
        unless ($fp_resolvedness == 0) {
            if (exists $interactions{$interaction}) {
                $interactions{$interaction} .= $resolvedness;
            }
            else {
                $interactions{$interaction} = $resolvedness;
            }
        }
    }
}
else {
    $fp_regulator_hgnc = $hgnc_mapping{$fp_regulator};
    $interaction = $fp_regulator_hgnc . "\t" . $regulated_tf;
    $interactions{$interaction} = "";
    #for 1.1: as for combined tf-models.
    unless ($fp_resolvedness == 0) {
        if (exists $interactions{$interaction}) {
            $interactions{$interaction} .= $resolvedness;
        }
        else {
            $interactions{$interaction} = $resolvedness;
        }
    }
}
#-----
}
#for 1.1: unresolvedness is assigned to interactions with at least one unresolved motif
unless ($fp_resolvedness == 0) {
    foreach my $ia (keys %interactions) {
        my $res = $interactions{$ia};
        if ($res =~ "nrs") {
            $res = "nrs";
        }
        else {
            $res = "res";
        }
    }
}
#-----
return (\%interactions, \%regulated);
}

```

```

active_subnet.pl

#!/usr/bin/perl
use strict;
use warnings;
use files;
use Data::Dumper;

=head
Purpose: Derive active subnet from the cell-type specific net in which all interactions have been assigned. The active subnet contains:
    1) Interactions between expressed tf
    2) Interactions from expressed tf to non-expressed and vice-versa (= first neighbour interactions)
    3) Interactions between first neighbours
    1 + 2 + 3 = core active net
    4) Chains of interactions starting at the core, going across non-expressed tf and returning to the core
    4 = chain interactions
Input: <cell_type>_expressed_list (derived from RNA-seq expression calls, see /scratch/todtova/transcriptome_pipeline/cufflinks/expression_analysis),
      <cell_type>_edge_type_list (file produced by assign_tf_specific_interactions.pl)
Output: a) node.csv, edge.csv b) pajek.net
=cut

# TOP section
my $open = "<";
my $write = ">";
my $add = ">>";

my $cell_type = $ARGV[0];
my $indir = "inputs_to_active_net/";
my $expressed_file = $indir . $cell_type . "_expressed_list";
my $edge_file = $indir . $cell_type . "_edge_type_list";

my $expr_fh = files::write_or_open($expressed_file, $open);
my $edge_fh = files::write_or_open($edge_file, $open);
my @edges_list = <$edge_fh>;

#1) Create set of expressed tf from '_expressed_list'
my %expr_tf;
foreach (<$expr_fh>) {
    # TF expression is described in the form: TF-name\s<1/0>. all lines that contain a whitespace followed by "1" mark expressed TF.
    if ($_ =~ /\s1/) {
        my @split_line = split(/\s/, $_);
        my $expressed_tf = $split_line[0];
        $expr_tf{$expressed_tf} = "1";
    }
}

#2) Create set of interactions from '_edge_type_list' file
my %interactions;
foreach $_ (@edges_list) {
    chomp $_;
    my @split_int = split(/\t/, $_);
    # first and second columns contain regulator and target
    my $regulator = $split_int[0];
    my $target = $split_int[1];
    my $int = $regulator . "\t" . $target;

    # next three columns contain information describing the interaction: interaction type, if the footprint that was considered to assign the interaction type is resolved,
    # and last, if the effect of the interaction is carried out by the regulator alone, or together, in combination with other regulators.
    my $int_type = $split_int[2];
    my $resolvedness = $split_int[3];
    my $combinedness = $split_int[4];
    my $description = $int_type . "\t" . $resolvedness . "\t" . $combinedness;
    $interactions{$int} = $description;
}

