Université Sorbonne Paris Cité

Thèse de doctorat

de l'Université Sorbonne Paris Cité

Préparée à l'Université Paris Diderot

Ecole doctorale Frontières Du Vivant ED 474

Laboratoire "Epigénétique et Cancer" CNRS UMR9198/FRE3377

MasterPATH: network analysis of functional genomics screening data

Par Natalia Rubanova

Thèse de doctorat de Biologie

Dirigée par Nadya Morozova

Présentée et soutenue publiquement à l'IHÉS le 22/02/2018

Président du jury :	Soulé, Christophe / Dr. / CNRS, IHÉS
Rapporteurs :	Campalans, Anna / Dr. / CEA, Université Paris Diderot
	van Baalen, Minus / Dr. / CNRS, Institut Biologie de l'ENS
Examinateurs :	Zinovyev, Andrei / Dr. / Institut Curie, Université PSL
	Ginestier, Christophe / Dr. / INSERM, Université d'Aix-Marseille
	Polesskaya, Anna / Dr. / CNRS, Ecole Polytechnique
Directeur de thèse :	Morozova, Nadya / Dr. / CNRS, Université Paris Sud, CEA Saclay
Membre invités :	Harel-Bellan, Annick / Dr. / IHÉS

Titre : MasterPATH : l'analyse de réseau des données expérimentales de la génomique fonctionnelle.

Résumé : Les technologies « omics » et notamment les technologies de la génomique fonctionnelle utilisent des approches expérimentales à haut débit dont le but est déterminer des composants biologiques (des gènes, l'acide ribonucléique messager, l'acide ribonucléique micro, des protéines…) relevant de ce phénotype. La perte de fonction de l'étude basée au mécanisme ARN-interférence ou CRISPR/Cas9 et le profilage du transcriptome (en utilisant des puces d l'ADN ou des méthodes RNA-seq) sont les technologies « omics » les plus répandues de nos jours. Telles études génèrent de grands volumes des données qui sont représenté d'habitude par une liste rangée des composants biologiques d'où on peut obtenir une soi-disant « hit-liste » en utilisant des méthodes statistiques convenantes. Bien qu'il suffit pour définir les composants biologiques principaux qui sont importants pour ce phénotype, les méthodes de rangement ne peuvent pas aider à déterminer la figure du système biologique étudié la plus détaillée.

Depuis 15 ans on a élaboré une grande quantité des méthodes calculatoires pour l'interprétation des données « omics » et la définition des mécanismes moléculaires qui sont la base de ce phénotype. On peut diviser ces méthodes en deux catégories : (1) les méthodes de l'analyse des voies moléculaires définissant les voies moléculaires canoniques ou les listes annotées des composants biologiques qui sont sur- ou sous-représenté dans la liste rangée (ou dans la « hit-liste »); (2) les méthodes de l'analyse de réseau qui utilisent les réseaux des interactions moléculaires comme l'information supplémentaire. La plupart des méthodes de l'analyse de réseau publiées définissent un sous-réseau qui est considérablement enrichi par les composants biologiques de la liste rangée (ou la « hit-liste »). Telle sous-réseau peut aider à définir des relations importantes entre les composants biologiques dans le contexte de ce système biologique.

Dans ce travail nous avons élaboré une nouvelle méthode de l'analyse de réseau à définir des membres possibles des voies moléculaires qui sont important pour ce phénotype en utilisant la « hit-liste » des expériences « omics » qui travaille dans le réseau intégré (le réseau comprend des interactions protéine-protéine, de transcription, l'acide ribonucléique micro-l'acide ribonucléique messager et celles métaboliques). La méthode tire des sous-réseaux qui sont construit des voies de quatre types les plus courtes (qui ne se composent des interactions protéine-protéine, ayant au minimum une interaction de transcription, ayant au minimum une interaction l'acide ribonucléique micro-l'acide ribonucléique messager, ayant au minimum une interaction métabolique) entre des hit –gènes et des soi-disant « exécuteurs terminaux » - les composants biologiques qui participent à la réalisation du phénotype finale (s'ils sont connus) ou entre les hit-gènes (si « des exécuteurs terminaux » sont inconnus). La méthode calcule la valeur de la centralité de chaque point culminant et de chaque voie dans le sous-réseau comme la quantité des voies les plus courtes trouvées sur la route précédente et passant à travers le point culminant et la voie. L'importance statistique des valeurs de la centralité est estimée en comparaison avec des valeurs de la centralité dans les sous-réseaux construit des voies les plus courtes pour les hit-listes choisi occasionnellement. Il est supposé que les points culminant et les voies avec les valeurs de la centralité statistiquement signifiantes peuvent être examinés comme les membres possibles des voies moléculaires menant à ce phénotype. S'il y a des valeurs expérimentales et la P-valeur pour un grand nombre des points culminant dans le réseau, la méthode fait possible de calculer les valeurs expérimentales pour les voies (comme le moyen des valeurs expérimentales des points culminant sur la route) et les P-valeurs expérimentales (en utilisant la méthode de Fischer et des transpositions multiples).

A l'aide de la méthode masterPATH on a analysé les données de la perte de fonction criblage de l'acide ribonucléique micro et l'analyse de transcription de la différenciation terminal musculaire et les données de la perte de fonction criblage du procès de la réparation de l'ADN. On peut trouver le code initial de la méthode si l'on suit le lien https://github.com/daggoo/masterPATH.

Mots clefs : Analyse des réseaux, perte de fonction criblage, profilage du transcriptome, CRISPR/Cas9, voies moléculaires.

Title : MasterPATH: network analysis of functional genomics screening data.

Abstract : "Omics" and in particular functional genomics technologies rely on high-throughput experimental approaches aiming at identifying a set of biological components (genes, mRNA, miRNA, proteins…) relevant to a given phenotype. RNAi- or CRISPR/Cas9-based loss-of-function screenings and transcriptomic profiling (using microarrays or RNA-seq methods) are nowadays among the most widespread "omics" technologies. Although different in nature, such studies commonly generate large datasets, usually presented in the form of a ranked list of biological components, from which a so-called 'hit list' can be retrieved by applying an appropriate statistical threshold. While this is usually sufficient to identify key biological components relevant to a given phenotype, ranking methods fail to provide a broader picture of the biological system under study.

Numerous computational methods to interpret "omics" datasets and infer molecular machinery underlying a given phenotype have been developed in the past 15 years. They can be grouped into two categories: (1) pathway analysis methods, relying on the identification of canonical pathways or annotated biological components that are over- or underrepresented in a ranked dataset (or a hit list). (2) network analysis methods, using molecular interaction networks to provide complementary information. Most of the published network analysis methods aim at identifying a subnetwork significantly enriched with the biological components from a ranked dataset (or a hit list). The subnetwork can thus facilitate the identification of important relationships between the biological components, in the context of a specific biological system.

In this work we developed a new exploratory network analysis method, that works on an integrated network (the network consists of protein-protein, transcriptional, miRNA-mRNA, metabolic interactions) and aims at uncovering potential members of molecular pathways important for a given phenotype using hit list dataset from "omics" experiments. The method extracts subnetwork built from the shortest paths of 4 different types (with only protein-protein interactions, with at least one transcription interaction, with at least one miRNA-mRNA interaction, with at least one metabolic interaction) between hit genes and so called "final implementers" – biological components that are involved in molecular events responsible for final phenotypical realization (if known) or between hit genes (if "final implementers" are not known). The method calculates centrality score for each node and each path in the subnetwork as a number of the shortest paths found in the previous step that pass through the node and the path. Then, the statistical significance of each centrality score is assessed by comparing it with centrality scores in subnetworks built from the shortest paths for randomly sampled hit lists. It is hypothesized that the nodes and the paths with statistically significant centrality score can be considered as putative members of molecular pathways leading to the studied phenotype. In case experimental scores and p-values are available for a large number of nodes in the network, the method can also calculate paths' experiment-based scores (as an average of the experimental scores of the nodes in the path) and experiment-based p-values (by aggregating p-values of the nodes in the path using Fisher's combined probability test and permutation approach).

The method is illustrated by analyzing the results of miRNA loss-of-function screening and transcriptomic profiling of terminal muscle differentiation and of 'druggable' loss-of-function screening of the DNA repair process. The Java source code is available on GitHub page https://github.com/daggoo/masterPATH.

Keywords : Network analysis, loss-of-function screening, RNA interference, transcriptome profiling, CRISPR/Cas9, molecular pathway

Table of Contents

Acknowledgements

I would like to thank my supervisors Dr. Annick Harel-Bellan and Dr. Nadya Morozova; my Thesis Advisory Committee members Dr. Anna Campalans and Dr. Andrey Zinoviev; Dr. Guillaume Pinna and Dr. Anna Poleskaya for participating in the analysis of the experimental data and providing valuable comments on the text.

List of Figures

List of Tables

1 Background

Functional genomics technologies

Three the most widespread functional genomics technologies are described below.

RNAi-based loss-of-function screening

This section is influenced by Bertil Daneholt's paper on RNA interference (Daneholt, 2006).

The biological mechanism for repression of gene expression by causing the degradation of mRNA is exploited in the RNA interference(RNAi) based loss-of-function screening.

RNA interference (RNAi) was first detected at least in early 1980s although it was known by other names and the different findings were considered as unrelated processes. First, it was discovered that about 100 nucleotides RNA molecules can bind to a complementary mRNA sequence and inhibit translation in *Escherichia coli* in 1984 and 1994. Also, a phenomenon called gene (or RNA) silencing was shown in plants around 1990 in the works of (Matzke et al., 1989) and (Wassenegger et al., 1994). It was shown that the gene activity can be induced, stimulated and the expression of homologous sequences can be inhibited in transgenic plants with an incorporated transgene into the genome. The latter was called homology-dependent gene silencing. The inhibition of gene activity was reported at the transcriptional level (transcriptional gene silencing, TGS) (Matzke et al., 1989; Wassenegger et al., 1994) and at the posttranscriptional level (posttranscriptional gene silencing, PTGS) (Napoli, 1990; van der Krol, 1990). A process similar to PTGS called quelling was also observed in the fungus *Neurospora crassa* (Romano et al., 1992). Also, two groups reported about regulation of translation by antisense RNA in the nematode worm *Caenorhabditis elegans* in 1993 (Rosalind C. Lee et al., 1993; Wightman et al., 1993). A second example of a small regulatory RNA - the 21-nucleotide *let-7* RNA was found in *Caenorhabditis elegans* (Reinhart et al., 2000) which was also present in other species was reported in 2000.

The immense potential of all these finding became evident after Andrew Fire and Craig Mello published their break-through study in 1998 (Fire et al., 1998) and a large number of small RNA molecules, called microRNAs (miRNAs), were revealed in 2001 (Lagos-Quintana et al., 2001; Lau et al., 2001; R C Lee et al., 2001).

Andrew Fire and Craig Mello studied the phenotypic effect of sense, antisense and annealed sense/antisense RNA introduced into the nematode worm *Caenorhabditis elegans* in their paper. They showed that only introduction of the annealed sense/antisense RNA could cause the predicted phenotype and led to an efficient degradation of the target mRNA. The main results of the paper can be summarized as follows (cited from (Daneholt, 2006):

> *First, silencing was triggered efficiently by injected double-stranded RNA (dsRNA), but weakly or not at all by sense or antisense single-stranded RNAs. Second, silencing was specific for an mRNA homologous to the dsRNA; other mRNAs were unaffected. Third, the dsRNA had to correspond to the mature mRNA sequence; neither intron nor promoter sequences triggered a response. This indicated a posttranscriptional, presumably cytoplasmic mechanism. Fourth, the targeted mRNA disappeared suggesting that it was degraded. Fifth, only a few dsRNA molecules per cell were sufficient to accomplish full silencing. This indicated that the dsRNA was amplified and/or acted catalytically rather than stoichiometrically. Sixth, the dsRNA effect could spread between tissues and even to the progeny, suggesting a transmission of the effect between cells. Furthermore, Fire and Mello made the remark that RNAi could provide an explanation for a phenomenon studied in plants for several years: posttranscriptional gene silencing (PTGS). Finally, they ended their paper by speculating about the possibility that "dsRNA could be used by the organism for physiological gene silencing"*.

Moreover, Fire provided evidences that dsRNAs target mRNA before translation and suggested that the RNAi mechanism could be a form of defense system to viral particles in lower organisms in his next paper published also in 1998 (Montgomery et al., 1998).

The presence of the RNAi mechanism was shown in other organisms, including fruit flies, trypanosomes, plants, planaria, hydra and zebrafish (Tuschl et al., 1999) very rapidly after these publications. However, specific dsRNA-mediated mRNA degradation by RNAi was not detected in commonly used mammalian cell cultures first. But later it was archived by introducing very short 21-nucleotide dsRNAs (Elbashir, Harborth, et al., 2001).

Next, it was shown that dsRNA is cleaved into 21-23 nucleotide long siRNAs (small interfering RNA) and it was proposed that these siRNAs work as guide to cleave mRNA (Zamore et al., 2000). The process was detected *in vivo* in *Caenorhabditis elegans* in 2000: it was observed that dsRNAs are cleaved into about 25-nucleotide RNAs and that antisense RNAs trigger specific dsRNA-mediated mRNA degradation via base-pairing to mRNA (Parrish et al., 2000).

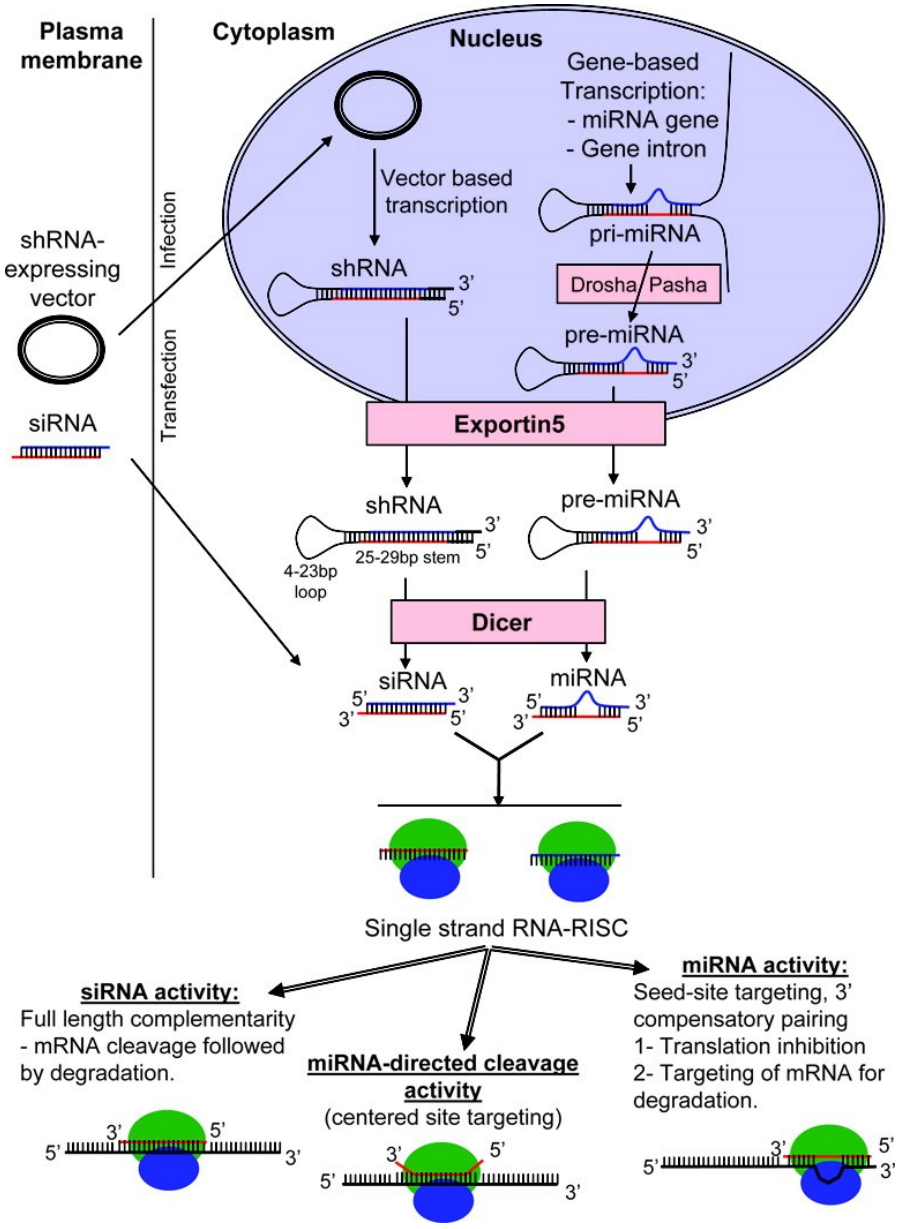
Later it was demonstrated in Drosophila cells that a large protein complex called RISC (RNA-induced silencing complex) is guided to the mRNA by a short antisense RNA and after that mRNA is cleaved and degraded (Hammond et al., 2000). The protein complex RISC contained at least one member of the argonaute protein family and it was suggested that it acted as an endonuclease. It was also shown that a ribonuclease III-like nuclease, called Dicer, trimmed dsRNA to form short RNAs (Bernstein et al., 2001) (Summarized in figure 1-1).

Meanwhile RNAi pathway for miRNAs was also studied. It was found that in contrary to exogenous nature of siRNAs, most miRNA are transcribed from independent transcription units (Lagos-Quintana et al., 2001; Lau et al., 2001; R C Lee et al., 2001) or from the introns of pre-mRNAs (about a quarter of the human miRNA genes). miRNAs are transcribed into pri-miRNAs (Y. Lee et al., 2002), that are cleaved by a protein complex called Drosha/Pasha into miRNA precursor molecules called pre-miRNAs (Y. Lee et al., 2002; Zeng et al., 2003). pre-miRNAs as shRNAs are transported into the cytoplasm by a mechanism involving Ran-GTP protein and the export receptor Exportin-5 (Lund et al., 2004; Yi et al., 2003). pre-miRNAs as shRNAs are processed by the enzyme Dicer (Y. Lee et al., 2003) which cleaves shRNAs into mature siRNAs and pre-miRNAs into miRNAs. siRNA and miRNA single strands are incorporated into RISC complexes and guide the complex to target mRNAs (Elbashir, Lendeckel, et al., 2001; Elbashir, Martinez, et al., 2001; Hammond et al., 2000; Martinez et al., 2002). The fate of the targeted mRNA depends on the extent of sequence pairing and it

can be cleaved, destroyed or translation can be inhibited (Bartel, 2004) (Summarized in figure 1-1).

Figure 1-1. dsRNA and miRNA interference pathway.

Reprinted with permission from (Sigoillot et al., 2011).



The importance of RNAi discovery is difficult to overestimate. One of the most significant consequences of the discovery was that the sequence specific action of RNAi immediately

suggested that it can be used to suppress specific genes and study the resulting phenotype. Initially it was used to silence individual genes of interest but the development of libraries for several organisms that contain synthesized sequence specific dsRNAs targeting most of the genes in the genome allowed to perform genome wide loss of function screening.

Different types of dsRNAs such as siRNAs, endoribonuclease-prepared siRNAs (esiRNAs) or siRNA precursors (such as short hairpin RNAs (shRNAs) or long double-stranded RNAs (dsRNAs) are used to perform screening. Typically, small interfering siRNAs or shRNA are used for mammalian cells (S. E. Mohr et al., 2014). Targeting non-coding RNA became also possible with development of libraries of reagents that inhibit or mimic microRNAs (S. E. Mohr et al., 2014).

The way RNAi reagents are introduced into the cells differ for different cell types and reagents: siRNAs or esiRNAs can be transfected into cells; shRNAs can be virally transduced into cells; dsRNAs in solution can be applied to D. melanogaster cells; dsRNAs (fruit fly or *Caenorhabditis elegans*) or shRNAs (fruit fly or mice) can be expressed from transgenic constructs; dsRNAs can be microinjected (*Caenorhabditis elegans*, fruit fly and some non-model insects); E. coli expressing dsRNAs can be fed to model organisms (*Caenorhabditis elegans* or planaria) (S. E. Mohr et al., 2014).

Typical components of a genome wide RNAi loss of function screening are (as described by (Boutros et al., 2008):

Developing readout assay (primary screening assay) – this step includes designing a readout assay specific for the biological process of interest, choosing positive and negative controls "to achieve high signal with the positive controls and low noise with the negative controls" and developing scoring strategy which can be either qualitative or quantitative. The readout assay for genome wide screening is often a compromise between specificity and feasibility. As examples of readout assay, one could mention simple visual assays of morphological changes, changes in the expression of GFP reporters, or imaging assays for cell culture-based screens which capture an image of each well and then each individual cell is scored with several phenotypic descriptors.

The pilot screening – this step includes a small-scale screening of a few hundred random genes, positive and negative controls. The pilot screen is needed to adjust RNAi reagents doses, incubation times, other experimental variables; to optimize readout assay responses to negative and positive control reagents; to assess the feasibility to perform the screen on a large scale (Sharma et al., 2009).

The genome-wide screening – this step includes genome-wide screening which is usually conducted in duplicates or more with several non-overlapping RNAi reagents per gene. This helps to avoid false positives caused by off-target effects and to increase confidence in true positives.

A cell-culture based RNAi screening is typically performed in either arrayed or pooled format. The library is randomly introduced into cells in the pooled format screening such that any given cell will contain approximately one gene-specific RNAi reagent. A library containing viral-encoded shRNAs is typically used for mammalian cells. Then the cells can be selected for presenting a certain feature (e.g. resistance to some treatment, being positive in the assay). The polymerase chain reaction (PCR) is used to amplify RNAi reagent present in the selected cells and sequencing is used to determine the RNAi reagents. The presence of the RNAi reagent after selection suggests the corresponding gene is a hit (S. Mohr et al., 2010).

Another option to perform selection is to use several subsets of cells or even different cell types that can be treated differently to create a “reference set” and one or more “experimental sets” (S. Mohr et al., 2010). It can be done before or after introducing RNAi library into the cells depending in the readout assay. PCR amplification and microarray analysis or sequencing are used then to detect which RNAi reagents are present in each set (e.g. using molecular “barcode”). This gives the information about which RNAi reagents are present and which quantities in the experimental and reference sets (S. Mohr et al., 2010).

In contrary to the pooled format, each gene is targeted separately by RNAi reagents in individual wells of a microtiter plate (that contain e.g. 96 or 384 wells) in arrayed format. The readout assays are usually use colorimetric, fluorescence, or luminescent measurements

at the well level or fluorescent measurements at the cellular or subcellular level using imaging (S. Mohr et al., 2010). Genome scale RNAi arrayed screenings use automated equipment for liquid handling and readout assays.

Secondary validation – this step includes secondary screening which validates positives from the genome-wide screening. This can be archived by using multiple RNAi reagents distinct from the genome-wide screening, improving specificity of the readout assay, developing a readout assay for testing for nonspecificity, using different biological system or using RNAi independent techniques.

Data analysis – this step includes raw data analysis from the genome-wide screening and secondary validation. The pipeline is specific to the screening and readout assay technologies, but typically includes data preprocessing, quality control, removal of spatial biases per plate and normalization between plates for arrayed format, microarray or sequencing analysis for pooled format, creating a ranked list of genes with experimental measurements and detection of statistically significant hits. Software packages such as cellHTS (Boutros et al., 2006), RNAither (Rieber et al., 2009), CARD (Dutta et al., 2016), shALIGN and shRNAseq (Sims et al., 2011), edgeR (Y. Chen et al., 2015) which implement the pipeline are available.