#MAIN section
=head
3) Get interactions for the 'core net':

```

```

a) interactions going out from expressed tf (-d to not expressed, +d between expressed tf)
b) interactions going out from not expressed to expressed tf (-i, 0u where target is expressed)
First neighbours are by definition not expressed nodes, that are regulated by expressed tf or regulate expressed tf. They are considered as a part of the core net.
=cut
my %core_net_interactions;
my %first_neighbours_going_out;
my %first_neighbours_going_in;
foreach $_ (keys %interactions) {
    # all interactions with direct evidence are part of the core net, because they go out from expressed tf
    if ($interactions{$_} =~ "d") {
        $core_net_interactions{$_} = $interactions{$_};
        # -d interactions are interactions to FNO (first neighbours going out)
        # targets of -d interactions are marked as FNO
        if ($interactions{$_} =~ /\-d/) {
            my @split_int = split(/\t/, $_);
            my $target = $split_int[1];
            $first_neighbours_going_out{$target} = "";
        }
    }
    if ($interactions{$_} =~ /\-i/) {
        # -i interactions go out from nexpr tf to expr tf - their regulators are first neighbours going in (FNI)
        # regulators of -i are stored as FNI
        $core_net_interactions{$_} = $interactions{$_};
        my @split_int = split(/\t/, $_);
        my $regulator = $split_int[0];
        $first_neighbours_going_in{$regulator} = "";
    }
    if ($interactions{$_} =~ "0u") {
        # a part of the 0u interactions also goes out from FNI to expressed tf
        # if the target of a 0u interaction is expressed, the regulator is marked as an FNI
        my @split_int = split(/\t/, $_);
        my $target = $split_int[1];
        my $regulator = $split_int[0];
        if (exists $expr_tf{$target}) {
            $core_net_interactions{$_} = $interactions{$_};
            $first_neighbours_going_in{$regulator} = "";
        }
    }
}

my %all_first_neighbours = (%first_neighbours_going_in, %first_neighbours_going_out);

=head
4) Get all interactions between first neighbours
1) Create set of potential chain interactions, which will run from FNO to non-expressed tf, between non-expressed tf, and from non-expressed tf to FNI.
   These interactions are +i or 0u by default, just as the interactions between first neighbours are.
2) Add interactions between first neighbours to the core net, and then remove these interactions from the set of potential chain interactions.
   When looking for chain interactions between non-expressed nodes, you will be only interested in those that do not go directly from FN to FN
=cut
# create a hash that will associate each unexpressed node with its #####unexpressed##### targets
my %potential_chain_nodes;
# initially, this will contain all +i interactions and 0u interactions with unexpressed targets, but then interactions directly from FN to FN will be removed
my %potential_chain_interactions;

foreach (keys %interactions) {
    if ($interactions{$_} =~ /\+i/) {
        $potential_chain_interactions{$_} = $interactions{$_}; #make keys for +i int + get description value
        my @tabs = split(/\t/, $_);
        my $regulator = $tabs[0];
        my $target = $tabs[1];
        #store both regulators and targets from +i as potential chain nodes
        $potential_chain_nodes{$regulator} = "";
        $potential_chain_nodes{$target} = ""; #
    }
    if ($interactions{$_} =~ "0u") {
        $potential_chain_interactions{$_} = $interactions{$_};
        my @tabs = split(/\t/, $_);
        my $regulator = $tabs[0];
        my $target = $tabs[1];
    }
}

```

```

        # we are only interested in 0u interactions that have an unexpressed target (regulator of 0u is by default unexpressed)
        unless (exists $expr_tf($target)) {
            $potential_chain_nodes($regulator) = "";
            $potential_chain_nodes($target) = "";
        }
    }
}

# now go through %potential chain interactions (contains all +i and 0u interactions (derived in previous block)), and seperate those interactions that go from FN to FN
my %int_between_first_neighbours;
foreach my $pot_int (Keys %potential_chain_interactions) {
    my @split_pot_int = split(/\t/, $pot_int);
    my $regulator = $split_pot_int[0];
    my $regulated = $split_pot_int[1];
    # if interaction is regulated by a first neighbour and the target is also a first neighbour -> store as interaction between first neighbours
    if (exists $all_first_neighbours($regulator)) {
        if (exists $all_first_neighbours($regulated)) {
            $int_between_first_neighbours($pot_int) = $potential_chain_interactions($pot_int); # assign interactions from potential chain set to first neighbour set
            # delete interactions between FN from potential chain set
            delete $potential_chain_interactions($pot_int);
        }
    }
}