One important point should be mentioned about RNAi loss-of-function screening. Soon after completion of the first genome-wide RNAi screenings, it became evident that many genome-wide RNAi screenings hits were false positives due to off-target effects (OTEs) (Birmingham et al., 2006; Jackson et al., 2003; S. Mohr et al., 2010). These OTEs can be grouped into two main categories: sequence-independent off-target effects and sequence-specific off-target effects (Sigoillot et al., 2011). The main causes sequence-independent off-target effects are (1) shRNAs can interfere with processing mechanism for microRNAs (Grimm et al., 2006); (2) siRNAs can displace microRNAs from RISC complex (Khan et al., 2009); (3) non-specific immune response can be induced after introduction of synthetic RNAi reagents (Bridge et al., 2003; Sledz et al., 2003); (4) high concentration of siRNAs can induce cell-stress response (Persengiev et al., 2004; Semizarov et al., 2003). The main cause of sequence-specific off-target effects is the potential of each individual RNAi reagent to act like a miRNA and to

downregulate the expression of several genes, when besides inducing the cleavage of the target mRNA with perfect base-pairing, the RNAi reagent can act as miRNA by interacting with target mRNA with incomplete base-pairing inside a short seed region (Carthew, Richard W. and Sontheimer, 2009; Doench et al., 2003). Several computational and experimental approaches were developed to address this issue, including computational prediction of potential OTEs using seed sequence alignments (e.g. Genome-wide Enrichment of Seed Sequence matches (GESS) (Sigoillot et al., 2012); designing RNAi reagents that will result in fewer OTEs (e.g. using as control altered RNAi reagent sequence to prevent perfect complementarity (Buehler et al., 2012); using multiple RNAi reagents with different sequence per gene (Bassik et al., 2009; S. E. Mohr et al., 2014).

CRISPR/Cas9-based loss-of-function screening

CRISPR/Cas system is bacteria and archaea adaptive immunity mechanism (Jinek et al., 2012). The first description of repeat structures that would later be called CRISPR was published in 1987 by a group of Japanese scientists from Osaka university who cloned by accident iap gene from the E. Coli genome with a part of CRISPR locus. These repeats consisted of a series of 29-nucleotide repeated sequence separated by unique 32-nucleotide 'spacer' sequences (Ishino et al., 1987), however its function was not known at that time. A group from Netherlands published two papers about similar repeat clusters in *Mycobacterium tuberculosis* genome in 1993 (Groenen et al., 1993; Van Soolingen et al., 1993). A cluster of interrupted direct repeats (DR) was found and the diversity of the DR-intervening spacers was studied in different strains of *Mycobacterium tuberculosis* in these papers. At the same time in 1993, similar repeated structures were observed by researchers from Spain in archaeal microbe *Haloferax mediterranei* (F. J.M. Mojica et al., 1993) and first hypotheses for the function of these structures were proposed (it was proposed that these repeats could have a role in partitioning of the replicated DNA) (F. J M Mojica et al., 1995).

Similar repeat clusters were recognized using bioinformatics methods in many genomes of Archaea and Bacteria in 2000 and 2002 and were named clustered regularly interspaced palindromic repeats (CRISPR) (Jansen et al., 2002; F. J M Mojica et al., 2000). It was also reported that CRISPR was accompanied by a set of homologous genes that were called cas genes (CRISPR-associated systems) (Jansen et al., 2002). Helicase and nuclease motifs were identified in the Cas proteins what suggested that these proteins might be involved into organizing the structure of the CRISPR loci (Jansen et al., 2002). Three independent groups showed that some CRISPR spacers matched phage and extrachromosomal DNA in 2005 (Bolotin et al., 2005; F. J M Mojica et al., 2005; Pourcel et al., 2005) which suggested that CRISPR loci encoded an adaptive immune system mechanism to protect bacteria and archaea. The first experimental evidence that proofed this suggestion was published in 2007 (Barrangou et al., 2007). It was shown that the CRISPR loci of the phage resistance strains of *Streptococcus thermophilus* contained phage-derived sequences. The number of acquired spacers correlated with the increased resistance. It was also shown that bacteria

needed Cas7 protein to gain phage resistance suggesting that Cas7 was important for spacer acquisition. On the other hand, Cas9 protein was necessary for preventing phage infection.

Later, it was observed that a complex of Cas proteins cleaved CRISPR RNA that is transcribed from the CRISPR locus into spacer-containing RNA molecules (crRNA) (Brouns et al., 2008). In the same year, it was also observed that CRISPR targeted DNA rather than RNA to prevent conjugation in *Staphylococcus epidermidis* (Marraffini et al., 2008) which was confirmed two years later by a study that gave direct experimental evidences that CRISPR-Cas system cleaves both strands of plasmid DNA in *S. thermophiles* (Garneau et al., 2010). The latter study also showed that the Cas9 nuclease cuts DNA at sequence specific positions encoded by crRNAs. Two more years later, Jinek et al. (Jinek et al., 2012) reported that the core CRISPR/Cas mechanism in *S. pyogenes* is based on a dual-RNA structure that consists of trans-activating crRNA (tracrRNA) and precursor crRNA hybridized together. This hybrid guides Cas9 endonuclease to introduce sequence specific double-stranded breaks in target DNA.

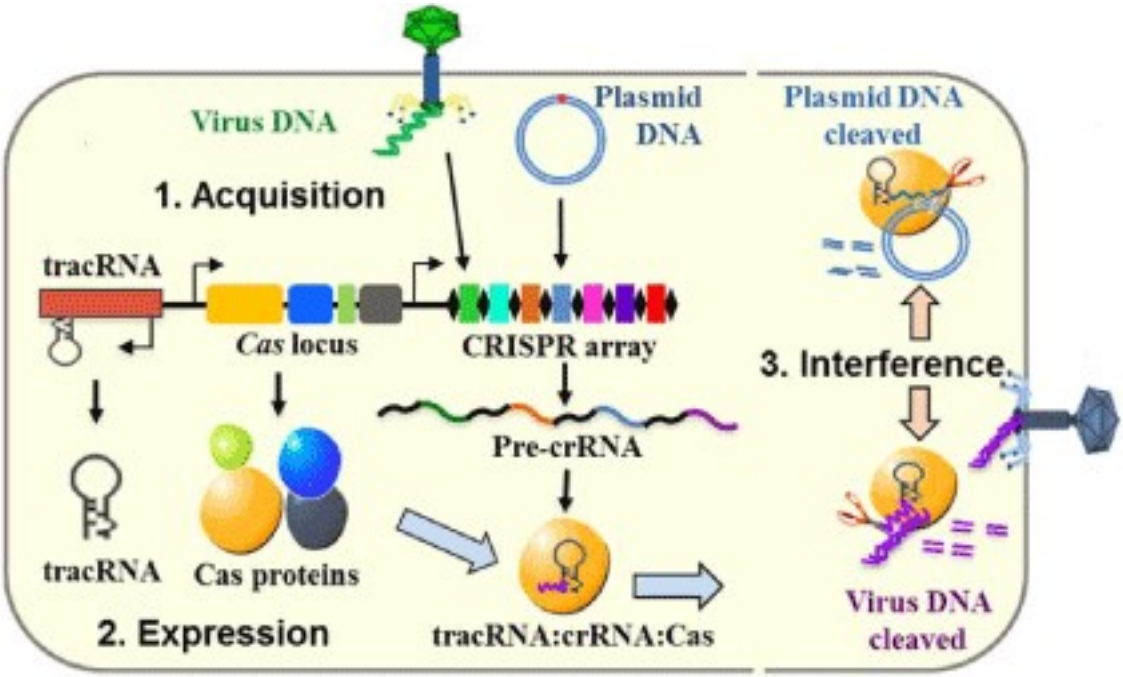
Nowadays the CRISPR defense system mechanism is divided into three stages (Makarova et al., 2011) (Figure 1-2):

1) Acquisition/adaptation stage – during this stage viral DNA is captured and is inserted into the CRISPR locus in a form of spacers. Two Cas proteins, Cas1 and Cas2, are observed in most of types of CRISPR-Cas systems (classification of CRISPR-Cas systems see below) and responsible for spacer acquisition.

2) Expression/processing stage – during this stage CRISPR is transcribed into long precursor CRISPR RNA (pre-crRNA) which is then processed into mature crRNA with the help of Cas proteins and other factors depending on the type of CRISPR-Cas system.

3) Interference stage – during this stage Cas protein(s) guided by crRNA destroy target nucleic acid.

Figure 1-2. Three stages of CRISPR-Cas immunity mechanism.
Adapted from (Yongmin Yan, 2013).



It is known that ~40% of sequenced bacterial genomes and ~90% of sequenced archaeal genomes contain at least one CRISPR locus encoding CRISPR/Cas system (Grissa et al., 2007). The current formal classification (Makarova et al., 2015) distinguishes two classes – Class I and Class II mostly defined by different sets of cas genes present in CRISPR locus. Class I comprises types I, III and IV systems. Class II comprises types II, V systems. Several types of CRISPR/Cas system can be present in a genome.

Class 1 CRISPR–Cas systems are defined by the presence of a multiprotein crRNA effector complex. The class includes type I, type III and type IV CRISPR–Cas systems.

Type I and type III CRISPR-Cas systems are distinguished by the presence of the signature genes cas3 and cas10 respectively. The multiprotein crRNA effector complexes mediate the processing and interference stages of the CRISPR defense system (Makarova et al., 2015). In

type I systems, this complex is known as the CRISPR-associated complex for antiviral defense (Cascade) complex. In type III-A and type III-B systems the complexes are known as Csm and Cmr complexes respectively (Makarova et al., 2015). Type IV systems are rare and lack proteins responsible for adaptation stage.

Class 2 CRISPR–Cas systems are defined by the presence of a single subunit crRNA effector complex. This class includes type II and V CRISPR–Cas systems.

Type II CRISPR-Cas system is distinguished by the presence of cas9 gene which combines the functions of the crRNA–effector complex and is the only protein that is required to perform DNA cleavage. Cas9 protein is also takes part in spacer adaptation and in processing of pre-crRNA. Besides pre-crRNA, another RNA known as tracrRNA is transcribed from the repeat region of CRISPR locus. tracrRNA forms dsRNA with pre-crRNA in the repeat region via base-pairing. The dsRNA is targeted by the housekeeping dsRNA- specific ribonuclease RNase III but not Cas protein in the presence of the Cas9 protein to produce crRNAs (Hale et al., 2009).

Type V CRISPR-Cas system is distinguished by the presence of cas12 gene which combines the functions of the crRNA–effector complex. This type did not require the additional tracrRNA for target cleavage unlike type II (Koonin et al., 2017).

After Marraffini and Sontheimer's 2008 paper that showed that CRISPR targeted and cut DNA in crRNA guided manner, big work devoted to adaptation of the mechanism to cutting, and as result, editing specific genomic loci had started, at the same time with continuation of investigation of basic biology of CRISPR-Cas defense system mechanism.

Two molecular techniques were mainly in use to edit eukaryotic DNA in a sequence specific manner within the cell at that time: zinc finger nucleases (ZFNs) and transcription activator–like effector nucleases (TALENs) by 2008. Zinc finger nuclease is an artificial fusion of DNA-binding domain of eukaryotic transcription factors (zinc finger proteins, which consist of several zinc finger motifs each recognizing approximately 3 bp of DNA) with a DNA-cleavage domain of the *Fok*I restriction enzyme (Urnov et al., 2010). Transcription activator-like effector nuclease is an artificial fusion of DNA binding domain of transcription

activator-like effectors (TALE) proteins (these proteins contain nearly identical 34-amino-acid repeats with only two variable amino acids per repeat, each repeat recognize a single base pair in target DNA) (Boch, 2011) from the pathogenic bacterium *Xanthomonas* with a DNA-cleavage domain of the *Fok*I restriction enzyme.

These sequence-specific nucleases introduce double stranded breaks in the DNA and promotes primarily either homology-directed repair (HDR) pathway in the presence of a template DNA that contain the mutation of interest, thereby archiving site specific mutagenesis, or non-homologous end-joining (NHEJ) pathway without a template DNA to disrupt a target gene. The latter pathway is an error-prone repair mechanism that can introduce short indel mutations at repair sites which cause a coding frameshift, resulting in a premature stop codon, as well as an early frameshift mutation or large indels resulting in a non-functional protein (Shalem et al., 2015).

Both approaches require engineering of new proteins for each target sequence which is hardly feasible for large scale experiments. On the other hand, type II CRISPR/Cas system required only short crRNA: tracrRNA hybrid and Cas9 protein to introduce sequence-specific double stranded breaks and it would provide a versatile tool for cutting/editing of DNA if it could be made to work in mammalian cells. It took several years to make it work. The first papers reporting that CRISPR/Cas9 system from *Streptococcus pyogenes* engineered in zebrafish embryos, human and mouse cells can induce targeted genetic modifications via NHEJ or HDR-mediated pathways appeared in 2013 (Cong et al., 2013; Hwang et al., 2013; Mali et al., 2013). Since these initials studies, many laboratories used CRISPR/Cas9 system for genome editing applications including developing and construction of targeting reagents for functional screenings.

The field of functional screenings using CRISPR/Cas9 reagents is rapidly developing. According to literature, the screening methodology consists of several steps (Miles et al., 2016; Shalem et al., 2015):

Choosing CRISPR library

CRISPR knockout libraries are the most common libraries for CRISPR/Cas9 functional screenings. They are available as pooled plasmid libraries (see screening format section below), containing either two separate plasmids for cas9 and sgRNA or a single plasmid with both cas9 and sgRNA. Genome-wide knockout libraries contained sgRNAs targeting over 18 000 genes and about 1000 miRNAs by 2016.

CRISPR-based interference (CRISPRi) libraries contain sgRNAs with catalytically dead Cas9 (dCas9) protein that does not cause permanent DNA modification but repress transcription via blocking RNA polymerase or via effector domain if fused with repressive effector domain like Kruppel-associated box (KRAB).

CRISPR gene activation (CRISPRa) libraries also contain sgRNAs with dCas9, but this time dCas9 is fused with activation domain proteins (Maeder et al., 2014; Perez-Pinera et al., 2013), causing an increase in gene expression level.

Choosing cell line

The main considerations for choosing a cell line for CRISPR/Cas9 functional screening are ploidy (or more precisely gene copy number) and state of the DNA repair pathways. The first one is important because complete gene knockout is easier to archive when there are few gene copies in the genome. The latter is important because if both HDR and NHEJ pathways are active in the cell line, the DSB can be repaired via HDR using sister chromatid as a template, so the probability of a complete gene knockout is lower than for cell lines with defective HDR.

Choosing screening format

In general, both arrayed and pooled screening formats can be used for CRISPR/Cas9 functional screening. However, all 84 high-throughput screenings performed in human cell lines submitted to GenomeCRISPR database by October 2016 were pooled screenings (Rauscher et al., 2016). One possible explanation for this is that pooled format screenings are less expensive and labor intensive; moreover, screening reagents for pooled formats are easier to produce (Shalem et al., 2015). However as mentioned for RNAi screenings, pooled

formats are limited to simple phenotype readout assays (as cell proliferation or survival) (Shalem et al., 2015).

The selection of the cells is performed after viral delivery in pooled screening. The goal of positive selection is to leave after selection only cells with survival-enhancing perturbation. The goal of the negative selection is to leave after selection only cells that could survive despite the perturbation.

The DNA is extracted, and PCR amplification is used to prepare samples for sequencing after selection.

Data analysis

The data analysis includes sequencing data processing, quality control, quantification of representation of each sgRNA, determination of the statistical significance of the changes in sgRNA representation and hit selection (Miles et al., 2016). The publicly available tools like edgeR (Y. Chen et al., 2015) or MAGeCK (Li et al., 2014) can be used to perform each step.

Validation

The goal of validation is to separate true positive hits from false positive screens. This can be done in a manner similar to RNAi screenings (secondary screening, orthogonal assay), however additional validation that the target gene was modified by Cas9 dependent DSB (for knockout libraries) or that transcription level changed due CRISPRi/CRISPRa methods should be performed.

Transcriptome profiling

Transcriptome profiling allows to study a cellular system at the transcriptome level. Transcriptome is the set of all RNA molecules in a particular cell. The most widespread techniques for transcriptome profiling are DNA microarrays and RNA-seq techniques (Lowe et al., 2017).

DNA microarray consists of a solid surface with thousands of microscopic spots attached to it. These spots contain probes, short specific (usually 25-60bp) DNA sequences, to which target fluorescently labeled complementary DNA (cDNA) hybridize via complementary base pairing between two DNA strands. Higher number of complementary bases means stronger bond between strands, that can stand washing step, when weaker non-specific bonding sequences are removed. The fluorescent label of cDNA generates signal for each spot. The strength of the signal depends on the amount of the cDNA bound to the probes. Comparing the strength of the signals from the same spot in two different conditions (e.g. control and experiment) microarrays allow to quantify relative changes in expression levels.

The two main types DNA microarrays from manufacturing point of view are spotted and in-situ microarrays. The probes for spotted microarrays are synthesized and stored in wells before placing on the array (DeRisi et al., 1996). Then they are placed at certain location using a robotic arm and spotting pins. This manufacturing technique allows to produce “in house” microarrays that can be highly customizable, since the set of probes can be specified for a particular experiment. On the other hand for in-situ DNA microarrays, probes are synthesized directly on the array surface, using for example photolithographic synthesis technique (Fodor et al., 1991). This technique relies on light-sensitive masking agents and one nucleotide is added at a time to applicable probes over the whole array. The applicable probes are unmasked using UV light allowing attachment of nucleotide from a solution of a single nucleotide.

DNA microarrays can be used for two-channel or one-channel detection. Two-channel microarrays can be hybridized with two samples simultaneously (e.g. control and experiment). The samples are labeled with two different fluorescent dyes which has

different fluorescence emission wavelengths. The microarray scanner excites the dyes with specific wavelengths and quantifies the relative intensities. The one-channel microarray can be hybridized with only one sample, so comparison between two samples requires two separate hybridizations.

A typical workflow of a DNA microarray experiment for measuring expression changes is:

1) Sample preparation. This step includes acquiring samples, extraction of the RNA, generating cDNA via reverse transcription, PCR amplification, labelling with fluorescent dye and mixing with hybridization solution.

2) Hybridization and washes. This step includes adding the mixture to the array, hybridization, washing away non-specific bonding sequences and drying.

3) Scanning. This step includes scanning the array with laser beam that excite fluorescent dye and quantifying the intensities of each spot.

4) Data analysis. This step includes normalization, quality control, filtering (spots with low intensity, noisy replicates, missing values) and statistical analysis. This can be done either by creating custom data analysis pipeline (e.g. using BioConductor (Gentleman et al., 2004) packages as affy (Gautier et al., 2004), affyPLM) or using integrated tools such as EXPANDER (Shamir et al., 2005) or Affymetrix Power Tools (APT).

RNA-seq refers to the combination of a high-throughput sequencing techniques with computational methods to measure the presence of transcripts in an RNA extract (Lowe et al., 2017).

A typical workflow of a RNA-seq experiment for measuring expression changes is:

1) Library preparation. During this step RNA is isolated from the cells. The isolated can be filtered to include only mRNA or ribosomal RNA (rRNA) can be depleted. The RNA

is converted to cDNA using reverse transcription and each fragment can also be amplified.

2) Sequencing. During this step the fragments are sequenced.

3) Data analysis. This step includes quality control of the raw sequences, alignment to reference genome (if known) or de novo assembly (if reference genome is unknown), quantification of transcripts expression and differential analysis between samples. These steps can be performed using such software tools as RNA workbench (Grüning et al., 2017) or RNA-seq workflow in Bioconductor (Love et al., 2015)

Computational analysis of functional genomics data

The result of functional genomics experiment is usually used either as a ranked list of biological components (genes, mRNAs, ..) where each gene is associated with an experimental score and a p-value or as a 'hit list' which contains only biological components with scores and p-values list with the scores resp. p-values that are above resp. below a certain threshold level. Numerous computational methods to interpret functional screening data and infer molecular machinery underlying the given phenotype can be grouped into two categories. The most widespread approaches can be grouped into two categories.

The first category is the pathway analysis methods that identify which canonical pathways or annotated gene sets are over- or underrepresented in a ranked list or a hit list. According to (Khatri et al., 2012), three generations of the pathway analysis approaches can be distinguished.

The first generation is **over representation analysis (ORA) approaches** which can be still considered as the gold standard approach for analysis of results of functional screenings. Typically, such methods include the following steps:

1) counting number of genes in the input list that belong to each pathway;
2) testing each pathway for over- or underrepresentation in the input list using statistical test such as hypergeometric, Fisher's exact, chi-square, or binomial tests.

The major limitations of such approaches are:

1) treating each gene in the input list equally, disregarding the experimental measurements;
2) using typically only hit list, disregarding other genes in the ranked list;
3) disregarding possible dependency between genes;
4) disregarding possible overlap between pathways.

Some examples of methods that implement this approach are ClueGO (Bindea et al., 2009), GOstat (Beißbarth et al., 2004), Enrichr (Kuleshov et al., 2016), David (Huang et al., 2009) and iGA (Breitling et al., 2004) tools.

ClueGO (Bindea et al., 2009) is a Cytoscape (Shannon et al., 2003) plugin which facilitates visualization, annotation, creation annotation networks and performing over representation analysis. It uses one- or two-sided tests based on the hypergeometric distribution and Gene Ontology(GO) terms as well as KEGG and BioCarta pathways databases as annotated gene sets.

Enrichr, David, GOstat are web based tools that uses either Fisher's exact test or chi-square tests to test overrepresentation.

Iterative Group Analysis (iGA) (Breitling et al., 2004) tool addresses the problem of using only the hit genes and disregarding others by using iterative approach. The tool uses ranked input list, assigns each gene to a class, e.g. based on GO term and determines the probability of change of each class by adding a gene at a time and finding the minimum probability to observe this many members of the class in the top of the list by chance [].

The second generation is **Functional Class Scoring (FCS) approaches** which aim at detecting coordinated impact of genes on a pathway. Typically, such methods include the following steps:

1) computing single pathway-level statistics using gene level statistics such as the Kolmogorov-Smirnov statistics, sum, mean of gene-level statistic;
2) testing the significance of the pathway-level statistics e.g. using phenotype or pathway permutation.

These approaches address the following limitations of the ORA approaches:

1) use experimental measurements;
2) use information from the whole ranked list;
3) take into account possible dependency between genes.

The major limitations of such approaches are:

1) treat each pathway independently;
2) usually consider only ranking but not exact values of experimental measurements.

Examples: GSEA (Subramanian et al., 2005), Enrichr (Kuleshov et al., 2016) tools.

Gene Set Enrichment Analysis (GSEA) is the first and one of the most popular tool implementing FSC approach. GSEA tool takes as an input a ranked list of genes and determines whether the members of a canonical pathway (or any other annotated gene set) are randomly distributed throughout the input list or mainly found in the top of the input list (Subramanian et al., 2005).

The three main steps of GSEA according to (Subramanian et al., 2005) are:

1) Calculation of an Enrichment Score (ES) for a pathway (or any other annotated gene set). This is done by starting in the top of the input list, moving down in the list increasing the running sum statistics if current gene belong to the pathway and decreasing otherwise. The increment is weighted using experimental measurements. ES is calculated as maximum of the sum across all genes in the input genes and it corresponds to a weighted Kolmogorov–Smirnov-like statistic.
2) Testing the significance of ES is done by permuting phenotype labels in the input list, calculating ES for permuted data and constructing distribution for ES. The p-value for the ES from step 1 is calculated using this distribution.
3) Adjusting p-value from step 3 for multiple hypothesis testing (if many canonical pathways are tested). This is done by normalizing each ES by the size of the pathway and computing permutation based False Discovery Rates (FDR).

Enrichr (Kuleshov et al., 2016) is another popular tool. Enrichr contains 125 frequently updated gene set libraries (divided into categories: transcription, pathways, ontologies, diseases/drugs, cell types and miscellaneous) and implements ORA using Fisher exact test as well as FSC method. The latter method takes as an input list of genes with coefficients from 0 to 1 that represent ‘the grade of membership’. The ‘grade of membership’ for a gene set is

defined as a sum of ‘gene grades’ and the statistical significance is calculated using hypergeometric distribution.

The third generation is **Pathway Topology (PT) approaches**. It is difficult to generalize about this category, but essentially, these approaches perform the same steps as FCS approaches, however for computing pathway level statistics they utilize not only experimental measurements but also topological information about pathways:

1) computing single pathway-level statistics using gene level statistics calculated using experimental measurements and topological information about pathways and/or molecular interactions information;
2) testing the significance of the pathway-level statistics.

The major limitations of such approaches are:

1) dependency on the pathway topology information which can be incomplete.

Examples: SPIA (Tarca et al., 2009), EnrichNet (Glaab et al., 2012).

Signaling pathway impact analysis (SPIA) method was proposed in 2009 by (Tarca et al., 2009). The method assesses the pathway significance using information from both standard over representation analysis and from possible abnormal perturbation of the pathway measured using expression changes of every gene in the pathway (the method was suggested for the analysis of transcriptome profiling data). To be able to quantify perturbation of the pathway, the authors first introduce a gene perturbation factor, which is calculated as a sum of the expression change of the gene and a normalized and weighted sum of perturbation factors of the upstream genes in the pathway, where the weights quantify the strength of the interactions in the pathway if such information is available. Next, the author introduced the net perturbation accumulation of every gene which is the perturbation factor minus expression change, and total net perturbation accumulation of entire pathway as a sum of net perturbation accumulation of every gene. Then, the p-value of getting a total perturbation accumulation of a pathway higher than given is calculated by permuting gene IDs in the pathway and constructing the null distribution. The total p-value

for a pathway is calculated as a probability to observe p-value from the standard ORA and p-value for perturbation accumulation as low as for a given pathway.

EnrichNet method was proposed in 2012 by (Glaab et al., 2012). The method takes a list of genes, annotated gene sets (e.g. canonical pathways) and a network of molecular interactions as input data. The method calculates the scores associated with the distance between the list of genes and annotated gene sets using a random walk with restart algorithm (RWR) and compares these scores with a background model. A random walk with restart algorithm enables to estimate the proximity of two nodes when each following step of the walk starting from a seed node on a graph is chosen randomly and the walk can be restarted with a certain probability. The authors argue that such distance measure better capture possible relationships between two nodes than for example the shortest path distance. Each annotated gene set is assigned a vector where each position is a RWR score from randomly chosen input gene to a gene in the pathway. Then, the deviation of these vectors from a background model, which is defined as “average distribution across all pathways” is assessed.

The second category of computational approaches for analysis and interpretation of the functional screening data is network analysis methods which usually use protein-protein interaction networks as complementary information. These methods aim at improving hit identification, finding functionally related biological components, finding significantly enriched/maximum weight subnetworks using ranked list or hit list. This is usually done by introducing network-based scoring methods that use network topology information and screening results. Below are examples.

(L. Wang et al., 2009) suggested a method implementing "guilt by association" principle and NePhe scoring system to address the problem of false positives and false negatives associated with RNAi screenings. The "guilt by association" principle was used in the sense that if a gene has many connections with other hit genes, then it is a true positive if it is a hit and false negative if it is not hit, and vice versa. The quantification of the connectedness between two genes was done using either direct neighbor information or the shortest path distance or diffusion kernels or association analysis-based transformations.

(Kaplow et al., 2009) suggested RNAiCut method to identify thresholds for ranked functional screening data by using connectivity of subgraphs of protein-protein interaction networks. The main hypothesis of the method is that true positive hits are densely interconnected in the protein-protein interactions networks (Kaplow et al., 2009). The method ranks the screening data and for each set of first k genes in the ranked list it computes the number of edges (connectivity) in the subgraph retrieved from PPI network. Then, the method estimates the p-value of getting such connectivity if the subnetwork was constructed of random nodes conserving the degree distribution. The number of genes k associated with the lowest p-value is used as the threshold.

(S. Ma et al., 2014) suggested NEST (Network Essentiality Scoring Tool) to assess gene essentiality and improve the quality of loss-of-function screening. The NEST score is calculated as a sum of expression values or scores from loss of function screenings information of neighbor genes in molecular network weighted by the interaction confidence. The authors showed the NEST scores are significantly higher for essential genes.

(Cornish et al., 2014) suggested a method called SANTA that also uses “guilt by association” principle to quantify association between a gene list and a network (including functionally annotates molecular networks). The authors introduced K^{net} and K^{node} scores that use adapted Ripley's K-Function for networks to measure network scores (K^{net}) and network-based nodes scores (K^{node}). The K^{net} scores assess quantity and strength of hits located in the proximity of each other in the network. This score can help to measure functional association of gene list with functionally annotates molecular networks. The K^{node} score quantify each node’s neighborhood for strong and close hits. By extracting nodes with the highest K^{node} score, significantly enriched for hits subnetworks can be identified.

(Kairov et al., 2012) suggested a method called OFTEN (Optimally Functionally Enriched Network) implemented in BiNoM (Bonnet et al., 2013) Cyroscape (Shannon et al., 2003) plugin to associate a ranked gene list with PPI subnetwork. The method aims at finding the largest connected component of a subnetwork formed by optimal number of top-ranked genes in the input list. The optimal number is identified by finding the number of top-ranked genes after which the percolation score rapidly goes down. The percolation score computes

using the size of the largest connected component of a subnetwork, the number of genes that were used to extract subnetwork and the mean value of the size of the largest connected component formed by randomly selected genes.

(Beisser et al., 2010) and (Dittrich et al., 2008) suggested a method called BioNet for finding maximum-weight connected subgraph which represent active subnetwork in the network. The method first annotates each gene with experimentally derived p-values. Then, these p-values are aggregated into one number (if several experimental conditions were tested simultaneously). The scoring function model both the signal and the noise component of the p-value using a beta-uniform mixture model, controlling false discovery rate. Then, the maximum-weight connected subgraph is found using linear programming algorithm.

Several other methods besides OFTEN and BioNet exist for finding active subnetworks. These are : jActiveModules (Ideker et al., 2002), ClustEx (Gu et al., 2010), PinnacleZ (Chuang et al., 2007), kwalk (Zheng et al., 2009), WMAXC (Amgalan et al., 2014), BMRF (L. Chen et al., 2013), COSINE (H. Ma et al., 2011). These methods as well as BioNet share the general idea: first, to introduce a subnetwork scoring function; then, to introduce an algorithm to find subnetwork with maximum or close to maximum score.

A completely different method one should mention is CausalR suggested by (Bradley et al., 2017) and (Chindelevitch et al., 2012). This method works with a network that consists of casual interactions that show the effect of the interaction between two proteins (inhibition or activation). It takes as starting points each gene (and indication whether it is up- or downregulated or there is no change) in a list of genes and finds points of convergence (usually upstream regulators) in the network that can explain the differential expression pattern seen in the starting points.

As for limitations, both pathway analysis and network analysis approaches have several general limitations. Firstly, pathway and network information can be incomplete and inaccurate. Secondly, both approaches do not give much information about the genes that are not included into pathways or networks. Lastly, networks are biased towards well studied genes.

2 Network analysis of functional genomics screening data

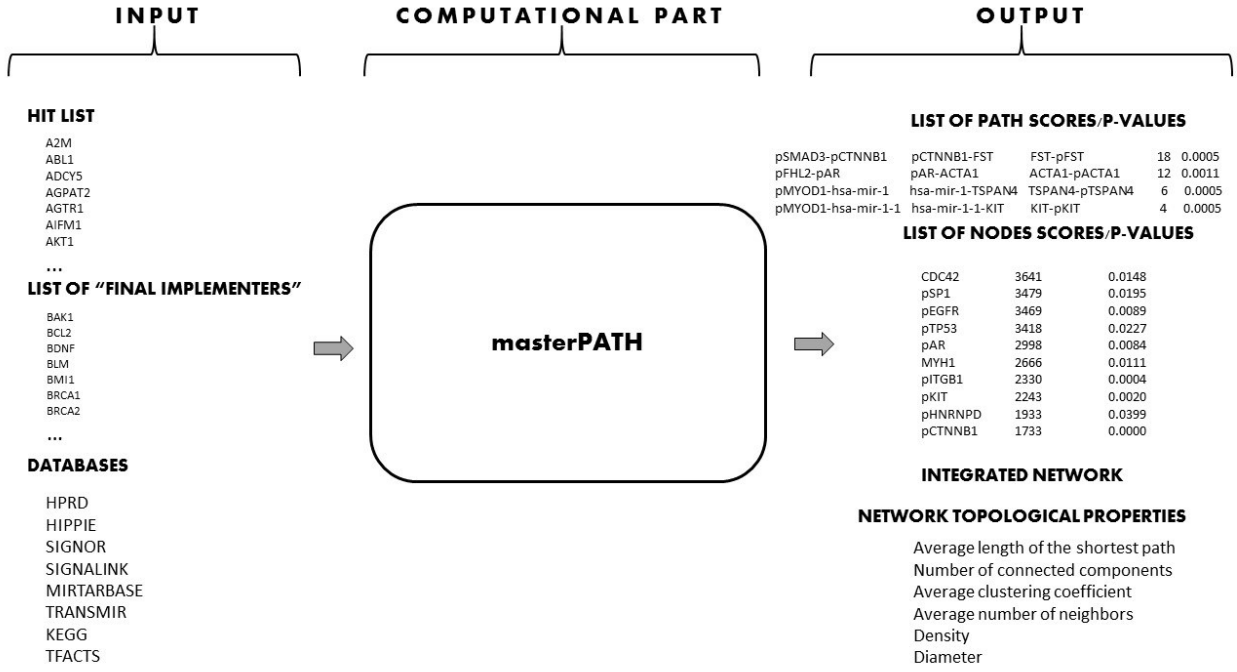
Some parts of this section are verbatim copy of two research papers being prepared for publication. One of them describes masterPATH method and its application to the analysis of the results of miRNA loss-of-function screening and transcriptomic profiling of terminal muscle differentiation and of 'druggable' loss-of-function screening of the DNA repair process; the other one is devoted to the study of the shortest path approach on the human interactome.

Here a new network analysis method to analyze functional genomics screening data – masterPATH – is presented. The method aims at elucidating members of molecular pathways leading to the studied phenotype using functional genomics experiments data in the hit list form. The method works on an integrated network that represent human interactome. The method constructs the integrated network from 8 databases: HPRD (Peri et al., 2003), HIPPIE(Schaefer et al., 2012), Signor (Surdo et al., 2017), SignaLink (Fazekas et al., 2013), tFacts (Essaghir et al., 2010), KEGG Metabolic Pathways (Ogata et al., 1999), transMir (J. Wang et al., 2009), mirTarBase (Hsu et al., 2011); and calculates basic topological properties. The method extracts subnetwork built from the shortest paths of 4 different types (with only protein-protein interactions, with at least one transcription interaction, with at least miRNA-mRNA interaction, with at least one metabolic interaction) between hit genes to so called "final implementers" - genes that are involved in molecular events responsible for final phenotypical realization (if known) or between hit genes (if "final implementers" are not known). The method calculates centrality score for each node and each linear path in the subnetwork as a number of the shortest paths found in the previous step that pass through the node and the linear path. Then, the statistical significance (p-value^{net}) of each centrality score is assessed by comparing it with centrality scores in subnetworks built from the shortest paths for randomly sampled hit lists (summarized in Figure 3-1). It is hypothesized that the nodes and the linear paths with statistically significant centrality score can be considered as putative members of molecular pathways leading to the studied phenotype. In case the experimental scores and p-values are available for a large number of nodes in the network, the method can also calculate experiment-based scores

(score^{exp}) and experiment-based p-values (p-values^{exp}) for the shortest paths or linear paths. The score^{exp} is calculated as an average of the experimental scores of the nodes in the path. The p-value^{exp} is calculating by aggregation of the p-values of the nodes in the path using Fisher's combined probability test and permutation approach.

The method is illustrated by analyzing the results of miRNA loss-of-function screening and transcriptomic profiling of terminal muscle differentiation and of 'druggable' loss-of-function screening of the DNA repair process. It is shown that known and new potentially interesting components for both biological systems are identified by the method.

Figure 2-1. Overview of masterPATH.



masterPATH algorithm

The following notions are used in the mechanistic model of pathway construction: an unweighted graph *G=(V, E)* represents a network of molecular interactions, where V are nodes that can be proteins, genes, small molecules or miRNAs; E are edges represent molecular interaction between nodes, interactions can be direct or undirect. List of hit genes

of size n is as a set $H = \{h_i : h_i \in V \text{ for } \forall i \in [1..n]\}$. List of "final implementers" of size m as a set $F = \{f_i : f_i \in V \text{ for } \forall I \in [1..m]\}$. A simple linear paths p between a pair of nodes $(v,u) : v,u \in V$ is as set of pairs of nodes that represent existing edge in the graph $G: p(v,u)= (v,v_1),(v_1,v_2) ... (v_n,u)$ where $v_i \in V \text{ for } \forall i$ and each node vi is distinct. Length L of the path $p(v,u)$ is the number of edges in the path p. We distinguish 4 different types of paths:

- protein-protein paths if all edges represent protein-protein interactions;
- transcriptional paths if there exist at least one edge that represent transcriptional interaction;
- miRNA paths if there exist at least one edge that represent miRNA-mRNA interaction;
- metabolic paths if there exist at least one edge that represent enzymatic reaction.

The algorithm of the method is the following. For a given network G, hit list H, list of "final implementers" F the method finds for each pair of hit gene and "final implementer" (h_i, f_j) all the shortest paths $\{p_i\}$ of four abovementioned types of length less or equal the maximum length L_{max} (defined by the user) in the network G. The search is done using breadth-first algorithm. Then the centrality score which resembles centrality score c is calculated for each node v and each path q (of length of several interactions) as a number of the shortest paths from $\{p_i\}$ that pass through the node v and the path q : $c(v)= |p \in \{p_i\} : v \in p|$; $c(q) = |p \in \{p_i\} : q \in p|$. After that, the statistical significance of each score is assessed. 10000 random hit lists are sampled from the set of nodes N preserving or not preserving the degree distribution of the initial hit lit. The probability ($p\text{-value}^{Net}$) of getting a node v or a path q with specific centrality score by chance is calculated as a proportion of sampled hit lists for which a node or a short path has the same or greater centrality score.

In case the experimental scores and p-values are available for the large number of nodes in the network G, experimental scores ($score^{exp}$) and p-values ($p\text{-value}^{exp}$) can also be calculated for the shortest path p, a path q, a node v. The $score^{exp}$ is calculated as an average of the absolute values of the experimental scores of the nodes in a path for a shortest path p_i and a path q; and equal to the absolute value of the experimental score for a node v. The

p-value^{exp} of the shortest path *p* and a path *q* is aggregated using Fisher's combined probability test. The path is considered as a set of unique nodes. First, the test statistic $X_{exp} = -2\sum_{i=1}^{m} \log p_i$is calculated for a path, where p_i is an experimental p-value of a node i. Then, a permutation test is used to assess the p-value of the X_{exp} (p-value^{exp}): the set of experimental p-values is shuffled many times (e.g. 1000), the test statistic $X_{perm}$is calculated for each shuffled sample, the p-value of the X_{exp} is calculated as a proportion of samples for which X_{perm} is at lease the same as X_{exp}. The p-value^{exp} a node *v* equal to the experimental p-value. The nodes without experimental values are not taken into account.

Databases

The following databases were used to construct integrated network: HIPPIE, Signor, SignaLink, tFacts, KEGG Metabolic pathways, transMir, mirTarBase; and protein-protein networks: Human Protein Reactions Database (HPRD) v9.1 and Human Integrated Protein-Protein Interaction rEference (HIPPIE) v2.0. All databases contain experimentally validated protein-protein interactions for human cells except SignaLink database which contains a small number of predicted miRNA-mRNA interactions. We distinguish high confidence, medium confidence and low confidence interactions is HIPPIE database using the confidence score from the database. The confidence score is assigned for each interaction in HIPPIE database and is calculated as a weighted sum of the number of studies in which an interaction was detected, the number and quality of experimental techniques used to measure an interaction and the number of non-human organisms in which an interaction was reproduced("HIPPIE Howto," n.d.). Predefined confidence levels by HIPPIE team were used to construct networks with only High Confidence interactions – HIPPIE HC (confidence level 0.73 - third quartile of the HIPPIE score) and with High and Medium Confidence interactions – HIPPIE HC+MC (confidence level 0.63 - second quartile of the score distribution).

All the databases use different types of gene ID. The IDs were converted to the HUGO gene nomenclature and this nomenclature was used to construct networks. Only interactions

between proteins, complexes and small molecules were used from Signor database. The information about the databases is summarized in Table 2-1.

Table 2-1. Description of the databases.

PPI stands for Protein-Protein Interactions, TF stands for Transcription Factor.

	Nodes	Interactions	Types of nodes	Types of interactions	Direction of interactions
HPRD	9 648	39 156	proteins	PPI	indirect
HIPPIE	16 403	237 958	proteins	PPI	indirect
HIPPIE (high and medium confidence)	15 857	193 576	proteins	PPI	indirect
HIPPIE (high confidence)	9368	41520	proteins	PPI	indirect
Signor	3977	13129	proteins, complexes, small molecules	PPI, enzymatic	direct, indirect
SignaLink	3285	27295	proteins, genes, miRNAs	PPI, miRNA-mRNA, TF-gene	direct, indirect
tFacts	2203	4312	TFs, genes	TF-gene	direct
KEGG metabolic pathways	2921	8231	proteins, small molecules	Enzymatic reactions	direct
transMir	324	647	TFs, miRNAs	TF-miRNA	direct
mirTarBase	2269	3511	miRNAs, genes	miRNA-mRNA	direct

Software implementation

The masterPATH method is implemented in Java program. Java code was developed using Java JDK 1.8, NetBeans IDE 8.0.2, GitHub version control systems and is available on GitHub page https://github.com/daggoo/masterPath.

The program consists of the following modules.

Database module:

- Loading databases from text/xml files.
- Unifying IDs (HGNC IDs).

Network module:

- Creating networks from databases.
- Extracting node/edge attributes from databases files.
- Merging networks.
- Loading lists hit genes and final implementers.
- Finding all length-bound pathways for two lists of nodes.

Pathway module:

- Finding the shortest paths of 4 different types for a set of pathways.
- Calculating $scores^{exp}$ and $p\text{-}values^{exp}$ for a set of pathways.
- Filtering pathways.
- Creating files for Cytoscape Software (network visualization).
- Finding the 'strongest' paths using $scores^{exp}$ and $p\text{-}values^{exp}$.

Centrality module:

- Calculating nodes' and paths' centrality scores.
- Calculating paths' degree.
- Calculating nodes' and paths' $scores^{exp}$ and $p\text{-}values^{exp}$.
- Filtering paths.

Random module:

- Sampling random hit lists preserving or not degree distribution.
- Generating random networks preserving degree distribution.
- Calculating p-values^{net}.

Network topology module:

- Calculating basic topological properties for networks (average clustering coefficient, number of connected components, average number of neighbors, density, diameter, average length of the shortest path).

The detailed description for masterPATH Java code in the form of JavaDoc created with the help of the Doxygen program is in Appendix 2.

The study of the shortest path approach

This chapter is devoted to the study whether the shortest path approach gives valid molecular paths. In particular, does at least one of the shortest paths between biologically meaningful start and end points built in the cell interactome represents a valid molecular path and, if not, does the analysis of the centrality measure introduced in the previous chapters help at least partially reconstruct valid molecular path?

To do this parts of the NF-kappa B, MAPK, Jak-STAT, mTOR, ErbB, Wnt, TGF-beta signaling pathways, the signaling part of the apoptotic process canonical pathways from KEGG database (Ogata et al., 1999) that consist only of protein-protein interactions were taken for the analysis (Figures 1-8).

46 simple linear protein-protein paths of length from 2 to 5 interactions were found for 8 canonical pathways (Table 2-2). All possible combination of source and target points preserving the direction of signal transduction were found on these 55 paths that gave 245 paths of the length from 2 to 5 interactions: 91 paths of the length 2 of interactions, 74 paths of the length of 3 interactions, 51 paths of the length of 4 interactions and 29 paths of the length of 5 interactions.

Figure 2-2. NF-κB signal transduction pathway.

Nuclear factor-κB proteins are transcription factors that regulate wide range of biological processes including immunity, stress responses, inflammation and cell survival. The PPI part is highlighted with black line. Adapted from KEGG database http://www.genome.jp/kegg-bin/show_pathway?hsa04064

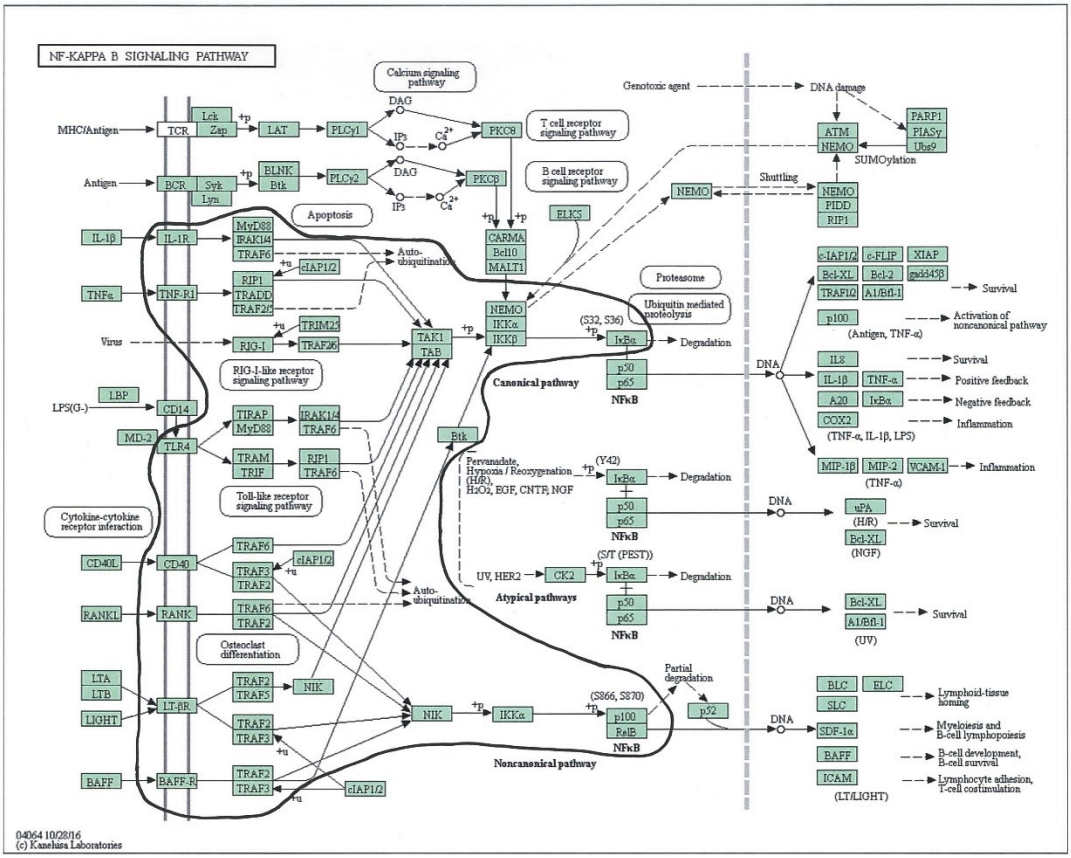


Figure 2-3. MAPK signal transduction pathway.