# Here, interactions between first neighbours are added to the core net
@core_net_interactions{keys %int_between_first_neighbours} = values %int_between_first_neighbours;

# Now, go through potential chain interactions and get the targets for each potential chain node.
foreach (keys %potential_chain_nodes) {
    my %targets;
    foreach my $int (keys %potential_chain_interactions) {
        if ($int =~ /^$_\t/) { # only look at interactions regulated by the current potential nexpr regulator $_
            if ($potential_chain_interactions{$int} =~ /\+i/) {
                my @spl_int = split(/\t/, $int);
                my $target = $spl_int[1];
                #save all targets in target hash
                $targets{$target} = "";
            }
            if ($potential_chain_interactions{$int} =~ /0u/) {
                my @spl_int = split(/\t/, $int);
                my $target = $spl_int[1];
                $targets{$target} = "";
            }
        }
    }
    #associate hash of all targets with current regulator $_
    $potential_chain_nodes{$_} = \%targets;
}

=head
#5) Get chains of interactions starting at the core, going across non-expressed tf and returning to the core.
    These chains will only go along +i or 0u edges (as stored in potential chain interactions)
    They have to start on a first_neighbour_going_out and to end on first_neighbour_going_in.

=cut
my %int_from_chains;
my %dead_ends;
foreach my $pot_cyc_int (keys %potential_chain_interactions) {
    my @split_pot_int = split(/\t/, $pot_cyc_int);
    my $regulator = $split_pot_int[0];
    my $regulated = $split_pot_int[1];
    # find potential chain interactions, in which the regulator is a FNO, and start looking for chain int from there
    if (exists $first_neighbours_going_out($regulator)) {
        my (%remember_int, %seen, %seen_in_rounds, %remove_int);
        my $loopcount = 0;
        $remember_int{$pot_cyc_int} = $potential_chain_interactions{$pot_cyc_int};
        $seen{$regulator} = "";
        $seen_in_rounds{$loopcount} = \%seen;
        # start to look for chain interactions using the function find_chains (see below for a description of the function arguments)
        if (exists $first_neighbours_going_in($regulated)) {

```

```

        $int_from_chains{$spot_cyc_int} = "";
    }
    else {
        my ($chain_int_ref, $remove_ref, $dead_end_ref, $FNI_ref) = find_chains(\%potential_chain_nodes, \%remember_int, \%first_neighbours_going_in, $regulated,
        \%seen_in_rounds, $loopcount, \%remove_int, \%dead_ends);
        my %chain_ints = %{ $chain_int_ref };
        if (keys %chain_ints > 1) {
            @int_from_chains{keys %chain_ints} = values %chain_ints;
            print "chain int found.\n";
        }
        else {
            #print "no int from chains for FNO $regulator\n\n";
        }
        @first_neighbours_going_in{keys %{ $FNI_ref }} = values %{ $FNI_ref };
        @dead_ends{keys %{ $dead_end_ref }} = values %{ $dead_end_ref };
    }
}

#a) Remove interactions that end on dead ends from int_from_chains
foreach (keys %dead_ends) {
    #print "dead node: $_\n";
    foreach my $found_chain_int (keys %int_from_chains) {
        my @int_partners = split (/\/t/, $found_chain_int);
        my $target = $int_partners[1];
        if ($_ eq $target) {
            delete $int_from_chains{$found_chain_int};
        }
    }
}