Mitogen-activated protein kinase (MAPK) cascade is a conserved pathway that is involved in many cellular functions such as cell proliferation, differentiation and migration. The PPI part is highlighted with black line. Adapted from KEGG database http://www.genome.jp/kegg-bin/show_pathway?hsa04010

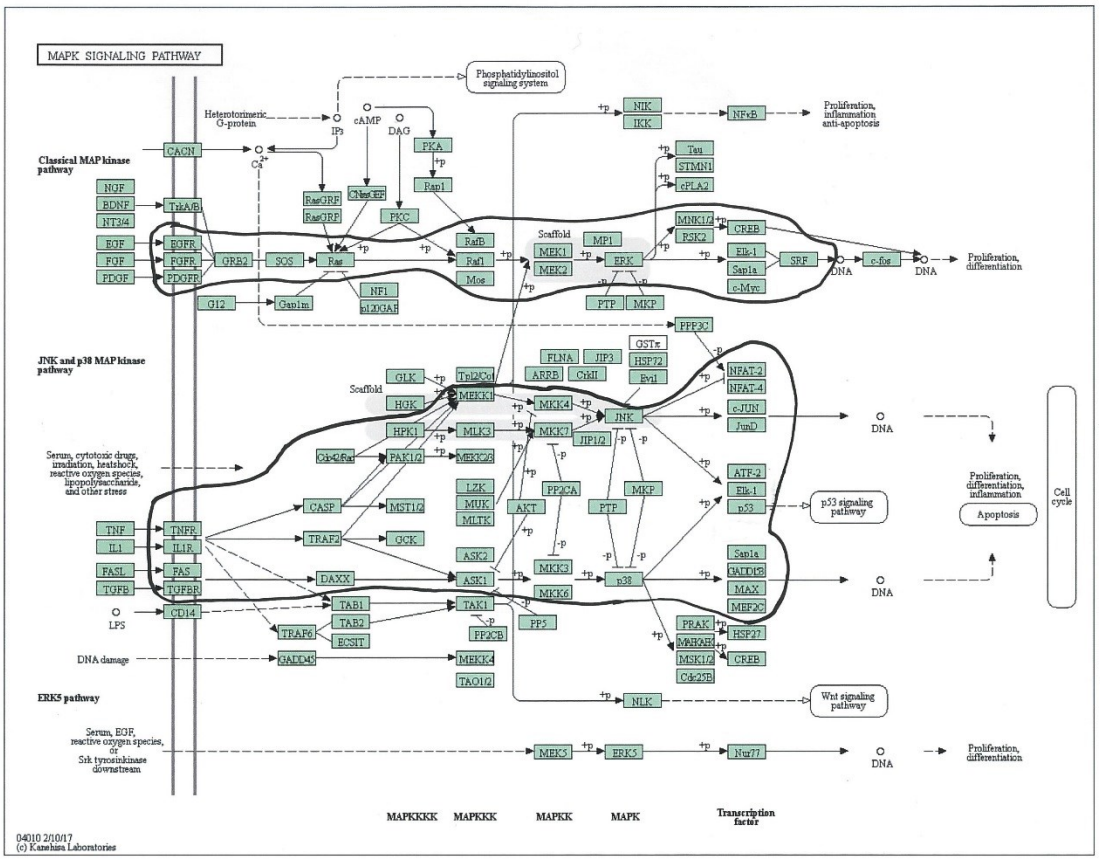


Figure 2-4. JAK/STAT signal transduction pathway.

The Janus kinase/signal transducers and activators of transcription (JAK/STAT) signaling pathway is the main signaling pathway for a wide range of cytokines and growth factors important for development and homeostasis. The PPI part is highlighted with black line. Adapted from KEGG database http://www.genome.jp/kegg-bin/show_pathway?hsa04630

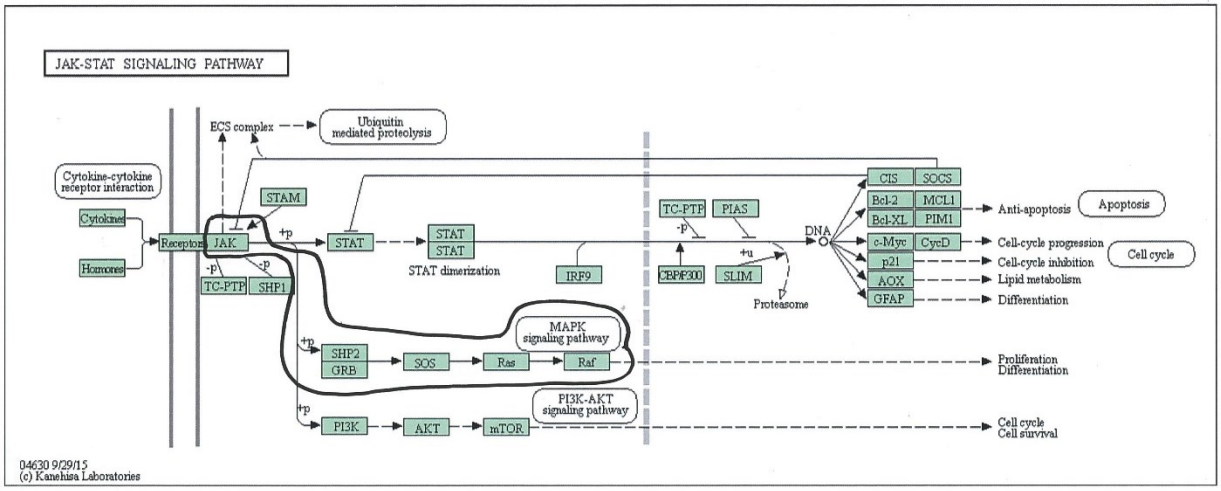


Figure 2-5 mTOR signal transduction pathway.

The mammalian target of rapamycin (mTOR) signaling pathway regulates many biological processes, including lipid metabolism, autophagy, protein synthesis, ribosome biogenesis. The PPI part is highlighted with black line. Adapted from KEGG database http://www.genome.jp/kegg-bin/show_pathway?hsa04150

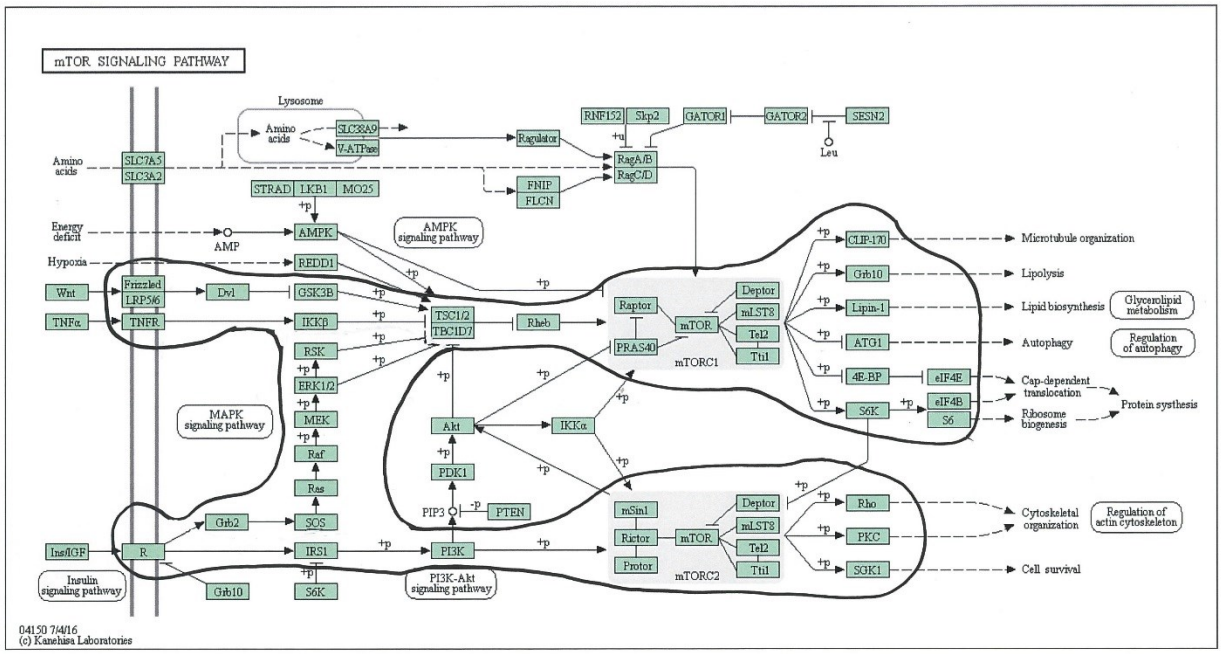


Figure 2-6 ErbB signal transduction pathway.

The ErbB signaling pathway regulates many biological processes, including proliferation, differentiation, cell motility, and survival. The PPI part is highlighted with black line. Adapted from KEGG database http://www.genome.jp/kegg-bin/show_pathway?hsa04012

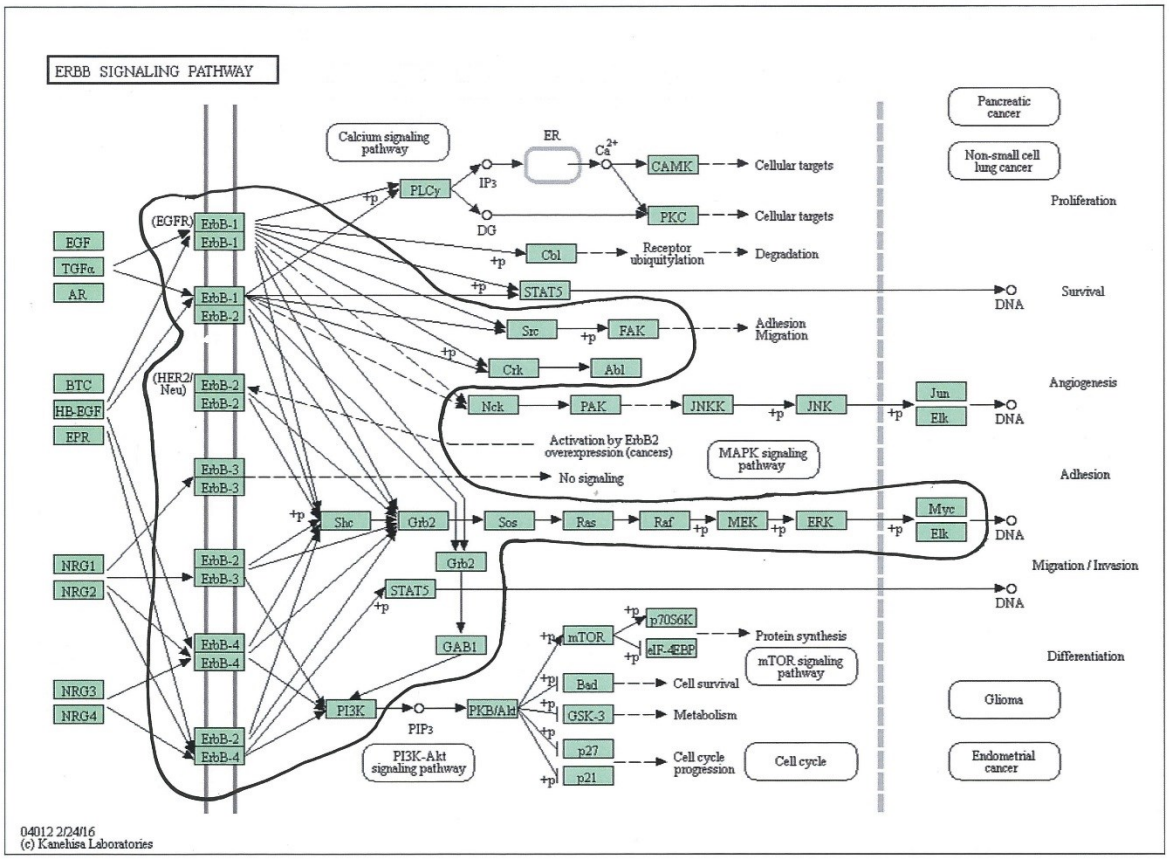


Figure 2-7. Wnt signal transduction pathway.

The Wnt signaling pathway regulates many biological processes, including regulation of gene transcription, remodeling of the cytoskeleton, cell adhesion and motility, cytoplasmic calcium. The PPI part is highlighted with dotted line. Source: KEGG database http://www.genome.jp/kegg-bin/show_pathway?hsa04310

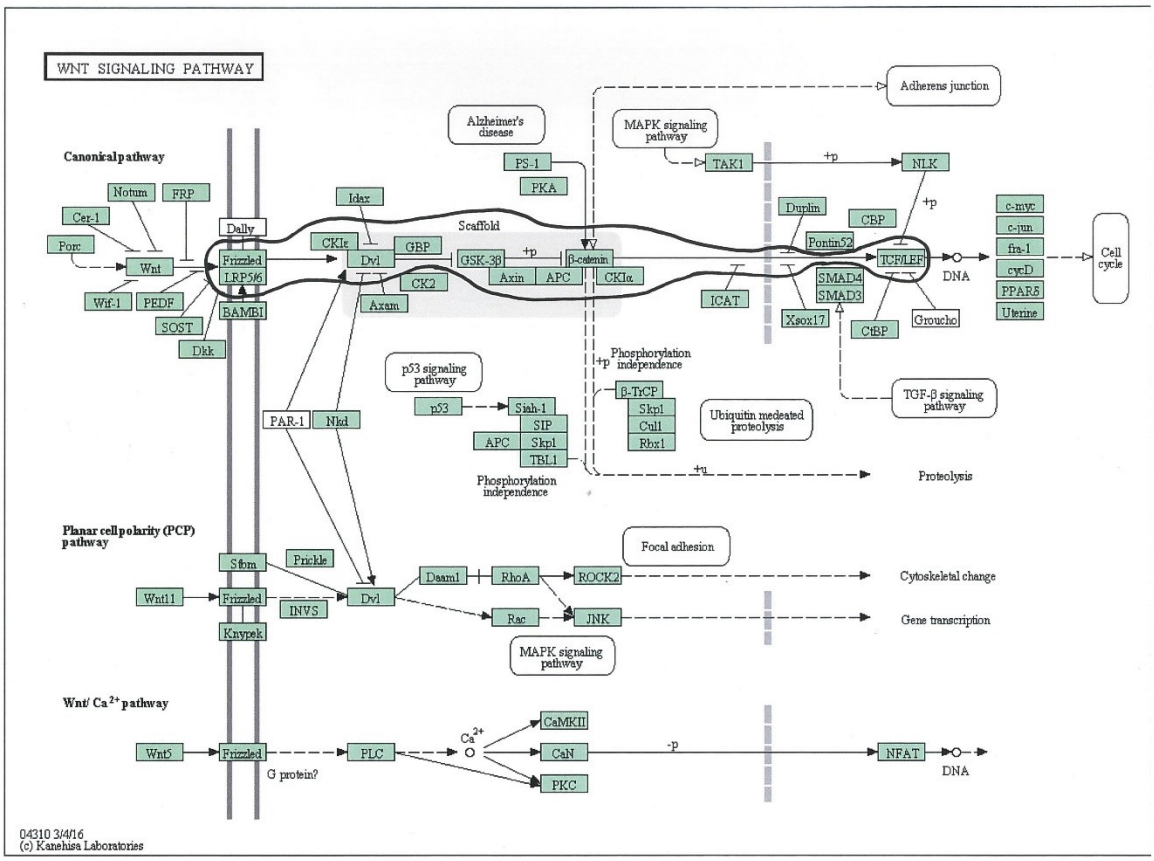


Figure 2-8. TGF-beta signal transduction pathway.

The transforming growth factor-beta (TGF-beta) signaling pathway regulates many biological processes, including proliferation, apoptosis, differentiation and migration. The PPI part is highlighted with black line. Adapted from KEGG database http://www.genome.jp/kegg-bin/show_pathway?hsa04350

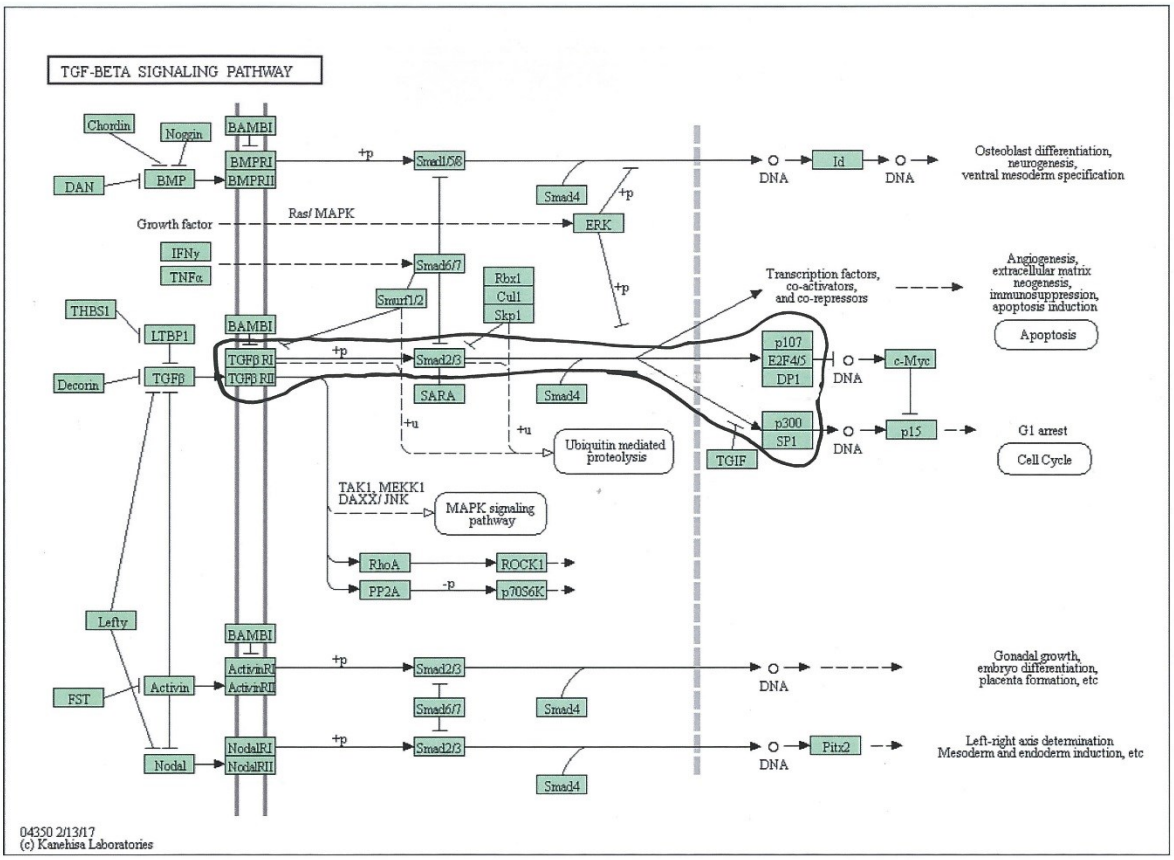


Figure 2-9. Apoptotic pathway.

Apoptotic extrinsic and intrinsic pathways regulate mechanisms of activation of effector caspases, such as caspase-3 and caspase-7. The PPI part is highlighted with black line. Adapted from KEGG database http://www.genome.jp/kegg-bin/show_pathway?hsa04210

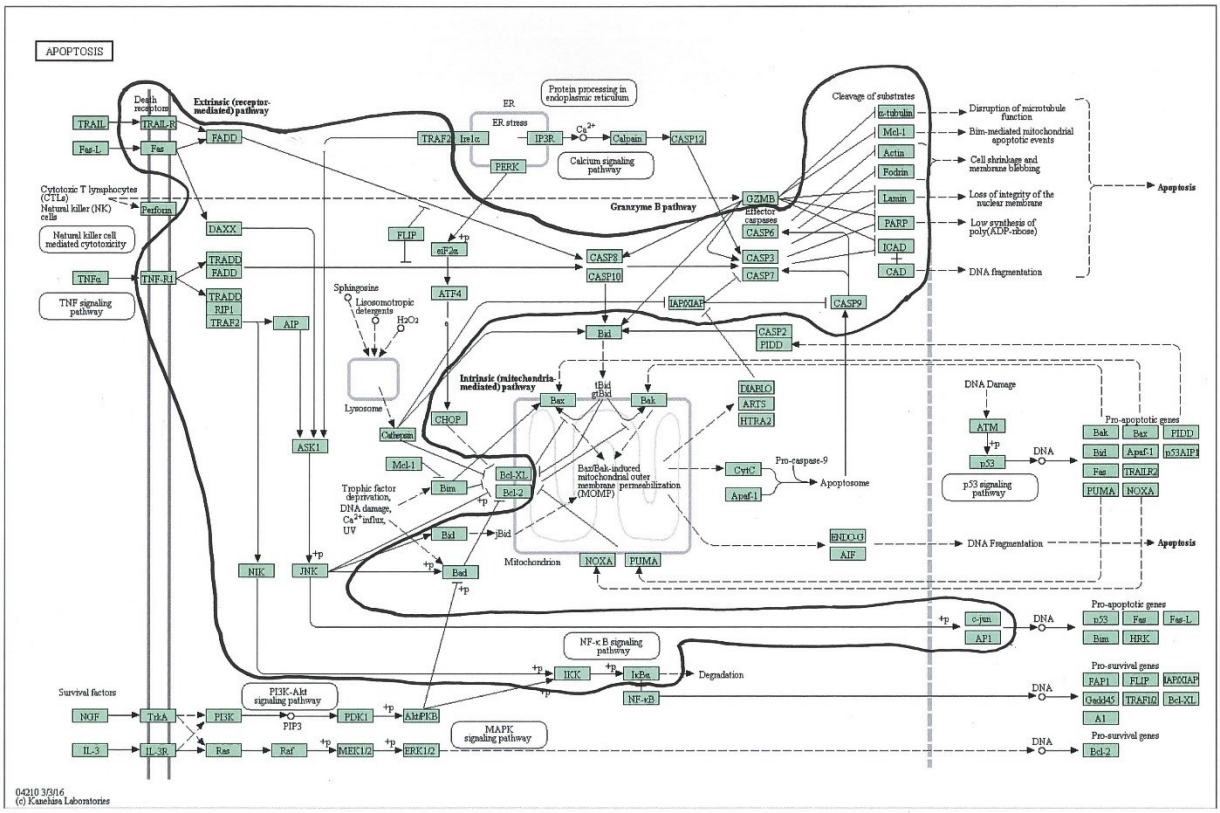


Table 2-2. 46 simple linear protein-protein paths of length 2-5 interactions found for 8 canonical pathways.