#b) Create the set of found chain nodes. Then get interactions from FNI to chain nodes
#(we cannot get them otherwise because function find_chains stops after finding FNI)
my %found_chain_nodes;
foreach (keys %int_from_chains) {
    my @chain_nodes = split (/\/t/, $_);
    foreach my $node (@chain_nodes) {
        $found_chain_nodes{$node} = "";
    }
}

foreach (keys %potential_chain_nodes) {
    if (exists $first_neighbours_going_in{$_}) {
        foreach my $target (keys %{ $potential_chain_nodes{$_} }) {
            if (exists $found_chain_nodes{$target}) {
                my $FNI_int = $_ . "\t" . $target;
                $int_from_chains{$FNI_int} = "";
            }
        }
    }
}

#c) Get interactions from chain nodes to FNO
#d) Get autoloops on chain interactions
#e) Get interaction description for each chain interaction (look up in %interactions)
foreach (keys %int_from_chains) {
    $int_from_chains{$_} = $interactions{$_};
    my @split_chain_int = split (/\/t/, $_);
    foreach my $int_partner (@split_chain_int) {
        #Go through all interactions of potential chain nodes
        foreach my $possible_target (keys %{ $potential_chain_nodes{$int_partner} }) {
            if (exists $first_neighbours_going_out{$possible_target}) {
                my $int_to_FNO = $int_partner . "\t" . $possible_target; #c
                $int_from_chains{$int_to_FNO} = $interactions{$int_to_FNO}; #e
            }
            #if a chain node has itself as a possible target, the interaction is an autoloop and is added to chain interactions
            if ($possible_target eq $int_partner) {
                my $autoloop = $int_partner . "\t" . $possible_target; #d
                $int_from_chains{$autoloop} = $interactions{$autoloop}; #e
            }
        }
    }
}

```

```

    }
}

my %active_net_interactions = (%core_net_interactions, %int_from_chains);

# OUTPUT section
my $edges_out = "active_nets/" . $cell_type . ".edges.active.csv"; # "active_nets/dnaser_fp/" . $cell_type . ".edges.active.csv";
my $nodes_out = "active_nets/" . $cell_type . ".nodes.active.csv";
my $pajek_out = "active_nets/" . $cell_type . ".active.net";
my $pajek_ref_out = "active_nets/" . $cell_type . ".active.net.deg";

my $eo_fh = files::write_or_open($edges_out, $write);
my $no_fh = files::write_or_open($nodes_out, $write);
my $paj_fh = files::write_or_open($pajek_out, $write);
my $paj_ref_fh = files::write_or_open($pajek_ref_out, $write);

my %active_nodes;
print $eo_fh "Source\tTarget\tEvidence\tResolvedness\tCombinedness\n";
foreach (sort keys %active_net_interactions) {
    my @split_active_int = split(/\t/, $_);
    $active_nodes{$split_active_int[0]} = "";
    $active_nodes{$split_active_int[1]} = "";
    print $eo_fh "$_\t$active_net_interactions{$_}\n";
}
close $eo_fh;

my $index = 1;
my $no_act_nodes = keys %active_nodes;
my %indexed_nodes;
print $no_fh "ID\tExpr\n";
foreach (sort keys %active_nodes) {
    my $state;
    if (exists $expr_tf{$_}) {
        $state = 1;
    } else {
        $state = 0;
    }
    print $no_fh "$_\t$state\n";
    $active_nodes{$_} = $index;
    $indexed_nodes{$index} = $_;
    $index++;
}
close $no_fh;

my %index_instead_of_tf_names_in_int;
foreach (keys %active_net_interactions) {
    my @split_active_int = split(/\t/, $_);
    my $regulator = $split_active_int[0];
    my $target = $split_active_int[1];
    my $ix_reg = $active_nodes{$regulator};
    my $ix_tar = $active_nodes{$target};
    my $ix_int = $ix_reg . "\t" . $ix_tar;
    $index_instead_of_tf_names_in_int{$ix_int} = $active_net_interactions{$_};
}

print $paj_fh "**Vertices $no_act_nodes\n";
foreach (sort {$a<=>$b} keys %indexed_nodes) {
    print $paj_fh "$_\t\"$indexed_nodes{$_}\"\\n\n";
}
print $paj_fh "**Arcs\n";
foreach (keys %index_instead_of_tf_names_in_int) {
    my $val = $index_instead_of_tf_names_in_int{$_};
    if ($val =~ "d") {
        print $paj_fh "$_\t1\n";
    }
    if ($val =~ "i") {
        print $paj_fh "$_\t2\n";
    }
}

```