NFKB

NFRSF13C-TRAF2;TRAF3-BTK-IKBKG;CHUK;IKBKB-NFKBIA

NFRSF11A-TRAF6;TRAF2-TAB1;TAB2;TAB3;MAP3K7-IKBKG;CHUK;IKBKB-NFKBIA

NFRSF11A-TRAF6;TRAF2-MAP3K14-CHUK-NFKB2;RELB

TBR-TRAF2;TRAF5-MAP3K14-TAB1;TAB2;TAB3;MAP3K7-IKBKG;CHUK;IKBKB-NFKBIA

TBR-TRAF2;TRAF3-MAP3K14-CHUK-NFKB2;RELB

D40-TRAF2;TRAF3-MAP3K14-CHUK-NFKB2;RELB

D40-TRAF6-TAB1;TAB2;TAB3;MAP3K7-IKBKG;CHUK;IKBKB-NFKBIA

LR4-TICAM2;TICAM1-RIPK1;TRAF6-TAB1;TAB2;TAB3;MAP3K7-IKBKG;CHUK;IKBKB-NFKBIA

LR4-TIRAP;MYD88-IRAK1;IRAK4;TRAF6-TAB1;TAB2;TAB3;MAP3K7-IKBKG;CHUK;IKBKB-NFKBIA

DX58-TRAF2;TRAF6-TAB1;TAB2;TAB3;MAP3K7-IKBKG;CHUK;IKBKB-NFKBIA

NFRSF1A-RIPK1;TRADD;TRAF2;TRAF5-TAB1;TAB2;TAB3;MAP3K7-IKBKG;CHUK;IKBKB-NFKBIA

L1R1-MYD88;IRAK1;IRAK4;TRAF6-TAB1;TAB2;TAB3;MAP3K7-IKBKG;CHUK;IKBKB-NFKBIA

MAPK

SF1R;EGFR;EPHA2;FGFR1;FGFR2;FGFR2;FGFR4;FLT1;FLT3;FLT4;IGF1R;INSR;KDR;KIT;NGFR;NTRK1;NTRK2;PDGFRA;PDGFRB;

-GRB2-SOS1;SOS2-RRAS2;MRAS;HRAS;KRAS;NRAS;RRAS-BRAF;RAF1;ARAF-MAP2K1;MAP2K2-MAPK1;MAPK3-

MKNK2;MKNK1;RPS6KA6;RPS6KA1;RPS6KA2;RPS6KA3-ATF4

SF1R;EGFR;EPHA2;FGFR1;FGFR2;FGFR2;FGFR4;FLT1;FLT3;FLT4;IGF1R;INSR;KDR;KIT;NGFR;NTRK1;NTRK2;PDGFRA;PDGFRB;

-GRB2-SOS1;SOS2-RRAS2;MRAS;HRAS;KRAS;NRAS;RRAS-BRAF;RAF1;ARAF-MAP2K1;MAP2K2-MAPK1;MAPK3-

LK1;ELK4;MYC-SRF

NFRSF1A-TRAF2-MAP3K5-MAP2K3;MAP2K6-MAPK14;MAPK11;MAPK13;MAPK12-ELK4;DDIT3


0;FRZD2;FZD5;FZD3;FZD4;FZD6;FZD7;FZD8;FZD9;LRP6;LRP5-DVL1;DVL2;DVL3-GSK3B-TSC1;TSC2;TBC1D7-RHEB-
1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-CLIP1
0;FRZD2;FZD5;FZD3;FZD4;FZD6;FZD7;FZD8;FZD9;LRP6;LRP5-DVL1;DVL2;DVL3-GSK3B-TSC1;TSC2;TBC1D7-RHEB-
1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-GRB10
0;FRZD2;FZD5;FZD3;FZD4;FZD6;FZD7;FZD8;FZD9;LRP6;LRP5-DVL1;DVL2;DVL3-GSK3B-TSC1;TSC2;TBC1D7-RHEB-
1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-LPIN1
0;FRZD2;FZD5;FZD3;FZD4;FZD6;FZD7;FZD8;FZD9;LRP6;LRP5-DVL1;DVL2;DVL3-GSK3B-TSC1;TSC2;TBC1D7-RHEB-
1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-EIF4EBP1-EIF4E;EIF4E1B;EIF4E2
0;FRZD2;FZD5;FZD3;FZD4;FZD6;FZD7;FZD8;FZD9;LRP6;LRP5-DVL1;DVL2;DVL3-GSK3B-TSC1;TSC2;TBC1D7-RHEB-
1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-RPS6KB1;RPS6KB2-EIF4B;RPS6
KBKB-TSC1;TSC2;TBC1D7-RHEB-RPTOR;AKT1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-CLIP1
KBKB-TSC1;TSC2;TBC1D7-RHEB-RPTOR;AKT1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-GRB10
KBKB-TSC1;TSC2;TBC1D7-RHEB-RPTOR;AKT1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-LPIN1
KBKB-TSC1;TSC2;TBC1D7-RHEB-RPTOR;AKT1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-EIF4EBP1-EIF4E;EIF4E1B;EIF4E2
KBKB-TSC1;TSC2;TBC1D7-RHEB-RPTOR;AKT1S1;MTOR;DEPTOR;MLST8;TELO2;TTI1-RPS6KB1;RPS6KB2-EIF4B;RPS6
IRS1-PIK3CA;PIK3CB;PIK3D;PIK3R1;PIK3R2;PIK3R3-MAPKAP1;RICTOR;PRR5;MTOR;DEPTOR;MLST8;TELO2;TTI1-RHOA
IRS1-PIK3CA;PIK3CB;PIK3D;PIK3R1;PIK3R2;PIK3R3-MAPKAP1;RICTOR;PRR5;MTOR;DEPTOR;MLST8;TELO2;TTI1-PRKCA;PRKC
IRS1-PIK3CA;PIK3CB;PIK3D;PIK3R1;PIK3R2;PIK3R3-MAPKAP1;RICTOR;PRR5;MTOR;DEPTOR;MLST8;TELO2;TTI1-SGK1

FBR2-SMAD2;SMAD3;ZFYVE9;ZFYVE16-CREBBP;EP300;SP1

FBR2-SMAD2;SMAD3;ZFYVE9;ZFYVE16-RBL1;E2F4;E2F5;TFDP1

S

;TNFRSF10C;TNFRSF10B;TNFRSF10A-FADD-CASP8;CASP10-CASP3;CASP7

CASP8;CASP10-CASP3;CASP7

FADD;TRADD-CASP8;CASP10-CASP3;CASP7

TRADD;RIPK1;TRAF2-DAB2IP-MAP3K5-MAPK8;MAPK9;MAPK10-JUN;FOS

TRADD;RIPK1;TRAF2-DAB2IP-MAP3K5-MAPK8;MAPK9;MAPK10-BCL2L1;BCL2

MAP3K5-MAPK8;MAPK9;MAPK10-JUN;FOS

MAP3K5-MAPK8;MAPK9;MAPK10-BCL2L1;BCL2

3K5-MAPK8;MAPK9;MAPK10-JUN;FOS

3K5-MAPK8;MAPK9;MAPK10-BCL2L1;BCL2

CTSD;CTSH;CTSK;CTSL;CTSV;CTSO;CTSS;CTSW;CTSZ;CTSF-BIRC2;BIRC3;BIRC5;XIAP-CASP9-CASP3;CASP7;CASP6

The following steps were performed:

1. The shortest paths were found for each source and target points. If a source or a target point was a complex or implies several homologs all proteins were taken.

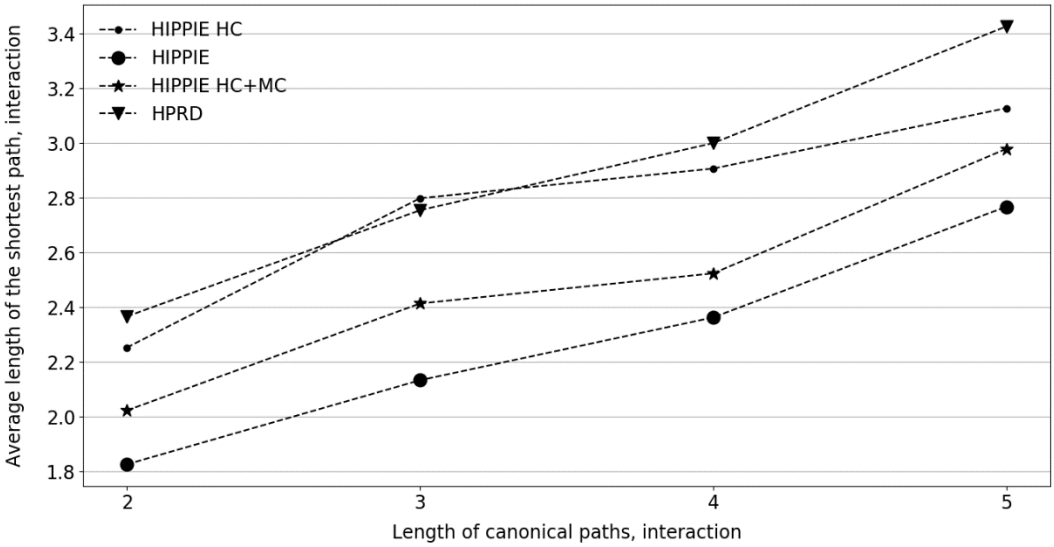
2. The average length of the shortest counterparts was calculated for the canonical paths of length 2, 3, 4, 5.

3. The canonical paths were compared with the shortest counterparts.

4. The centrality score was calculated for each node/path in the subnetwork constructed from the shortest paths (the shortest paths subnetwork) for each source/target points. The number of nodes/paths with centrality score ≥ 2 that are members of the canonical path were calculated.

It was found that the average length of the shortest paths is less by at least 1 interaction for the canonical paths of length of 4 and 5 interactions (Figure 2-10) and but is less than the average length of the shortest path (ALSP) in the network (ALSP HPRD = 4.2, ALSP HIPPIE=3.2, ALSP HIPPIE HC + MC=3.3, ALSP HIPPIE HC=4.1)

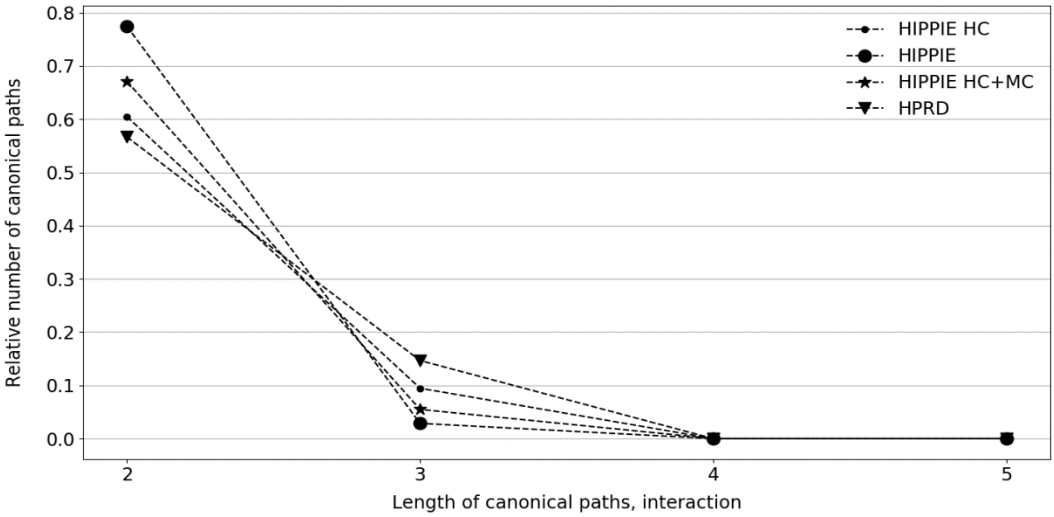
Figure 2-10. Average length in number of interactions of the shortest paths.



Next, it was checked whether one of the paths in the set of the shortest paths will match the canonical counterpart. It was found that the set of the shortest paths contains the full

canonical counterpart only for 56-77% of the canonical paths of length of 2 interactions, for 15-30% of the canonical paths of length of 3 interactions and for 0% of the canonical paths of length of 4 and 5 interactions (Figure 2-11).

Figure 2-11. Relative number of canonical paths for which the set of the shortest paths contains the canonical counterpart.



Next, the number of nodes/paths with centrality score ≥ 2 that are members of the canonical path and the number of start/end points for which at least one node/path have centrality score ≥ 1 (that means there are overlapping paths) were calculated. The result is presented in Figure 2-12 (nodes) and Figure 2-13 (paths). The numbers near the points shows how many canonical paths have at least one node/path with centrality score ≥ 1 in the shortest paths subnetwork.

Figure 2-12. Relative number of the canonical paths for which at least one node with centrality score ≥ 2 in the shortest paths subnetwork belong to the canonical path.

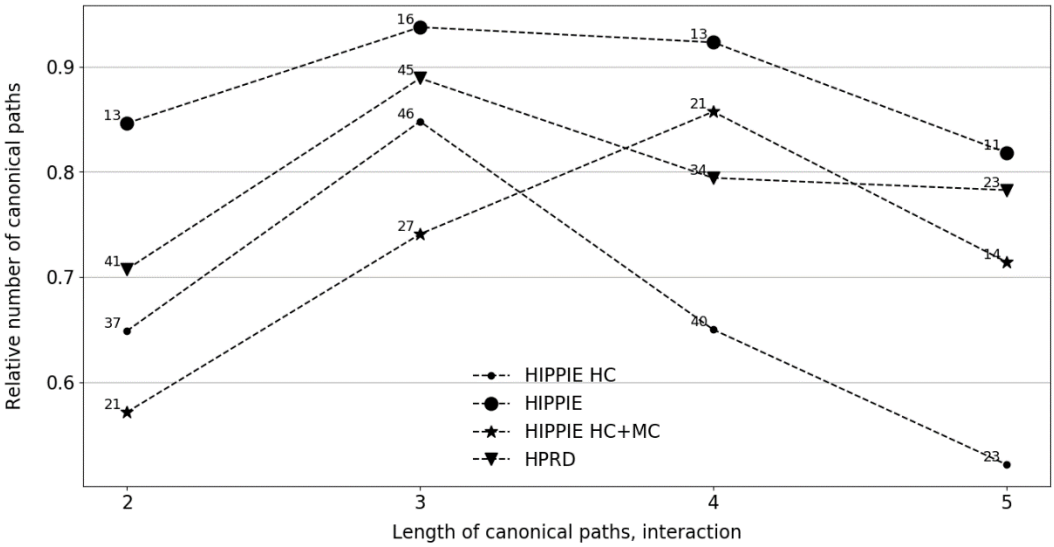
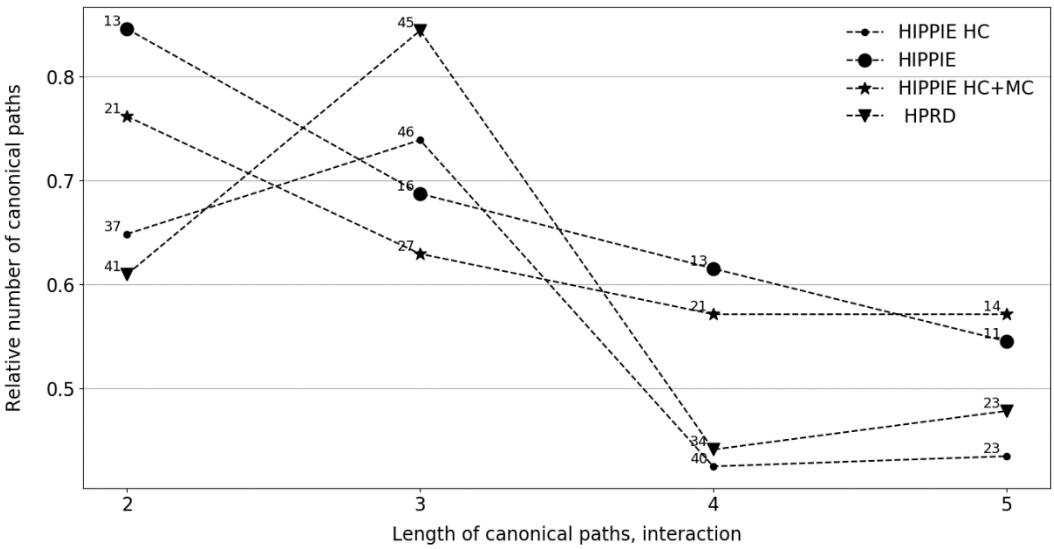


Figure 2-13. Relative number of canonical paths for which at least one path with centrality score ≥ 2 in the shortest paths subnetwork belong to the canonical path.



Topological properties of the PPI and the integrated networks

Table 2-3 gives the information about the basic topological properties for the PPI and integrated networks. The exponent of the fitted power-law distribution in the degree distribution was calculated with powerlaw (Alstott et al., 2014) Python package.

Table 2-3**.** Topological properties of the PPI and the integrated networks.

	HPRD	HIPPIE	HIPPIE HC+MC	HIPPIE HC	Integrated network
Nodes	9 648	16 403	15 857	9 365	13 419
Interactions	39 156	237 958	193 576	41 329	85 277
Average clustering coefficient	0.2	0.26	0.23	0.31	0.11
Number of connected components	112	9	17	121	126
Average number of neighbors	7.8	28.6	24	8.6	9.2
Density	0.0008	0.002	0.002	0.0009	0.0007
Diameter	14	8	8	12	25
Average length of the shortest path	4.2	3.2	3.3	4.1	4.5
Exponent of fitted power-law distribution (total degree)	2.71	3.31	2.62	2.74	3.16

Exponent of fitted power-law distribution (in degree)	3.93
Exponent of fitted power-law distribution (out degree)	2.36

Application: human muscle differentiation miRNA loss-of-function screening and transcriptome profiling

The screening data from the study by A. Polesskaya et al. (Polesskaya et al., 2013) was taken as the hit list for terminal human muscle differentiation process. In this study, genome-wide miRNA loss-of-function screening on late differentiating human muscle precursor cell line (LHCN) was performed in two-step approach. The primary screening was done in duplicate with LNA antisense inhibitors library targeting 870 miRNAs and a readout assay that detects multinucleated Myosin Heavy Chain (MHC) positive myotubes. Those miRNAs whose depletion resulted in differences to the negative control ≥ 2 SD were selected for the secondary screen which was done in triplicate. The total number of nuclei was checked in addition to the readout assay from the primary screen. A total of 63 miRNAs whose depletion resulted in differences to the negative control ≥ 2 SD were confirmed in the secondary screen (Table 2-4).

Table 2-4. miRNA loss-of-function screening: hit list.

hsa-mir-100	hsa-mir-130b	hsa-mir-326
hsa-mir-106a	hsa-mir-138-1	hsa-mir-331-3p
hsa-mir-17	hsa-mir-145	hsa-mir-339-5p
hsa-mir-1227	hsa-mir-1538	hsa-mir-365
hsa-mir-1233	hsa-mir-18b	hsa-mir-429
hsa-mir-125b	hsa-mir-223	hsa-mir-454
hsa-mir-1267	hsa-mir-296-5p	hsa-mir-445-3p

hsa-mir-484	hsa-mir-1224-3p	hsa-mir-346
hsa-mir-485-3p	hsa-mir-1228	hsa-mir-361-3p
hsa-mir-501-3p	hsa-mir-1249	hsa-mir-483-3p
hsa-mir-512-3p	hsa-mir-125a-5p	hsa-mir-486-3p
hsa-mir-532-3p	hsa-mir-1260	hsa-mir-574-3p
hsa-mir-541	hsa-mir-1280	hsa-mir-629
hsa-mir-600	hsa-mir-129-3p	hsa-mir-885-5p
hsa-mir-625	hsa-mir-1296	hsa-mir-193b
hsa-mir-636	hsa-mir-133a	hsa-mir-369-3p
hsa-mir-663	hsa-mir-133b	hsa-mir-381
hsa-mir-664	hsa-mir-150	hsa-mir-886-5p
hsa-mir-766	hsa-mir-197	hsa-mir-940
hsa-mir-770-5p	hsa-mir-204	hsa-mir-98
hsa-mir-93	hsa-mir-328	hsa-mir-631
hsa-let-7b	hsa-mir-342-3p	

The screening data from the study by J. Kropp et al. (Kropp et al., 2015) was taken as the second hit list for terminal human muscle differentiation process. Transcriptomic profiling for proliferation and late differentiation stages in human muscle precursor cell line (LHCN) was performed using Affymetrix Human Gene 1.1 ST arrays (Kropp et al., 2015). A total of 571 differentially expressed genes during late differentiation compared to proliferation stage were found with 2-fold change and p-value≤0.05 threshold (Table 2-5).

11 proteins were used as a list of “final implementers”. These are MSTN, IGF2, ACTA1, MYH1, MYLPF, ARF6, CD81, CD9, CDC42, EHD2, MYOF proteins responsible for *inhibition*, *activation*, and *facilitating* of fusion of myotubes (Alzhanov et al., 2010; Bourmoum et al., 2016; Doherty et al., 2008; Gunning et al., 2001; S.-J. Lee, 2004; Tachibana et al., 1999; Vasyutina et al., 2009; Y. Wang et al., 2007). The analysis was performed in the integrated human interactome.

Table 2-5. Transcriptome profiling: hit list.

ABCA1	B3GALT5	CASQ2	CRIM1	FBN1	HES6
ABCC3	BAIAP2L1	CASS4	CTNNAL1	FBXL22	HEY1
ABHD5	BASP1	CASZ1	CTSH	FGF1	HFE2
ABLIM3	BBS12	CBFB	CUEDC1	FGF2	HIP1
ACCN2	BCAM	CCDC102B	CUZD1	FGFR4	HIPK3
ACOT11	BCHE	CCDC99	CYB5R1	FGL2	HIST1H3J
ACTA1	BCL2L11	CCNA2	DAAM2	FILIP1	HIST2H4A
ADAM9	BCL6	CCND3	DAG1	FITM1	HIVEP1
ADAMTS6	BCL7A	CD36	DBH	FLVCR2	HOMER2
ADAMTSL1	BDKRB1	CD68	DBNDD1	FMN2	HOXA7
ADAMTSL5	BEND7	CDC25B	DCBLD1	FMNL1	HRC
AFAP1L1	BHLHE41	CDH2	DCP2	FMO3	HSD11B2
AFAP1L2	BIN1	CDK1	DDI2	FNDC5	HSD17B10
AGPAT5	BIRC5	CDK15	DGKD	FOXM1	HSPA4L
AGRN	BLCAP	CDKN1C	DIAPH3	FRK	HSPB8
AIF1L	BNC1	CDKN3	DIO2	FRMPD1	ID1
AIM1	BNIP3	CDON	DLG2	FSTL3	ID2
AKAP1	BTG1	CELA2B	DLL1	FXYD1	ID3
ALCAM	BTG3	CENPA	DMBT1	FXYD5	IFFO1
AMIGO2	C10orf10	CENPE	DNAJB1	FXYD6	IGF1
ANK2	C10orf72	CGB	DNASE1L1	FYCO1	IGFBP6
ANKRD10	C11orf71	CGB1	DOCK10	G6PD	IL1R1
ANKRD13A	C12orf34	CGB2	DOCK11	GAB2	IL34
ANKS1A	C14orf159	CGB5	DOCK5	GADD45G	IMPA2
APOBEC2	C14orf43	CGB7	DPT	GALNT5	IMPAD1
APOBEC3C	C1S	CHCHD8	DSP	GATM	INHBA
APOC1	C1orf105	CHD7	DTNA	GCNT1	INPP4B
APOD	C2	CHIT1	EAF1	GCOM1	INPPL1
APOE	C20orf103	CHRNB1	EBF1	GGH	ITGA10
AQP3	C21orf119	CHRND	ECH1	GGT5	ITGA7
ARHGAP9	C21orf33	CHRNG	ECM1	GLIPR1	ITGB1BP2
ARHGEF2	C21orf59	CIAO1	EDN1	GLRX	ITGB1BP3
ARHGEF3	C21orf88	CKB	EFHD2	GMPR	ITM2C
ARHGEF6	C22orf13	CKM	EGLN3	GNAZ	KCNA7
ARPC1B	C3	CLCA2	ELN	GPM6B	KCNIP3
ARPP21	C3orf39	CLCN5	ENO3	GPNMB	KCNJ15
ASNS	C4A	CLSTN2	ENPP1	GPR1	KCNJ2
ASPM	C7orf55	COL5A3	ENPP2	GPR125	KCNN3
ASS1	C7orf58	COL6A2	EPHB2	GPR153	KCNQ4
ASTN2	C9orf46	COL6A3	ERBB3	GPR155	KDELR3
ATOH8	CABLES1	CORO2B	ERO1L	GPR176	KDM4B
ATP10A	CACNA1S	CORO7	EZR	GREM1	KDM6B
ATP1A1	CACNA2D1	CPA4	FAIM2	GSTT2	KIAA1632
ATP6V0D1	CACNB1	CPNE2	FAM101B	GTF2F2	KIAA1644
ATP6V0E2	CACNG1	CPS1	FAM171A2	GUSB	KIF13B
ATP8B1	CAMK2D	CREB3L1	FAM174A	H1F0	KIF1B
ATXN7L1	CASP9	CREG1	FAM65B	HES1	KIF20A

KLHL28
KREMEN1
KY
LAMA4
LAMA5
LAPTM5
LDB3
LDLRAD3
LETMD1
LMBRD2
LMOD3
LNX1
LOC645166
LOC654342
LOX
LRIG1
LRRC16A
LRRC17
LRRFIP1
LRRN1
LRRN4CL
LSP1
LSS
MACF1
MACROD1
MAMLD1
MAN2A2
MAP3K14
MAPRE3
MDM4
MEF2D
MEGF10
MEOX1
MFAP4
MFAP5
MGC39372
MICAL1
MICAL2
MIR133B
MIR206
MIR503
MLLT6
MOAP1
MOBKL1A
MORC4
MPI
MPP6
MRAP2
MRC2
MT1E
MTSS1
MTSS1L
MUSK
MYBPC2
MYBPH
MYF5
MYH3
MYH8
MYL4
MYL5
MYL6B
MYLPF
MYO18B
MYO5A
MYOG
MYOZ2
NCAPG
NCOA1
NDC80
NDRG2
NDRG4
NEB
NEDD4
NES
NEU1
NIPAL3
NOTCH3
NRG1
NRP2
NTM
NUDT14
NUP93
OLFM2
OLFML2A
OPHN1
OSR2
OTOF
OTUD3
OXTR
P2RX6
PAAF1
PACS2
PADI2
PAN2
PAR5
PAWR
PCBD1
PCK2
PDE2A
PDGFC
PDK1
PDLIM1
PFKM
PGA3
PGA4
PGA5
PGAM2
PGBD5
PGD
PGM2
PGM3
PID1
PLAC9
PLAU
PLEKHA4
PLOD2
PLXNA1
PLXNA2
PODN
PODXL
PORCN
PPFIA4
PRICKLE2
PRKAG3
PRKAR1A
PRKX
PRUNE
PSEN2
PSG4
PTGDS
PTGFR
PTGS1
PTN
PTPRF
PTTG1
PURB
PUS7L
RAB15
RAB3B
RAI14
RALA
RAPH1
RARRES3
RASA1
RASSF4
RBM24
RCL1
RCSD1
RDH5
RGR
RGS16
RGS2
RNASE3
ROS1
RPL39L
RPRD1B
RTN2
RYR1
SARS2
SBK2
SCARNA13
SCD
SCG2
SCGB1C1
SCN4A
SEMA6B
SEPT4
SEPW1
SERPINB8
SERPINE1
SERPINE2
SERPINF1
SETD7
SGOL1
SHD
SHF
SIM1
SIRT2
SIX4
SLC12A2
SLC12A7
SLC29A1
SLC38A3
SLC38A4
SLC38A6
SLC3A2
SLC43A2
SLC7A5
SLC7A7
SMAD1
SMPX
SMURF2
SMYD1
SNAI1
SNED1
SNORD113-4
SNORD116-17
SNORD116-20
SNORD116-3
SOX8
SPARCL1
SPOCK1
SPPL2B
SPRY1
SREBF1
SREBF2
SRF
SRGN
SRPK3
SSPO
SSX4
ST3GAL5
ST6GALNAC4
ST8SIA5
STAC3
STARD5
STAT3
STK10
SUN2
SYNPO
TAS1R1
TBC1D2
TBX15
TCEAL1
TCF12
TCN2
TDRKH
TEAD4
TES
TFCP2
TFF3
TGFBI
TGM2
THBD
THRA
TIMM8A
TIMP3
TMED4
TMEM119
TMEM169
TMEM171
TMEM205
TMEM25
TMEM38A
TMEM8C
TMOD1
TMTC1
TNFRSF11B
TNFRSF25
TNIK
TNNC2
TNNI1
TNNI2
TNNT3
TOP2A
TPCN1
TPM3
TPPP3
TRAPPC2L
TRIB2
TRIM62
TRIO
TSPAN14
TSPAN33
TTC13
TTN
TTYH2
UBE2G2
UBE2W
UCP2
UGDH
UNC119
UNC45B
UTS2R
VANGL1
VAX2
VEGFC
VGLL2
WDR5
WIPF1
WRNIP1
WWTR1
ZBTB7C
ZC3H12A
ZC3H8
ZNF238
ZNF512
ZNF778

It was found 2609 the shortest paths of 4 types of length from 2 to 5 interactions from each miRNA in the hit list from *loss-of-function miRNA screening* to each protein in the list of "final implementers". The subnetwork constructed from these paths consists of 1063 nodes and 2710 edges without duplicated edges. Then the centrality score and the p-value were calculated for each node and path. 240 nodes with centrality score ≥ 3 and with p-value < 0.05 (Table 2-6); 294 paths of length of 3 to 4 interactions, with centrality score ≥ 3 and p-value < 0.05 (Table 2-7) were taken for manual analysis. Analysis of the paths with high centrality scores had highlighted a possible role for a number of nuclear receptors (AR, ARDB1, NR3C1) in skeletal muscle differentiation, as well as suggested functions in myogenesis for such proteins as arrestin (ARRB1 and 2), intersectin (ITSN1), the Rho GTP echange factor VAV3, and the teratocarcinoma-derived growth factor (TDGF1). Interestingly, while the IGF1 regulatory role in myogenesis is very well studied, our approach allowed us to include the arrestin proteins in these pathways, and thus elaborate the known IGF1 network in skeletal muscle differentiation. Obviously, the MEF2D, p300, CCND1 functions in differentiation have been abundantly demonstrated, and their presence among the results serves as a proof of efficiency of the analysis.

Table 2-6. miRNA loss-of-function screening: top 50 nodes with the highest centrality scores.

Node's name	Centrality score	p-value^{Net}
ACTA1	443	0.000
pEGFR	255	0.000
pKIT	231	0.000
pEP300	156	0.000
pESR1	143	0.001
pHNRNPD	141	0.000
pIGF1R	127	0.000
IGF1R	121	0.000
ESR1	119	0.004
pMDM2	117	0.000
pTP53	116	0.002
pTCAP	115	0.000
IGF2	99	0.000
pSP1	99	0.000
pPRKCA	96	0.000
pITGB1	95	0.000
pAP2M1	84	0.001
pAR	79	0.000
pSLC2A4	75	0.001
pARRB1	73	0.000
pJUN	72	0.000
MYH1	69	0.000
pERBB2	66	0.005
ERBB2	66	0.005

Node's name	Centrality score	p-value^{Net}
pE2F1	61	0.000
pCDK2	60	0.002
E2F1	60	0.000
CDK2	59	0.008
TP53	58	0.011
pGRB2	50	0.002
pETS1	49	0.007
ETS1	49	0.007
pYWHAZ	49	0.000
pMAPK1	47	0.050
pNOTCH1	47	0.000
pSMAD4	44	0.048
SMAD4	44	0.047
pMAPK3	43	0.006
MAPK3	43	0.006
pIQGAP1	43	0.000
pIGFBP3	42	0.002
pSHC1	42	0.002
pITGA4	40	0.000
pAKT1	39	0.026
pCD19	39	0.000
EP300	37	0.038
pUBE2I	36	0.043
pSRC	36	0.006

Table 2-7. miRNA loss-of-function screening: top 50 paths with the highest centrality scores.

Centrality score	Path				p-value^{Net}
12	IGF1R-pIGF1R	pEGFR-pIGF1R	pEGFR-pARF6		0.000
12	IGF1R-pIGF1R	pIGF1R-pARRB2	pARF6-pARRB2		0.000
12	IGF1R-pIGF1R	pARRB1-pIGF1R	pARF6-pARRB1		0.000
11	IGF1-pIGF1	pIGF1-pIGFBP3	pIGF2-pIGFBP3		0.000
11	ABL2-pABL2	pABL2-pEGFR	pEGFR-pARF6		0.000
11	IGF1-pIGF1	pIGF1-pIGFBP1	pIGF2-pIGFBP1		0.000
9	IGF1R-pIGF1R	pIGF1R-pMDM2	pTCAP-pMDM2	pTCAP-pMSTN	0.000
9	IGF1R-pIGF1R	pIGF1R-pMDM2	pTCAP-pMDM2		0.000
9	pIGF1R-pMDM2	pTCAP-pMDM2	pTCAP-pMSTN		0.000
8	HIPK2-pHIPK2	pEP300-pHIPK2	pEP300-ACTA1	ACTA1-pACTA1	0.000
8	HIPK2-pHIPK2	pHIPK2-pMDM2	pTCAP-pMDM2	pTCAP-pMSTN	0.000
8	CCND1-pCCND1	pCCND1-pAR	pAR-ACTA1	ACTA1-pACTA1	0.002
8	CCND1-pCCND1	pEP300-pCCND1	pEP300-ACTA1	ACTA1-pACTA1	0.002
8	MEF2D-pMEF2D	pMEF2D-MYH1	MYH1-pMYH1		0.000
8	ADRB1-pADRB1	pADRB1-pARRB1	pARF6-pARRB1		0.000
8	ITSN1-pITSN1	pITSN1-pARFIP2	pARFIP2-pARF6		0.000
8	pHIPK2-pMDM2	pTCAP-pMDM2	pTCAP-pMSTN		0.000
8	HIPK2-pHIPK2	pHIPK2-pMDM2	pTCAP-pMDM2		0.000
8	pEP300-pHIPK2	pEP300-ACTA1	ACTA1-pACTA1		0.000
8	HIPK2-pHIPK2	pEP300-pHIPK2	pEP300-ACTA1		0.000
8	CCND1-pCCND1	pEP300-pCCND1	pEP300-ACTA1		0.002
8	pCCND1-pAR	pAR-ACTA1	ACTA1-pACTA1		0.002
8	CCND1-pCCND1	pCCND1-pAR	pAR-ACTA1		0.002
8	pEP300-pCCND1	pEP300-ACTA1	ACTA1-pACTA1		0.002
7	ACVR2B-pACVR2B	pACVR2B-pACVR1B	pACVR1B-pTDGF1	pTDGF1-pMSTN	0.004
7	SNX6-pSNX6	pSNX6-pEGFR	pEGFR-pARF6		0.000
7	VAV3-pVAV3	pVAV3-pEGFR	pEGFR-pARF6		0.000
7	E2F1-pE2F1	pSTAT1-pE2F1	pKIT-pSTAT1		0.000
7	KPNA3-pKPNA3	pARRB1-pKPNA3	pARF6-pARRB1		0.001
7	KPNA3-pKPNA3	pARRB2-pKPNA3	pARF6-pARRB2		0.001
7	ACVR2B-pACVR2B	pACVR2B-pACVR1B	pACVR1B-pTDGF1		0.004
7	pACVR2B-pACVR1B	pACVR1B-pTDGF1	pTDGF1-pMSTN		0.004
6	E2F1-pE2F1	pE2F1-pSP1	pSP1-ACTA1	ACTA1-pACTA1	0.000
6	E2F1-pE2F1	pMDM2-pE2F1	pTCAP-pMDM2	pTCAP-pMSTN	0.000
6	E2F1-pE2F1	pTBP-pE2F1	pTBP-ACTA1	ACTA1-pACTA1	0.000
6	TP53-pTP53	pTP53-hsa-mir-125b	hsa-mir-125b-IGF2	IGF2-pIGF2	0.000
6	E2F1-pE2F1	pEP300-pE2F1	pEP300-ACTA1	ACTA1-pACTA1	0.000
6	E2F1-pE2F1	pE2F1-hsa-let-7a	hsa-let-7a-IGF2	IGF2-pIGF2	0.000
6	NCOA3-pNCOA3	pNCOA3-pAR	pAR-ACTA1	ACTA1-pACTA1	0.001
6	NCOA3-pNCOA3	pEP300-pNCOA3	pEP300-ACTA1	ACTA1-pACTA1	0.001
6	E2F1-pE2F1	pE2F1-hsa-let-7a	hsa-let-7a-IGF2		0.000
6	pE2F1-hsa-let-7a	hsa-let-7a-IGF2	IGF2-pIGF2		0.000
6	pMDM2-pE2F1	pTCAP-pMDM2	pTCAP-pMSTN		0.000
6	E2F1-pE2F1	pE2F1-ARF6	ARF6-pARF6		0.000
6	TP53-pTP53	pTP53-hsa-mir-125b	hsa-mir-125b-IGF2		0.000
6	pTP53-hsa-mir-125b	hsa-mir-125b-IGF2	IGF2-pIGF2		0.000
6	E2F1-pE2F1	pMDM2-pE2F1	pTCAP-pMDM2		0.000
6	E2F1-pE2F1	pE2F1-pSP1	pSP1-ACTA1		0.000
6	pTBP-pE2F1	pTBP-ACTA1	ACTA1-pACTA1		0.000
6	E2F1-pE2F1	pTBP-pE2F1	pTBP-ACTA1		0.000
6	pE2F1-pSP1	pSP1-ACTA1	ACTA1-pACTA1		0.000

It was found 47714 the shortest paths of 4 types and of length from 1 to 5 interactions from each gene in the hit list from *transcriptome profiling* to each protein in the list of "final implementers". The subnetwork constructed from these paths consists of 2847 nodes (303 of which are genes) and 13032 edges without duplicated edges. The centrality score and the p-value were calculated for each node and each path in the subnetwork. 328 nodes with centrality score ≥ 3 and p-value < 0.05 (Table 2-8 A); 1623 paths of length of 3 to 4 interactions and centrality score ≥ 3, with p-value < 0.05 (Table 2-9) were taken for further analysis. The scores^{exp} and p-values^{exp} were also calculated for each of these nodes/paths. There are 18 miRNAs among these 328 nodes (Table 2-8 B). 4 of them (hsa-mir-125b, has-mir-133a, has-mir-133b, has-mir-145) are the hit miRNAs in the loss-of-function screen. 5 with the highest centrality score are hsa-mir-125b, hsa-mir-29a, hsa-mir-371, hsa-mir-216a, hsa-mir-1. All these miRNAs except hsa-mir-371 were shown to be involved in muscle differentiation and/or proliferation previously (Callis et al., 2007; Crist et al., 2009; Meyer et al., 2015; Winbanks et al., 2011). Moreover, almost half of these miRNAs are known to participate in terminal muscle differentiation, and potential roles in myogenesis could be predicted for other miRNAs in this list e.g that regulate cellular proliferation (such as miR-132, miR-145 or miR-224), as well as cardiac hypertrophy (miR-378). Interestingly, the majority of these miRNAs were not found in the original loss-of-function screen, most likely due to the redundancy of miRNA family members. Indeed, as the miRNAs of the same family share the seed sequence, an efficient loss-of-function screen should have contained not only individual miRNA inhibitors, but also the inactivators of whole miRNA families, in order to avoid false negative results. In this sense, our analysis of these data have been very important in supplementing a group of miRNA targets that could have been overlooked. This possibility is highlighted by the presence of known myogenesis regulatory miRNAs (miR-1, miR-29a, miR-216a) in the list resulting from the analysis, whereas they have not been picked up by the original experimental screen.

The analysis of paths allowed to identify a potentially novel and interesting pathway in regulation of myogenesis, involving the Myc-associated factor X (MAX), the clathrin-coated pathway regulatory protein AP2M1, and the EH-domain protein EHD2, which links the clathrin coated transport to actin cytoskeleton, and also binds to myoferlin, a factor

promoting myotube fusion (Figure 2-14). Together with integrin subunits ITGA4 and ITGB1, the extracellular matrix component fibronectin (FN1), and the protein chaperon HSP90, these proteins indicate a possible involvement of specific protein transport pathways in terminal myogenic differentiation. In addition, there is a possibility of involvement of beta-catenin (CTNNB1), C-KIT and PRKC in these processes. It should be noted that these three regulatory factors, while extensively studied in a multitude of biological models, have never been shown to be specifically implicated in skeletal myogenesis. Two major regulatory molecules of skeletal myogenesis, MYOD and SMAD3, have been highlighted, together with their previously known muscle-related targets (TGFB1, CDC42, CTCF). Also, they are linked to a number of proteins that have not been previously studied in the context of muscle differentiation (MAX, KIT, AP2M1…).

Figure 2-14. Transcriptome profiling: MAX - AP2M1 - EHD2 path.

'P' denotes a protein. Nodes in yellow are hit genes' products. Node in green is 'final implementer'.

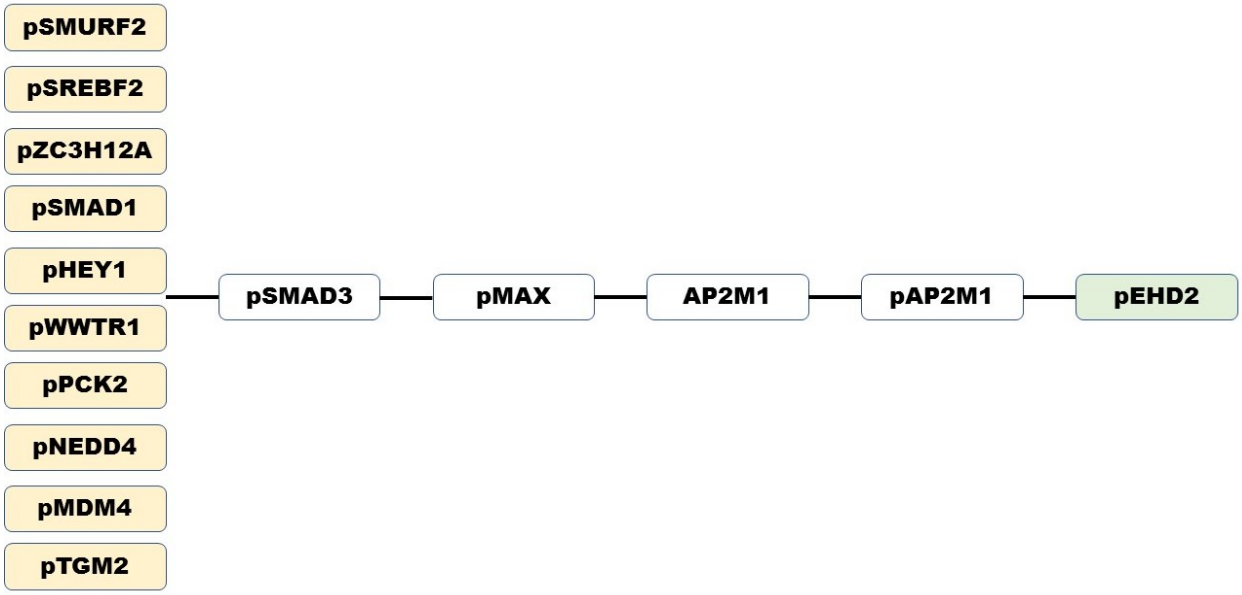


Table 2-8. Transcriptome profiling: top 50 nodes with the highest centrality scores and miRNAs.

(A) Top 50 nodes with the highest centrality scores.

Node's name	Centrality score	p-value^{Net}	Score^{Exp}	p-value^{Exp}
CDC42	3641	0.015	1.170	0.300
pSP1	3479	0.019	1.170	0.457
pEGFR	3469	0.009	1.130	0.841
pTP53	3418	0.023	1.330	0.238
pAR	2998	0.008	1.490	0.065
MYH1	2666	0.011	1.100	0.090
pITGB1	2330	0.000	1.400	0.029
pKIT	2243	0.002	1.000	1.000
pHNRNPD	1933	0.040	1.020	0.789
pCTNNB1	1733	0.000	1.100	0.587
CD81	1420	0.036	1.520	0.014
pHSP90AA1	1414	0.017	1.040	0.678
pSRC	1245	0.013	1.130	0.046
pIGFBP3	1150	0.000	1.770	0.005
pSMAD3	1060	0.000	1.290	0.454
pJUN	1023	0.034	1.140	0.223
pNOV	911	0.009	1.140	0.099
pNOTCH1	812	0.002	2.800	0.021
pTF	764	0.013	1.030	0.551
pFN1	753	0.010	1.340	0.000
pPOU2F1	743	0.020	1.050	0.695
pRXRA	687	0.021	1.320	0.164
pMEF2C	680	0.008	8.180	0.034
ITGB1	671	0.000	1.400	0.029
hsa-mir-125b	606	0.009	1.170	0.124
pPIK3R1	576	0.038	1.070	0.514
pSTAT3	554	0.018	2.420	0.001
pBTRC	553	0.009	1.220	0.322

Node's name	Centrality score	p-value^{Net}	Score^{Exp}	p-value^{Exp}
pBTRC	553	0.009	1.220	0.322
hsa-mir-29a	551	0.018	1.110	0.505
pABL1	549	0.017	1.340	0.016
pMEF2D	519	0.042	2.200	0.002
pIQGAP1	515	0.038	1.910	0.000
pTCF3	503	0.010	1.200	0.430
pFLNA	494	0.042	1.450	0.017
pFHL2	492	0.003	1.960	0.076
pCD19	490	0.002	1.080	0.555
pMYOD1	467	0.006	2.940	0.103
pRB1	466	0.035	1.350	0.030
EGFR	426	0.025	1.130	0.841
pFRA11B	422	0.002	-	-
pVTN	399	0.008	1.010	0.931
pRAC1	370	0.006	1.300	0.292
pPRKAA1	344	0.000	1.100	0.070
pTRIM63	340	0.037	1.020	0.870
pFST	327	0.000	1.810	0.173
pAURKB	321	0.007	2.420	0.049
hsa-mir-371	318	0.000	-	-
pTDGF1	312	0.003	1.000	0.990
FST	294	0.000	1.810	0.173
pIGFBP5	283	0.044	1.750	0.232
pACTN1	276	0.044	1.430	0.012
pGATA2	271	0.018	1.210	0.155
TDGF1	260	0.000	1.000	0.990
pUSP15	258	0.015	1.300	0.209
pTCF4	258	0.032	1.040	0.873

(B) List of nodes that are miRNAs.

Node's name	Centrality score	p-value^{Net}	Score^{Exp}	p-value^{Exp}
hsa-mir-125b	606	0.009	1.170	0.124
hsa-mir-29a	551	0.018	1.110	0.505
hsa-mir-371	318	0.000	-	-
hsa-mir-216a	206	0.001	1.270	0.122
hsa-mir-1	148	0.000	-	-
hsa-mir-224	109	0.003	1.020	0.728
hsa-mir-378	44	0.000	-	-
hsa-mir-1-1	39	0.002	1.050	0.679
hsa-mir-145	29	0.003	1.040	0.656

Node's name	Centrality score	p-value^{Net}	Score^{Exp}	p-value^{Exp}
hsa-mir-206	16	0.010	7.550	0.000
hsa-mir-449b	13	0.049	-	-
hsa-mir-25	10	0.036	1.080	0.883
hsa-mir-1-2	7	0.044	1.020	0.787
hsa-mir-132	6	0.013	1.090	0.477
hsa-mir-34b	6	0.046	1.060	0.411
hsa-mir-133a	4	0.010	1.110	0.083
hsa-mir-133b	3	0.047	14.700	0.000
hsa-mir-181a-2	3	0.010	1.180	0.328

Table 2-9. Transcriptome profiling: top 50 paths with the highest centrality scores.