```

    }
    if ($val =~ "0u") {
        print $paj_fh "$_\\t3\\n";
    }
}
close $paj_fh;

# the second pajek file will contain as vertices all possible vertices from the HOCOMOCO set (for network degree clustering)
my $all_tf_file = "/scratch/todtova/tf_regions/HGNC_symbols_humantf.sorted.txt";
my $all_tf_fh = files::write_or_open($all_tf_file, $open);
my @ref_tf = <$all_tf_fh>;

my $index2 = 1;
my %ref_tf_ind;
foreach (@ref_tf) {
    chomp $_;
    $ref_tf_ind{$_} = $index2;
    $index2++;
}
my $no_ref_nodes = keys %ref_tf_ind;
my %ref_tf_interactions;
foreach (keys %active_net_interactions) {
    my @sp = split(/\\t/, $_);
    my $regulator = $sp[0];
    my $target = $sp[1];
    my $ix_reg = $ref_tf_ind{$regulator};
    my $ix_tar = $ref_tf_ind{$target};
    my $ix_int = $ix_reg . "\\t" . $ix_tar;
    $ref_tf_interactions{$ix_int} = $active_net_interactions{$_};
}

print $paj_ref_fh "*Vertices $no_ref_nodes\\n";
foreach (sort keys %ref_tf_ind) {
    print $paj_ref_fh "$ref_tf_ind{$_}\\t\\\"$_\"\\n";
}
print $paj_ref_fh "*Arcs\\n";
foreach (keys %ref_tf_interactions) {
    my $val = $ref_tf_interactions{$_};
    if ($val =~ "d") {
        print $paj_ref_fh "$_\\t1\\n";
    }
    if ($val =~ "i") {
        print $paj_ref_fh "$_\\t2\\n";
    }
    if ($val =~ "0u") {
        print $paj_ref_fh "$_\\t3\\n";
    }
}
close $paj_ref_fh;

# SUBROUTINES
=head
Description of find_chains
The function explores the interactions of potential chain nodes in an iterative manner. With each step, it is checked if
    a) the node is a dead end
    b) the node has only targets that are dead ends
    c) targets are checked
        c1) is target an FNI?
        c2) was the target already seen before
        c3)
Input Arguments:
    $pot_regulators_ref: hashref that associates each potential chain node with all its targets (hash of hashes)
    $remember_int_ref: hashref that contains all interactions of potential chain. In the last step of the function, interactions from loops are removed.
    $first_neighbours_ingoin_ref: hashref containing all nodes that are either direct FNI pointing to the active core, or nodes that point to FNI
    $current_node: the "current" node whose interactions are checked
    $seen_in_rounds_ref: hashref containing the nodes that have been seen in each round (hash of hashes)
    $lc: counter of the rounds / current round counter
    $remove_ref: hashref containing all loop interactions and int to dead ends, which are removed in the very last step
    $dead_end_ref: hash of nodes that either have no targets or point to nodes that have no targets

```

=cut