Centrality count	Path				p-value^{Net}	Score^{Exp}	p-value^{Exp}
18	pMYOD1-hsa-mir-1	hsa-mir-1-TSPAN4	TSPAN4-pTSPAN4		0.000	0.812	0.557
18	pMYOD1-hsa-mir-1-1	hsa-mir-1-1-KIT	KIT-pKIT		0.000	0.657	0.616
13	pFHL2-pAR	pAR-ACTA1	ACTA1-pACTA1		0.049	14.770	0.001
11	pSMAD3-pCTCF	pCTCF-AP2M1	AP2M1-pAP2M1		0.035	1.358	0.117
11	pSMAD3-pCTCF	pCTCF-AP2M1	AP2M1-pAP2M1	pEHD2-pAP2M1	0.035	1.363	0.104
11	pSMAD3-pCTNNB1	pCTNNB1-FST	FST-pFST		0.046	1.412	0.882
11	pSMAD3-pCTNNB1	pCTNNB1-TDGF1	TDGF1-pTDGF1		0.046	1.615	0.671
11	pSMAD3-pCTNNB1	pCTNNB1-TDGF1	TDGF1-pTDGF1	pTDGF1-pMSTN	0.046	1.400	0.744
11	pSMAD3-pCTNNB1	pCTNNB1-FST	FST-pFST	pFST-pMSTN	0.046	1.130	0.959
11	pSMAD3-pMAX	pMAX-AP2M1	AP2M1-pAP2M1		0.035	1.310	0.195
11	pSMAD3-pMAX	pMAX-AP2M1	AP2M1-pAP2M1	pEHD2-pAP2M1	0.035	1.300	0.185
10	pSMAD3-hsa-mir-145	hsa-mir-145-TRIP10	TRIP10-pTRIP10		0.024	0.645	0.333
10	pSMAD3-hsa-mir-145	hsa-mir-145-TRIP10	TRIP10-pTRIP10	pTRIP10-pCDC42	0.024	0.540	0.251
10	pSMAD3-hsa-mir-200b	hsa-mir-200b-ERRFI1	ERRFI1-pERRFI1		0.027	0.707	0.272
10	pSMAD3-hsa-mir-200b	hsa-mir-200b-ERRFI1	ERRFI1-pERRFI1	pERRFI1-pCDC42	0.024	0.614	0.189
10	pSMAD3-pAR	pAR-hsa-mir-216a	hsa-mir-216a-CDC42		0.024	0.777	0.336
10	pSMAD3-pAR	pAR-hsa-mir-216a	hsa-mir-216a-CDC42	CDC42-pCDC42	0.024	0.777	0.336
10	pSMAD3-pCEBPA	pCEBPA-hsa-mir-29a	hsa-mir-29a-CDC42		0.024	0.657	0.880
10	pSMAD3-pCEBPA	pCEBPA-hsa-mir-29b	hsa-mir-29b-CDC42		0.024	0.657	0.880
10	pSMAD3-pCEBPA	pCEBPA-hsa-mir-29b	hsa-mir-29b-CDC42	CDC42-pCDC42	0.024	0.657	0.880
10	pSMAD3-pCEBPA	pCEBPA-hsa-mir-29a	hsa-mir-29a-CDC42	CDC42-pCDC42	0.024	0.657	0.880
10	pSMAD3-pJUN	pJUN-hsa-mir-29a	hsa-mir-29a-CDC42		0.024	0.677	0.611
10	pSMAD3-pJUN	pJUN-hsa-mir-29a	hsa-mir-29a-CDC42	CDC42-pCDC42	0.024	0.677	0.611
10	pSMAD3-pTGFB1	pTGFB1-hsa-mir-29a	hsa-mir-29a-CDC42		0.024	0.711	0.557
10	pSMAD3-pTGFB1	pTGFB1-hsa-mir-224	hsa-mir-224-CDC42		0.024	0.711	0.557
10	pSMAD3-pTGFB1	pTGFB1-hsa-mir-224	hsa-mir-224-CDC42	CDC42-pCDC42	0.024	0.711	0.557
10	pSMAD3-pTGFB1	pTGFB1-hsa-mir-29a	hsa-mir-29a-CDC42	CDC42-pCDC42	0.024	0.711	0.557
10	pSMAD3-pYY1	pYY1-hsa-mir-29b	hsa-mir-29b-CDC42		0.024	0.660	0.766
10	pSMAD3-pYY1	pYY1-hsa-mir-29a	hsa-mir-29a-CDC42		0.024	0.660	0.766
10	pSMAD3-pYY1	pYY1-hsa-mir-29c	hsa-mir-29c-CDC42		0.024	0.660	0.766
10	pSMAD3-pYY1	pYY1-hsa-mir-29b	hsa-mir-29b-CDC42	CDC42-pCDC42	0.024	0.660	0.766
10	pSMAD3-pYY1	pYY1-hsa-mir-29a	hsa-mir-29a-CDC42	CDC42-pCDC42	0.024	0.660	0.766
10	pSMAD3-pYY1	pYY1-hsa-mir-29c	hsa-mir-29c-CDC42	CDC42-pCDC42	0.024	0.660	0.766
9	hsa-mir-1-1-KIT	KIT-pKIT	pCD81-pKIT		0.001	1.260	0.292
9	hsa-mir-1-1-KIT	KIT-pKIT	pKIT-pCD9		0.001	1.095	0.765
9	hsa-mir-378-ITGA4	ITGA4-pITGA4	pCD81-pITGA4		0.000	1.320	0.242
9	pELAVL1-pTCF3	pELAVL1-pPRKCA	pPRKCA-pCD9		0.006	1.165	0.497
9	pMYOD1-hsa-mir-1	hsa-mir-1-TSPAN4	TSPAN4-pTSPAN4	pCD81-pTSPAN4	0.000	0.930	0.227
9	pMYOD1-hsa-mir-1	hsa-mir-1-TSPAN4	TSPAN4-pTSPAN4	pTSPAN4-pCD9	0.000	0.875	0.543
9	pMYOD1-hsa-mir-1-1	hsa-mir-1-1-KIT	KIT-pKIT	pCD81-pKIT	0.000	0.780	0.260
9	pMYOD1-hsa-mir-1-1	hsa-mir-1-1-KIT	KIT-pKIT	pKIT-pCD9	0.000	0.733	0.624
9	pMYOD1-hsa-mir-378	hsa-mir-378-ITGA4	ITGA4-pITGA4		0.000	0.930	0.225
9	pMYOD1-hsa-mir-378	hsa-mir-378-ITGA4	ITGA4-pITGA4	pCD81-pITGA4	0.000	0.812	0.575
9	pSMAD3-pAKT1	pAKT1-hsa-mir-125b	hsa-mir-125b-IGF2		0.045	0.721	0.421
9	pSMAD3-pAKT1	pAKT1-hsa-mir-125b	hsa-mir-125b-IGF2	IGF2-pIGF2	0.045	0.721	0.421
9	pTP53-pSMAD3	pTP53-hsa-mir-125b	hsa-mir-125b-IGF2		0.045	0.784	0.269
9	pTP53-pSMAD3	pTP53-hsa-mir-125b	hsa-mir-125b-IGF2	IGF2-pIGF2	0.045	0.784	0.269
8	pBTRC-pCTNNB1	pCTNNB1-TDGF1	TDGF1-pTDGF1		0.016	1.597	0.643
8	pBTRC-pCTNNB1	pCTNNB1-FST	FST-pFST		0.016	1.395	0.836
8	pBTRC-pCTNNB1	pCTNNB1-FST	FST-pFST	pFST-pMSTN	0.016	1.107	0.926
8	pBTRC-pCTNNB1	pCTNNB1-TDGF1	TDGF1-pTDGF1	pTDGF1-pMSTN	0.016	1.377	0.680

Next, the lists of nodes and the lists of paths found for loss-of-function miRNA screening and transcriptomic profiling were compared. It was found 29 nodes (Table 2-10) and 16 paths (Table 2-11) common for both for transcriptomic profiling and loss-of-function screening. 3 out of 5 nodes with the highest centrality score - IGF1R, CBL, E2F1 - have been suggested to play key roles in the growth, development, and differentiation of skeletal muscle (Fernández et al., 2002; Nakao et al., 2009; Zappia et al., 2016). Most of the paths pass through nodes with high centrality score (E2F2, KIT, MDM2, CD81, HNRNPD, IGF1R, ESR1). One of the paths with the highest score consists of CDKN1A, MDM2, TCAP, MSTN proteins. The interaction between MDM2 and TCAP is known to be important for cardiac hypertrophy (Tian et al., 2006), it was also shown that TCAP controls secretion of MSTN(Nicholas et al., 2002). The analysis shows that this path might be activated by the depletion of hsa-mir-17, hsa-mir-106a, hsa-mir-125a, hsa-mir-145, hsa-mir-93 and/or by interaction with differentially expressed genes products (Figure 2-15). I can also be noted, that not only androgen receptor (AR), but also the estrogen receptor ESR1 can play a role in human skeletal myogenesis. Interestingly, specific integrins (ITGB1, ITGA6) and adaptor proteins (CRKL) have also been found, confirming the importance of certain membrane/adherence structures. Strikingly, both the receptor of activated C kinase (RACK1), and the inhibitor of this kinase (YWHAB, a 14-3-3 protein), as well as multiple other protein-processing enzymes (peptidase inhibitor SERPIN1, casein kinase CSNK1A1, proprotein convertase FURIN, and activator or protein secretion CHRM3) were found by the analysis, attracting the attention to the role of protein metabolism in myogenesis. It was also very interesting to see the chromosome breakpoint generation factor FRA11B among these potential novel factors that might impact on the differentiation of human myoblasts. This comparison has shown potentially novel paths originating from well-known actors in muscle differentiation (such as IGF1R - RACK1 - CD81); and vice versa, has shown that previously unknown potential regulators of myogenesis, such as YWHAB or FRA11B, can act upon proteins that are well known to regulate myotube hypertrophy and/or fusion (IGFR1, CD81).

The fact that the comparison resulted only in a few number of paths might indicate, that although these two experimental systems study one biological process, they interfere biological machinery on two different levels – on the level of translation (miRNAs) and on

the level on transcription (transcriptome), which might account on two different regulation mechanisms.

Table 2-10. Common nodes for miRNA loss-of-function screening and transcriptome profiling.

(TF – transcriptome profiling; miRNA LOF – miRNA loss-of-function screening)

Centrality score TF	Centrality score miRNA LOF	Node's name
151	127	pIGF1R
192	28	pCBL
56	119	ESR1
31	121	IGF1R
61	60	E2F1
89	27	pACVR1B
69	14	ERRFI1
68	10	pIGF2
57	16	CDKN1A
42	4	pSOS1
37	9	CLTC
10	33	FSCN1
35	3	MET
33	5	AR
29	7	pITGA6
29	3	pFURIN
28	3	PAK5
22	6	pSERPINE1

Centrality score TF	Centrality score miRNA LOF	Node's name
20	8	TGFBR1
5	17	HNF4A
16	5	pBNIPL
10	10	pIGF2R
8	7	pACVR2B
10	3	pRARG
8	5	YWHAQ
10	2	pITGB4
5	7	SOCS6
6	5	pBIRC6
8	2	ITGB4
7	3	FURIN
3	6	CSNK1A1
2	7	SNX6
4	2	PNP
2	4	KRT7
2	4	SMURF1

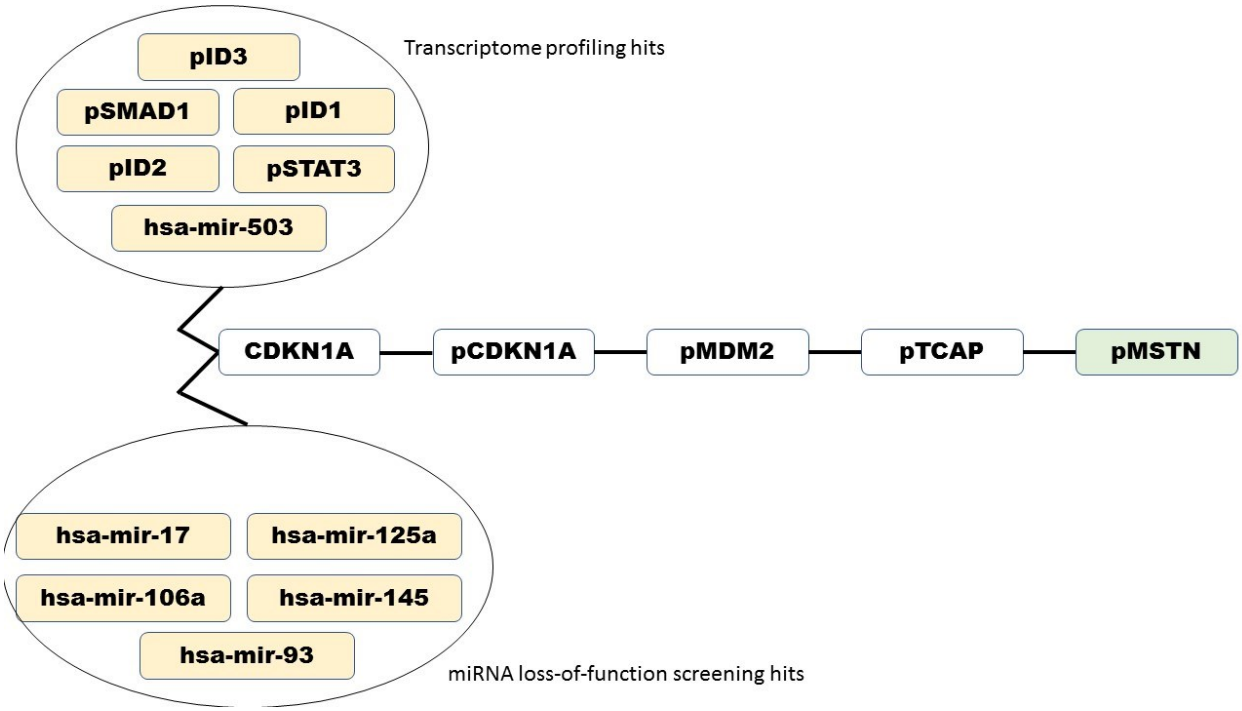
Table 2-11. Common paths for loss-of-function screening and transcriptome profiling.

(TF – transcriptome profiling; miRNA LOF – miRNA loss-of-function screening.)

Length	Centrality score TP	Centrality score miRNA LOF	Path			
3	3	7	E2F1-pE2F1	pSTAT1-pE2F1	pKIT-pSTAT1	
4	3	6	E2F1-pE2F1	pMDM2-pE2F1	pTCAP-pMDM2	pTCAP-pMSTN
3	3	6	E2F1-pE2F1	pMDM2-pE2F1	pTCAP-pMDM2	
4	6	5	CDKN1A-pCDKN1A	pCDKN1A-pMDM2	pTCAP-pMDM2	pTCAP-pMSTN
3	6	5	CDKN1A-pCDKN1A	pCDKN1A-pMDM2	pTCAP-pMDM2	
3	3	4	pIGF1R-pRASA1	pRASA1-pKIT	pCD81-pKIT	
3	3	4	pRACK1-pIGF1R	pRACK1-pITGB1	pCD81-pITGB1	
3	3	4	pIGF1R-pMDM2	pMDM2-pHNRNPD	pCD81-pHNRNPD	
3	3	4	pYWHAB-pIGF1R	pYWHAB-pITGB1	pCD81-pITGB1	
3	3	4	pFRA11B-pIGF1R	pKIT-pFRA11B	pCD81-pKIT	
3	3	4	pIGF1R-pCRK	pCRK-pKIT	pCD81-pKIT	
3	3	4	pIGF1R-pPIK3R2	pPIK3R2-pKIT	pCD81-pKIT	
3	3	4	pPTPN11-pIGF1R	pPTPN11-pKIT	pCD81-pKIT	
3	3	4	pIGF1R-pESR1	pESR1-pHNRNPD	pCD81-pHNRNPD	
3	3	4	pCRKL-pIGF1R	pCRKL-pKIT	pCD81-pKIT	
4	2	4	E2F1-pE2F1	pE2F1-pBRCA1	pHNRNPD-pBRCA1	pCD81-pHNRNPD
3	3	3	CSNK1A1-pCSNK1A1	pCSNK1A1-pCHRM3	pARF6-pCHRM3	

Figure 2-15. Comparison: CDKN1A-MDM2-TCAP-MSTN path.

'P' denotes a protein. Nodes in yellow are hit genes' products; node in green is 'final implementer'.



We also found that 14 paths from the analysis of transcriptomic profiling have miRNAs hits from the loss-of-function screening (Table 2-12) and 60 paths from the analysis of loss-of-function screening have hit genes from the transcriptomic profiling on them (Table 2-13). These are the paths from the analysis of transcriptomic profiling that pass through hsa-mir-125b which controls IGF2 gene and the paths that pass through has-mir-145 that control TRIP10 protein which (Figure 2-16 A), according to OMIM database, has highest expression levels in skeletal muscle (McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, n.d.) and interacts with CDC42 protein. Also, analyzing these paths one can notice the factors participating in at least three major cellular pathways, that, however, have not been extensively studied in skeletal muscle differentiation. These factors include beta-transduction repeat containing protein (BTRC), which has a strong impact on both beta-catenin and NF-kappa B signaling (Figure 2-16 B), as well as the p53-related

protein TP73 (Figure 2-16 A), and, finally, the protein LRIG1 (Figure 2-16 C) that has a strong negative effect on the expression of epidermal growth factor receptor. These pathways represent interesting new directions to follow, in order to further understand the mechanics of skeletal myogenesis.

Table 2-12. Transcriptome profiling paths that pass through miRNA hits.

Length	Centrality score	Path			
4	10	pSMAD3-hsa-mir-145	hsa-mir-145-TRIP10	TRIP10-pTRIP10	pTRIP10-pCDC42
3	10	pSMAD3-hsa-mir-145	hsa-mir-145-TRIP10	TRIP10-pTRIP10	
4	9	pSMAD3-pAKT1	pAKT1-hsa-mir-125b	hsa-mir-125b-IGF2	IGF2-pIGF2
4	9	pTP53-pSMAD3	pTP53-hsa-mir-125b	hsa-mir-125b-IGF2	IGF2-pIGF2
3	9	pSMAD3-pAKT1	pAKT1-hsa-mir-125b	hsa-mir-125b-IGF2	
3	9	pTP53-pSMAD3	pTP53-hsa-mir-125b	hsa-mir-125b-IGF2	
4	7	pMYOD1-pSTAT3	pSTAT3-hsa-mir-125b	hsa-mir-125b-IGF2	IGF2-pIGF2
4	7	pTP53-pBTRC	pTP53-hsa-mir-125b	hsa-mir-125b-IGF2	IGF2-pIGF2
4	7	pNFKB1-pBTRC	pNFKB1-hsa-mir-125b	hsa-mir-125b-IGF2	IGF2-pIGF2
3	7	pTP53-pBTRC	pTP53-hsa-mir-125b	hsa-mir-125b-IGF2	
3	7	pMYOD1-pSTAT3	pSTAT3-hsa-mir-125b	hsa-mir-125b-IGF2	
3	7	pNFKB1-pBTRC	pNFKB1-hsa-mir-125b	hsa-mir-125b-IGF2	
4	3	pTP73-hsa-mir-145	hsa-mir-145-TRIP10	TRIP10-pTRIP10	pTRIP10-pCDC42
3	3	pTP73-hsa-mir-145	hsa-mir-145-TRIP10	TRIP10-pTRIP10	
3	3	hsa-mir-133b-IGF1R	IGF1R-pIGF1R	pFRA11B-pIGF1R	

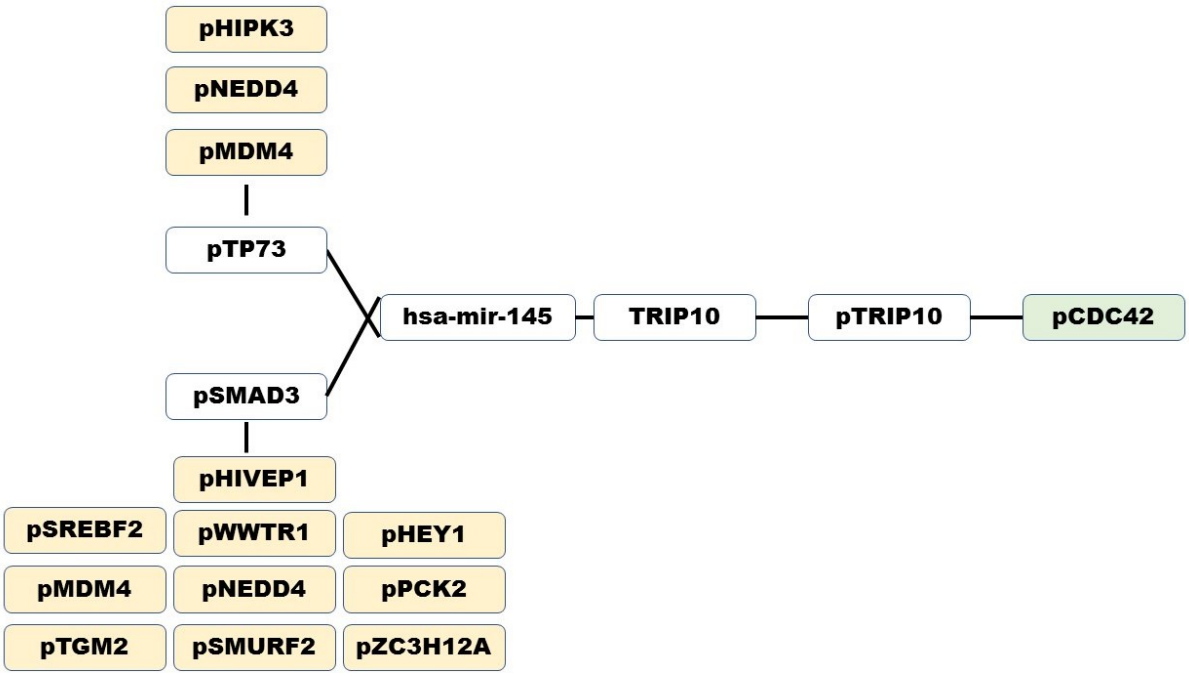
Table 2-13. Comparison: top 50 miRNA loss-of-function screening paths that pass through transcriptome profiling hits.

Length	Centrality score	Path			
3	11	IGF1-pIGF1	pIGF1-pIGFBP3	pIGF2-pIGFBP3	
3	11	IGF1-pIGF1	pIGF1-pIGFBP1	pIGF2-pIGFBP1	
3	8	MEF2D-pMEF2D	pMEF2D-MYH1	MYH1-pMYH1	
3	8	pEP300-pHIPK2	pEP300-ACTA1	ACTA1-pACTA1	
3	8	pEP300-pCCND1	pEP300-ACTA1	ACTA1-pACTA1	
3	8	pCCND1-pAR	pAR-ACTA1	ACTA1-pACTA1	
4	6	E2F1-pE2F1	pE2F1-pSP1	pSP1-ACTA1	ACTA1-pACTA1
4	6	E2F1-pE2F1	pTBP-pE2F1	pTBP-ACTA1	ACTA1-pACTA1
4	6	E2F1-pE2F1	pEP300-pE2F1	pEP300-ACTA1	ACTA1-pACTA1
3	6	pE2F1-pSP1	pSP1-ACTA1	ACTA1-pACTA1	
3	6	pTBP-pE2F1	pTBP-ACTA1	ACTA1-pACTA1	
3	6	pEP300-pE2F1	pEP300-ACTA1	ACTA1-pACTA1	
3	6	pNCOA3-pAR	pAR-ACTA1	ACTA1-pACTA1	
3	6	LRIG1-pLRIG1	pLRIG1-pEGFR	pEGFR-pARF6	
3	6	pEP300-pNCOA3	pEP300-ACTA1	ACTA1-pACTA1	
4	5	NOTCH1-pNOTCH1	pNOTCH1-pEP300	pEP300-ACTA1	ACTA1-pACTA1
3	5	pFOS-pRPS6KA2	pFOS-ACTA1	ACTA1-pACTA1	
3	5	pFOXO3-pEP300	pEP300-ACTA1	ACTA1-pACTA1	
3	5	pSMAD2-pJUN	pJUN-ACTA1	ACTA1-pACTA1	
3	5	pJUN-pRPS6KA2	pJUN-ACTA1	ACTA1-pACTA1	
3	5	pSMAD2-pSP1	pSP1-ACTA1	ACTA1-pACTA1	
3	5	pNRIP1-pJUN	pJUN-ACTA1	ACTA1-pACTA1	
3	5	pEP300-pSMAD2	pEP300-ACTA1	ACTA1-pACTA1	
3	5	pMEF2D-pSP1	pSP1-ACTA1	ACTA1-pACTA1	
3	5	pRPS6KA3-pYBX1	pYBX1-ACTA1	ACTA1-pACTA1	
3	5	pNRIP1-pCTBP2	pCTBP2-ACTA1	ACTA1-pACTA1	
3	5	pNOTCH1-pEP300	pEP300-ACTA1	ACTA1-pACTA1	
3	4	pHNF4A-pSP1	pSP1-ACTA1	ACTA1-pACTA1	
3	4	NEDD4-pNEDD4	pARRB2-pNEDD4	pARF6-pARRB2	
3	4	pJUN-pETS1	pJUN-ACTA1	ACTA1-pACTA1	
3	4	NEDD4-pNEDD4	pARRB1-pNEDD4	pARF6-pARRB1	
3	4	pRANBP9-pAR	pAR-ACTA1	ACTA1-pACTA1	
3	4	pEP300-pWDR82	pEP300-ACTA1	ACTA1-pACTA1	
3	4	GAB2-pGAB2	pGAB2-pEGFR	pEGFR-pARF6	
3	4	pIGF1R-pRASA1	pRASA1-pKIT	pCD81-pKIT	
3	4	pNR2C2-pAR	pAR-ACTA1	ACTA1-pACTA1	
4	3	PTEN-pPTEN	pPTEN-hsa-mir-302d	hsa-mir-302d-ACTA1	ACTA1-pACTA1
4	3	NR3C1-pNR3C1	pJUN-pNR3C1	pJUN-ACTA1	ACTA1-pACTA1
4	3	MAP3K5-pMAP3K5	pEP300-pMAP3K5	pEP300-ACTA1	ACTA1-pACTA1
4	3	MYC-pMYC	pMYC-pSP1	pSP1-ACTA1	ACTA1-pACTA1
4	3	MYC-pMYC	pMYC-pTBP	pTBP-ACTA1	ACTA1-pACTA1
4	3	ABL1-pABL1	pABL1-pJUN	pJUN-ACTA1	ACTA1-pACTA1
4	3	PTEN-pPTEN	pAR-pPTEN	pAR-ACTA1	ACTA1-pACTA1
4	3	NR3C1-pNR3C1	pTBP-pNR3C1	pTBP-ACTA1	ACTA1-pACTA1
4	3	NR3C1-pNR3C1	pNR3C1-pEP300	pEP300-ACTA1	ACTA1-pACTA1
4	3	MYC-pMYC	pMYC-pEP300	pEP300-ACTA1	ACTA1-pACTA1
4	3	NR3C1-pNR3C1	pAR-pNR3C1	pAR-ACTA1	ACTA1-pACTA1
4	3	VAV2-pVAV2	pTTN-pVAV2	pTTN-pTCAP	pTCAP-pMSTN
4	3	MYC-pMYC	pMYC-pJUN	pJUN-ACTA1	ACTA1-pACTA1
4	3	NEDD4-pNEDD4	pMDM2-pNEDD4	pTCAP-pMDM2	pTCAP-pMSTN

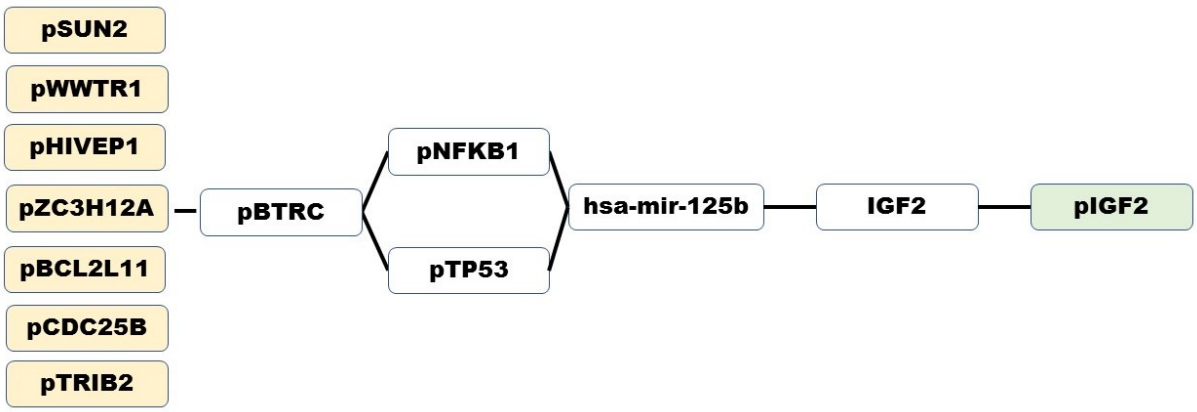
Figure 2-16. Three paths for muscle differentiation process.