```
sub find_chains {
    my ($spot_regulators_ref, $remember_int_ref, $first_neighbours_ingoining_ref, $current_node, $seen_in_rounds_ref, $lc, $remove_ref, $dead_end_ref) = @_;
    my %spot_regulators = %{$spot_regulators_ref};
    my %remember_int = %{$remember_int_ref};
    my %first_neighbours_ingoining = %{$first_neighbours_ingoining_ref};
    my %seen_rounds = %{$seen_in_rounds_ref};
    my %seen_last_round = %{$seen_rounds{$lc}};
    my %remove_from_int = %{$remove_ref};
    my %dead_ends = %{$dead_end_ref};
    my %chain_int;
    my $chain_int_ref = \%chain_int;
    my %chain_int_to_remove_ref = \%remove_from_int;
    my %dead_ref = \%dead_ends;
    my $FNI_ref = \%first_neighbours_ingoining;
    $lc++;
    my %seen_this_round;
    @seen_this_round{keys %seen_last_round} = values %seen_last_round;
    $seen_rounds{$lc} = \%seen_this_round;
    #print "\ncounter: $lc\n";

    #a) target from last int, have you got targets? if not, you're a dead end
    if (keys %{$spot_regulators{$current_node}} == 0) {
        #print "$current_node target of last int is a dead end.\n";
        $dead_ends{$current_node} = "";
    }

    #b) then, check if all targets of the current_node are dead ends.
    my $count_dead_targets = 0;
    my $num_targets_of_current_node = keys %{$spot_regulators{$current_node}};
    foreach (keys %{$spot_regulators{$current_node}}) {
        if (exists $dead_ends{$_}) {
            $count_dead_targets++;
        } #if all targets are dead ends, make regulator a dead end itself
        if ($count_dead_targets == $num_targets_of_current_node) {
            $dead_ends{$current_node} = "";
        }
    }
    $seen_this_round{$current_node} = "";

    #c) check each target of the current node
    foreach my $target_of_current (keys %{$spot_regulators{$current_node}}) {
        my $spot_int = $current_node . "\t" . $target_of_current;
        # check if target of current is an FNI
        if (exists $first_neighbours_ingoining{$target_of_current}) {
            #print "B: $target_of_current, target of $current_node is a first neighbour going in\n";
            $remember_int{$spot_int} = "";
            #check if you've already walked all paths of $current_node. if yes, itself becomes an FNI
            my $check_paths = check_all_paths_walked(\%spot_regulators, \%dead_ends, \%remember_int, $current_node);
            unless ($check_paths eq "NULL") {
                $first_neighbours_ingoining{$current_node} = "";
            }
        } else {
            #check if the target of current is in the collection of nodes which you have seen up to last round
            if (exists $seen_this_round{$target_of_current}) {
                #print "C: $target_of_current, target of $current_node i've seen you already\n";
                my $number_of_seen_ints = 0;
                foreach my $target_of_target (keys %{$spot_regulators{$target_of_current}}) {
                    #check target of target, if its an FNI
                    if (exists $first_neighbours_ingoining{$target_of_target}) {
                        my $int = $target_of_current . "\t" . $target_of_target;
                        $remember_int{$spot_int} = "";
                        #print "$target_of_target, target of $target_of_current is an FNI going in, $spot_int\n";
                        my $check_paths = check_all_paths_walked(\%spot_regulators, \%dead_ends, \%remember_int, $target_of_current);
                        if ($check_paths eq "TRUE") {
                            $first_neighbours_ingoining{$target_of_current} = "";
                        }
                    }
                }
            }
        }
    }
}
```

```

# check if you have seen target of target already
else {
    my $count_seen_targets_of_target = 0;
    if (exists $seen_this_round{$target_of_target}) {
        $seen_this_round{$target_of_target} = "";
        $count_seen_targets_of_target++;

        # if you find you have already seen all targets of target, you might be in a loop. try to break the loop!
        if ($count_seen_targets_of_target == keys %{$spot_regulators{$target_of_current}}) {
            #print "F: i already know all targets of the current target \"$target_of_current\", i'm in a loop\n";
            my (%found_in_loop, %remember_chain_int);
            # try to break loop will return an array of ints if it broke the loop, and a hash if it did not.
            my $loop_ints= try_to_break_loop(\%pot_regulators, \%seen_this_round, \%found_in_loop,

\%remember_chain_int, $target_of_current,1);

            if (ref($loop_ints) eq "HASH") {
                #print "i did not break the loop\n";
                @remove_from_int{keys %{$loop_ints}} = values %{$loop_ints};
            }
            elsif (ref($loop_ints) eq "ARRAY") {
                foreach (@{$loop_ints}) {
                    @found_in_loop{keys %{$_}} = values %{$_};
                }
                @remember_int{keys %found_in_loop} = values %found_in_loop;
            }
        } else {
            unless (exists $first_neighbours_ingoin{$target_of_target}) {
                unless (exists $dead_ends{$target_of_target}) {
                    $seen_this_round{$target_of_target} = "";
                    #print "E: $target_of_target i have not seen this target of $target_of_current yet, i'm going

along here.\n";

find_chains(\%potential_chain_nodes, \%remember_int, \%first_neighbours_going_in, $target_of_target, \%seen_rounds, $lc, \%remove_from_int, \%dead_ends);
                    my $int_from_iterations = %{$chain_int_ref};
                    @remember_int{keys %int_from_iterations} = values %int_from_iterations;
                    @remove_from_int{keys %{$chain_int_to_remove_ref}} = values %{$chain_int_to_remove_ref};
                    @dead_ends{keys %{$dead_ref}} = values %{$dead_ref};
                    @first_neighbours_ingoin{keys %{$FNI_ref}} = values %{$FNI_ref};
                }
            }
        }
    }
}

} # if you have not seen target_of_current yet, send it to the next round
else {
    unless (exists $first_neighbours_ingoin{$target_of_current}) {
        unless (exists $dead_ends{$target_of_current}) {
            $seen_this_round{$target_of_current} = "";
            #print "C: $target_of_current i have not seen this target of $current_node yet, i'm going along.\n";
            $remember_int{$spot_int} = "";
            ($chain_int_ref, $chain_int_to_remove_ref, $dead_ref, $FNI_ref) = find_chains(\%potential_chain_nodes, \%remember_int,

\%first_neighbours_going_in, $target_of_current, \%seen_rounds, $lc, \%remove_from_int, \%dead_ends);
            my $int_from_iterations = %{$chain_int_ref};
            @remember_int{keys %int_from_iterations} = values %int_from_iterations;
            @remove_from_int{keys %{$chain_int_to_remove_ref}} = values %{$chain_int_to_remove_ref};
            @dead_ends{keys %{$dead_ref}} = values %{$dead_ref};
            @first_neighbours_ingoin{keys %{$FNI_ref}} = values %{$FNI_ref};
        }
    }
}

}

}

@remember_int{keys %chain_int} = values %chain_int;
foreach (keys %remove_from_int) {
    delete $remember_int[$_];
}

return (\%remember_int, \%remove_from_int, \%dead_ends, \%first_neighbours_ingoin);

```

```

}

#after you discovered one of the targets of $current_node is an FNI, this sub checks if you have already walked all paths going out from $current_node
#if you have, you will make $current_node a new FNI in order to stop going over already walked paths.
sub check_all_paths_walked {
    my ($spot_regulators_ref, $dead_ref, $remember_ref, $current_node) = @_;
    my %spot_regulators = %{ $spot_regulators_ref };
    my %dead_ends = %{ $dead_ref };
    my %remember_int = %{ $remember_ref };
    my %remembered_paths;
    my $num_targets = keys %{ $spot_regulators{ $current_node } };

    #go through all remembered ints and store those in %remembered_paths that have the current node as regulator
    foreach (keys %remember_int) {
        my @int_partners = split (/\\t/, $_);
        my $regulator = $int_partners[0];
        if ($regulator eq $current_node) {
            $remembered_paths[$_] = "";
        }
    }

    #if regulator has dead ends as targets, store these as well in %remembered_paths
    foreach (keys %{ $spot_regulators{ $current_node } }) {
        if (exists $dead_ends{$_}) {
            $remembered_paths[$_] = "";
        }
    }

    my $number_of_seen_ints = keys %remembered_paths;
    if ($number_of_seen_ints == $num_targets) {
        return "TRUE";
    } else {
        return "FALSE";
    }
}

#for breaking the loop, you go through all seen targets and try to find a target of (target of target) which you have not seen yet.
#if you find one, the function will return an array, if you dont, it will return a hash.
sub try_to_break_loop {
    my ($regulators_to_targets_ref, $seen_before_ref, $found_yet_in_loop_ref, $remember_chain_int_ref, $start_of_loop, $counter) = @_;
    #print "Try to break loop...$counter\\n";
    my %regulators_to_targets = %{ $regulators_to_targets_ref };
    my %seen_before = %{ $seen_before_ref };
    my %found_yet = %{ $found_yet_in_loop_ref };
    my %remember_chain_int = %{ $remember_chain_int_ref };
    my (@ref_to_int, @ref_to_found);

    foreach my $node (keys %{ $regulators_to_targets{ $start_of_loop } }) {
        if (exists $seen_before{ $node }) {
            unless (exists $found_yet{ $node }) {
                $found_yet{ $node } = "found";
                my $int = $start_of_loop . "\\t" . $node;
                $remember_chain_int{ $int } = "";
                my $loop_int_ref = try_to_break_loop(\\%regulators_to_targets, \\%seen_before, \\%found_yet, \\%remember_chain_int, $node, $counter++);
                if (ref($loop_int_ref) eq "HASH") {
                    @remember_chain_int(keys %{ $loop_int_ref }) = values %{ $loop_int_ref };
                }
                elsif (ref($loop_int_ref) eq "ARRAY") {
                    my @temp = (@ref_to_int, @$loop_int_ref);
                    @ref_to_int = @temp;
                }
            }
        } else {
            my $int = $start_of_loop . "\\t" . $node;
            $found_yet{ $node } = "";
            $remember_chain_int{ $int } = "";
            my $reference = \\%remember_chain_int;
            push(@ref_to_int, $reference);
        }
    }

    if (scalar @ref_to_int == 0) {

```

```
        return \%remember_chain_int;  
    } else {  
        return \@ref_to_int;  
    }  
}
```

Kristina Todtova

Phone: +43 650 8170 958

email: kristina.todtova@gmx.at

Education / Training	
Master studies (Oct 11 – Sep 14)	Molecular Biology University of Vienna, Universitätsring 1, 1010 Vienna Specialization in Computational Systems Biology / Bioinformatics Master thesis on TF regulatory network topology and dynamics (Feb 13 – Sep 14) Specialization in Neurosciences (Oct 11 – Jan 13)
Summer school (July 13)	Lipari Summer School on Bioinformatics and Computational Biology Network biology approaches to understanding complex diseases, concepts and algorithms
Bachelor studies (Sep 08 – Jul 11)	Molecular Biotechnology FH Campus Wien, Helmut-Qualtinger-Gasse 2, 1030 Vienna
Research experience	
Master thesis (Feb 13 – Sep 14)	<i>“Common Network Structure in the Regulatory Core of Trait Development”</i> CUBE - Division for Computational Systems Biology, Dept. of Microbiology and Ecosystem Science Althanstrasse 14, 1090 Vienna
Elective laboratory traineeship in neurosciences (Oct 12 – Jan 13)	<i>“Circadian clock gene expression in the baso-lateral amygdala of mice”</i> with Prof. Daniela Pollak-Monje Quiroga, Dept. for Neurophysiology and Neuropharmacology, Medical University of Vienna, Schwarzspanierstrasse 17, 1090 Vienna
Skills	
Bioinformatics	Transcriptome bioinformatics (RNAseq), DNaseI footprint analysis, Modeling TF regulatory networks from high-throughput data
Programming	Perl, Bioperl, Bash, R
Wet lab	RNA isolation, qRT-PCR, fluorescent immunohistochemistry, cell culture
Languages	English, German, Slovakian – fluent (speaking, reading, writing) Swedish – basic (speaking, reading, writing)