'p' denotes a protein. Nodes in yellow are hit genes' products; node in green is 'final implementer'.

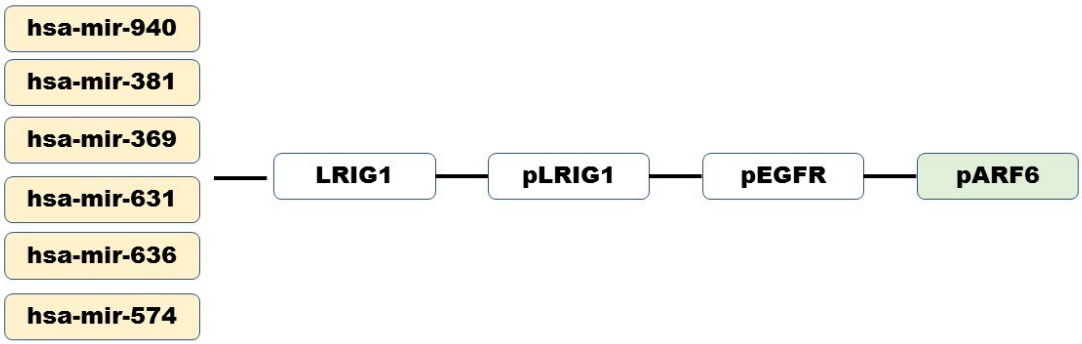
A. has-mir-145-TRIP10-CDC42 path



B. pBTRC-pTP53/pNFKB1-hsa-mir-125b-IGF2



C. LRIG1-EGFR-pARF6 path



Application: human DNA repair process RNAi loss-of-function screening

The data from the siRNA functional screening that identified 'druggable' genes involved in oxidative damaged DNA repair (Guyon et al., 2015; Robinson et al., 2017) were used as the hit list. The screening was performed on genetically engineered HeLa cells that express OGG1-GFP fusion protein. OGG1 is a DNA glycosylate protein that is recruited to chromatin to initiate the repair of oxidized chromatin (Guyon et al., 2015). Each 'druggable' gene was targeted by 3 siRNAs (Robinson et al., 2017). The intensity of chromatin-bound OGG1-GFP was detected after inducing DNA damage and 18 hit genes were identified (Table 2-14).

Table 2-14. DNA repair loss-of-function screening: hit list.

CHD4	MED14	RNF111
DDB1	PSMA1	SETD1B
DNM1	PSMA3	SMC1A
FANCA	PSMA4	SMC3
KIFC3	RAD21	SMURF1
MED12	RBX1	UBE2B

All 18 hit genes were used as a list of "final implementers". The analysis was performed in the protein-protein human interactome.

It was found 4876 the shortest protein-protein paths of length from 1 to 4 interactions from each gene product in the hit list to each gene product in the list of "final implementers". The subnetwork constructed from these paths consists of 381 nodes and 1764 edges without duplicated edges. The centrality score and the p-value were calculated for each node and path in the subnetwork according the procedure described in the masterPATH algorithm

section. 28 nodes with centrality score ≥ 3 and with p-value < 0.05 (Table 2-15) and 64 paths of length of 2 to 3 interactions, with centrality score ≥ 3 and p-value < 0.05 (Table 2-16) were found.

The top 10 nodes with the highest centrality score are histones 3H proteins: HIST1H3A, HIST1H3B, HIST1H3C, HIST1H3D, HIST1H3E, HIST1H3F, HIST1H3G, HIST1H3H, HIST1H3I, HIST1H3J. It is known that the DNA damage is associated with higher level of chromatin mobility (Dion et al., 2012; Miné-Hattab et al., 2012; Strzyz, 2017) and it was shown recently that the increase in chromatin mobility is governed by the proteasome-mediated degradation of core histones(Hauer et al., 2017). Other proteins with high centrality score are SETDB1 protein – a member of the SET1 family of proteins; WDR5 protein - a core component of SET1 family complexes (Ruthenburg et al., 2016); TP53BP1 - a binding partner of the tumour suppressor protein p53. SETDB1 and WDR5 are associated with post-translational histone modification which allows recruitment of the chromatin-associated proteins and protein complexes(Odho et al., 2010; Schultz et al., 2002). TP53BP1 protein is known to be an important regulator of the cellular response to DNA double-strand breaks (Panier et al., 2013).

Figure 3-4 presents subnetworks visualized with Cytoscape software (Shannon et al., 2003) for the paths of length 3 interactions with centrality score 3. Figure 2-17a shows that the method identified that two cohesin proteins SMC3 and SMC1A can act by interacting with RAD21 protein, known cohesin-RAD21 complex (Losada, 2014) enriched at DNA double-strand break sites and facilitates recombinational DNA repair (Watrin et al., 2009). Figure 2-17b shows possible mechanism of involvement of PSMA1, PSMA3, PSMA4 proteins, all members of the 20S proteasome (Coux et al., 1996), through interaction with AURKB, Aurora Kinase B (Shu et al., 2003), which in turn interacts with histones H3 (Crosio et al., 2002). The path ends with histones H3 – SETDB1 interaction. SETDB1 is a histone methyltransferase that specifically methylates histone H3 (Schultz et al., 2002) and is also a member of the hit list. The red arrow shows the direction of this interaction on Figure 2-17a. Considering this, histones H3 are the proteins where the signal converges from different members of the hit list and we hypothesize that histone H3 can be a “final implementer” for this system.

Table 2-15. DNA repair loss-of-function screening: nodes with the highest centrality scores.

Node's name	Centrality score	pvalue
pHIST1H3B	324	0.0462
pHIST1H3F	324	0.0462
pHIST1H3A	324	0.0463
pHIST1H3D	324	0.0463
pHIST1H3C	324	0.0463
pHIST1H3G	324	0.0467
pHIST1H3J	324	0.0469
pHIST1H3E	324	0.0472
pHIST1H3H	324	0.0473
pHIST1H3I	324	0.0476
pMCM6	318	0.0059
pWDR5	148	0.0409
pTP53BP1	90	0.0187
pAURKB	85	0.0409

Node's name	Centrality score	pvalue
pSETDB1	80	0.0291
pRBBP7	75	0.0349
pCXXC1	60	0.0448
pATM	52	0.0367
pCHAF1A	50	0.0342
pING2	44	0.0317
pMDC1	32	0.0405
pHIST2H2AC	30	0.0401
pRFC1	29	0.0156
pMSH6	28	0.0308
pMXD1	23	0.0176
pPOLA1	16	0.0474
pNDC80	12	0.0492
pKMT2B	6	0.0344

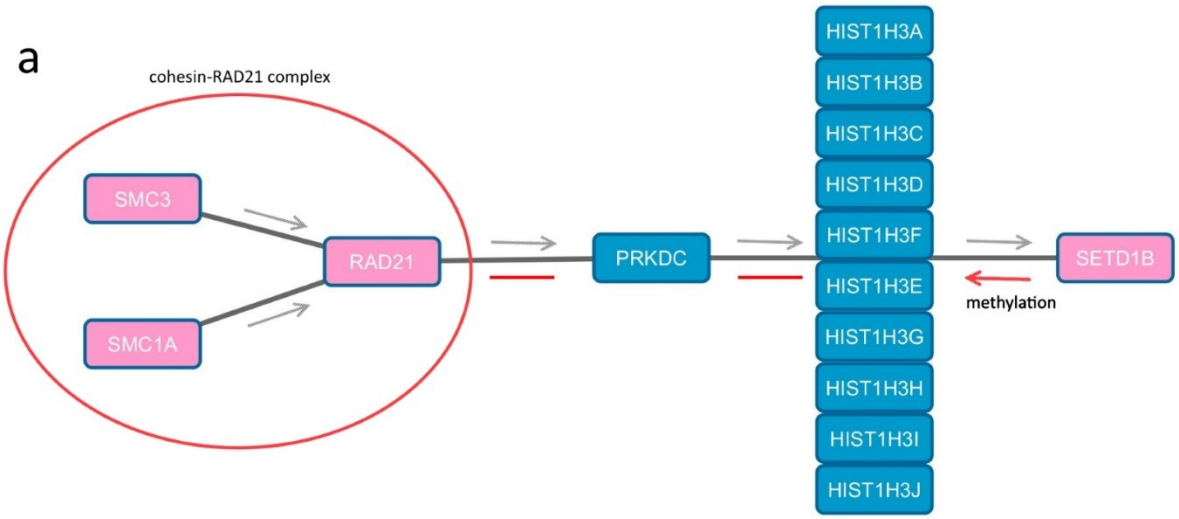
Table 2-16. DNA repair loss-of-function screening: top 50 paths with the highest centrality scores.

Centrality score	Path			Hit gene - Hit gene pair	pvalue
7	pHIST1H3A-pRBBP7	pSETD1B-pHIST1H3A		;pSMC3-pSETD1B_pp	0.0066
7	pHIST1H3J-pRBBP7	pSETD1B-pHIST1H3J		;pSMC3-pSETD1B_pp	0.0066
7	pHIST1H3I-pRBBP7	pSETD1B-pHIST1H3I		;pSMC3-pSETD1B_pp	0.0066
7	pHIST1H3H-pRBBP7	pSETD1B-pHIST1H3H		;pSMC3-pSETD1B_pp	0.0066
7	pHIST1H3G-pRBBP7	pSETD1B-pHIST1H3G		;pSMC3-pSETD1B_pp	0.0066
7	pHIST1H3D-pRBBP7	pSETD1B-pHIST1H3D		;pSMC3-pSETD1B_pp	0.0066
7	pHIST1H3F-pRBBP7	pSETD1B-pHIST1H3F		;pSMC3-pSETD1B_pp	0.0066
7	pHIST1H3B-pRBBP7	pSETD1B-pHIST1H3B		;pSMC3-pSETD1B_pp	0.0066
7	pHIST1H3E-pRBBP7	pSETD1B-pHIST1H3E		;pSMC3-pSETD1B_pp	0.0066
7	pHIST1H3C-pRBBP7	pSETD1B-pHIST1H3C		;pSMC3-pSETD1B_pp	0.0066
6	pSETDB1-pHIST1H3E	pSETD1B-pHIST1H3E		;pSMC3-pSETD1B_pp	0.0485
6	pSETDB1-pHIST1H3F	pSETD1B-pHIST1H3F		;pSMC3-pSETD1B_pp	0.0485
6	pSETDB1-pHIST1H3A	pSETD1B-pHIST1H3A		;pSMC3-pSETD1B_pp	0.0485
6	pSETDB1-pHIST1H3I	pSETD1B-pHIST1H3I		;pSMC3-pSETD1B_pp	0.0485
6	pSETDB1-pHIST1H3D	pSETD1B-pHIST1H3D		;pSMC3-pSETD1B_pp	0.0485
6	pSETDB1-pHIST1H3G	pSETD1B-pHIST1H3G		;pSMC3-pSETD1B_pp	0.0485
6	pSETDB1-pHIST1H3C	pSETD1B-pHIST1H3C		;pSMC3-pSETD1B_pp	0.0485
6	pSETDB1-pHIST1H3H	pSETD1B-pHIST1H3H		;pSMC3-pSETD1B_pp	0.0485
6	pSETDB1-pHIST1H3J	pSETD1B-pHIST1H3J		;pSMC3-pSETD1B_pp	0.0485
6	pSETDB1-pHIST1H3B	pSETD1B-pHIST1H3B		;pSMC3-pSETD1B_pp	0.0485
5	pSMC1A-pMCM6	pMCM6-pPSMA1		;pSMC1A-pPSMA4_p	0.0104
5	pCHAF1A-pHIST1H3D	pSETD1B-pHIST1H3D		;pSMC3-pSETD1B_pp	0.0238
5	pCHAF1A-pHIST1H3H	pSETD1B-pHIST1H3H		;pSMC3-pSETD1B_pp	0.0238
5	pCHAF1A-pHIST1H3C	pSETD1B-pHIST1H3C		;pSMC3-pSETD1B_pp	0.0238
5	pCHAF1A-pHIST1H3E	pSETD1B-pHIST1H3E		;pSMC3-pSETD1B_pp	0.0238
5	pCHAF1A-pHIST1H3G	pSETD1B-pHIST1H3G		;pSMC3-pSETD1B_pp	0.0238
5	pCHAF1A-pHIST1H3J	pSETD1B-pHIST1H3J		;pSMC3-pSETD1B_pp	0.0238
5	pCHAF1A-pHIST1H3A	pSETD1B-pHIST1H3A		;pSMC3-pSETD1B_pp	0.0238
5	pCHAF1A-pHIST1H3I	pSETD1B-pHIST1H3I		;pSMC3-pSETD1B_pp	0.0238
5	pCHAF1A-pHIST1H3F	pSETD1B-pHIST1H3F		;pSMC3-pSETD1B_pp	0.0238
5	pCHAF1A-pHIST1H3B	pSETD1B-pHIST1H3B		;pSMC3-pSETD1B_pp	0.0238
4	pSUMO2-pRAD21	pSUMO2-pPSMA4		;pSMC1A-pPSMA4_p	0.0061
4	pSUMO2-pSMC3	pSUMO2-pPSMA4		;pSMC1A-pPSMA4_p	0.0067
4	pHIST1H3A-pRBBP4	pSETD1B-pHIST1H3A		;pDDB1-pSETD1B_pp	0.0357
4	pHIST1H3J-pRBBP4	pSETD1B-pHIST1H3J		;pDDB1-pSETD1B_pp	0.0357
4	pHIST1H3H-pRBBP4	pSETD1B-pHIST1H3H		;pDDB1-pSETD1B_pp	0.0357
4	pHIST1H3D-pRBBP4	pSETD1B-pHIST1H3D		;pDDB1-pSETD1B_pp	0.0357
4	pHIST1H3C-pRBBP4	pSETD1B-pHIST1H3C		;pDDB1-pSETD1B_pp	0.0357
4	pHIST1H3G-pRBBP4	pSETD1B-pHIST1H3G		;pDDB1-pSETD1B_pp	0.0357
4	pHIST1H3E-pRBBP4	pSETD1B-pHIST1H3E		;pDDB1-pSETD1B_pp	0.0357
4	pHIST1H3B-pRBBP4	pSETD1B-pHIST1H3B		;pDDB1-pSETD1B_pp	0.0357
4	pHIST1H3I-pRBBP4	pSETD1B-pHIST1H3I		;pDDB1-pSETD1B_pp	0.0357
4	pHIST1H3F-pRBBP4	pSETD1B-pHIST1H3F		;pDDB1-pSETD1B_pp	0.0357
3	pRAD21-pPRKDC	pHIST1H3G-pPRKDC	pSETD1B-pHIST1H3G	;pSMC3-pSETD1B_pp	0.0039
3	pRAD21-pPRKDC	pHIST1H3C-pPRKDC	pSETD1B-pHIST1H3C	;pSMC3-pSETD1B_pp	0.0039
3	pRAD21-pPRKDC	pHIST1H3B-pPRKDC	pSETD1B-pHIST1H3B	;pSMC3-pSETD1B_pp	0.0039
3	pRAD21-pPRKDC	pHIST1H3E-pPRKDC	pSETD1B-pHIST1H3E	;pSMC3-pSETD1B_pp	0.0039
3	pRAD21-pPRKDC	pHIST1H3J-pPRKDC	pSETD1B-pHIST1H3J	;pSMC3-pSETD1B_pp	0.0039
3	pRAD21-pPRKDC	pHIST1H3H-pPRKDC	pSETD1B-pHIST1H3H	;pSMC3-pSETD1B_pp	0.0039
3	pRAD21-pPRKDC	pHIST1H3D-pPRKDC	pSETD1B-pHIST1H3D	;pSMC3-pSETD1B_pp	0.0039

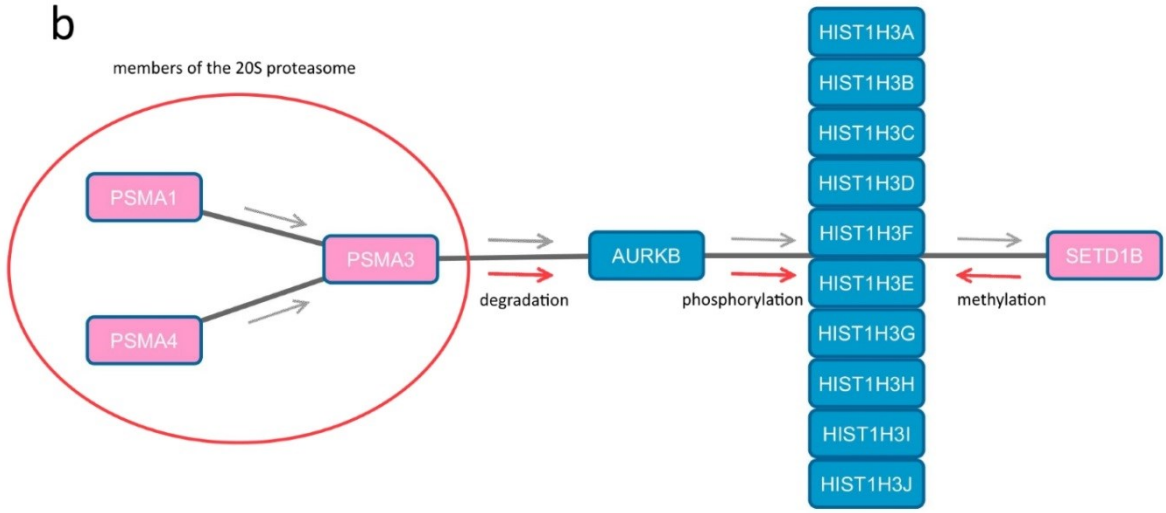
Figure 2-17. Subnetworks for two paths with centrality score 3.

The hit proteins are colored pink, the intermediate proteins are colored blue. The grey arrows show the direction of the paths when it is built by the method. The red arrows show the direction and the effect of the interaction found in literature.

a RAD21-PRKDC-histone H3-SETDB1 path.



b PSMA3-AURKB-histone H3- SETDB1 path.



3 Discussion

Two different types of networks were used in this work. The first one was a mixed direct and indirect network constructed from PPI, transcriptional, post-transcriptional and metabolic data. The second network was indirect PPI network. PPI networks are the most common networks used in network analysis. PPI networks are known to be incomplete and biased towards the well-studied proteins. Incorporating transcriptional, post-transcriptional and metabolic data, does not solve the issues associated with PPI networks, but adds information on directionality, positive or negative effect of the interactions and gives the ability to build heterogeneous paths thus allowing to study biological system at different levels.

The possible bias in the results towards highly connected nodes/paths that pass through highly connected nodes is removed by random sampling hit lists. We used the threshold of 0.05 for p-value^{net}. However, the nodes/paths with higher p-value^{net} can also be considered, it is just not guaranteed that the value of the centrality score is due to the specificity of the node/path to the phenotype or due to its high connectivity.

Other two general problems in network analysis is network specificity for the biological system and lack of information relating to some members of the hit list. Networks that represent human interactome were used in this work. However, the nodes that are not active can be excluded from the network for some biological systems, based for example on the transcriptomic data, to create a smaller but more specific network. On the other hand, the hit nodes from the functional screening might be poorly studied or might even not be present in the network. Low confidence or predicted interactions for hit nodes might be added to the network in this case. This might be particular relevant for miRNA-mRNA interactions, since miRNA are thought to be important regulators of different biological processes, but the number of experimentally validated interactions is quite low.

The notion of “final implementer” was introduced in this work. A “final implementer” was denoted as a biological component that is involved in events responsible for final phenotypical realization of the biological process. Modern molecular biology accumulated

vast amount of knowledge and for some biological processes such biological components are known, e.g. for apoptosis caspase 3, caspase 6 and caspase 7 could be considered as “final implementers”. In case such biological components are unknown, the members of the hit list can be used as a list of “final implementers” for the analysis on the PPI network and by studying directionality of the paths candidates for “final implementers” could be found as it was demonstrated for the DNA repair process.

The shortest path approach was used to retrieve connections between biological components in this work. The shortest path approach is used in many contexts in systems biology e.g it is used to construct networks for a set of genes (Yuan et al., 2017), to calculate topological properties for nodes prioritization (Zhang et al., 2012), to predict functional components and molecular pathways (Bromberg et al., 2009; Nakamura et al., 2012), to perform network modularization (Cabusora et al., 2005; Spirin et al., 2003) and to predict protein function (Arnau et al., 2017; Sharan et al., 2007). We performed an experiment to check whether this approach can give biologically valid paths by comparing the shortest paths between all possible source and target points on eight canonical pathways with the canonical counterparts. We found that the shortest paths between these points built in the human interactome constructed from HPRD or HIPPIE databases usually do not match the canonical counterpart. But we also found that the nodes and the paths with centrality score ≥2 in the network built from the shortest path between these points belong to the canonical counterpart for many pairs of source and target points.

One possible way to enhance the shortest paths approach is to find first all possible length-bounded paths (e.g with maximum length 5, most probably the maximum length should not be very big since at least signaling cascades are not long); then to use $score^{exp}$ and/or $p\text{-}value^{exp}$ to filter these paths in order to create a set of the “strongest” paths instead of the shortest (these two steps are also implemented in masterPATH). However, this is more appropriate if genome-wide loss-of-function screening data is available, because genes important for the biological prosses might not show differential expression profile.

4 Conclusion

We presented here masterPATH method. MasterPATH is a new exploratory network analysis method to find potential members of molecular pathways important for a given phenotype, which can work on both protein-protein and integrated networks. The method employs the shortest path approach, centrality score and phenotype label permutation approach to find nodes and paths with significant centrality scores in the subnetwork that is constructed from the shortest paths between hit genes and so called “final implementers” or between hit genes. Centrality score of a node or a path was defined as the number of the shortest paths found in the previous step that pass through the node and the path. “Final implementers” were defined as biological components involved in molecular events responsible for final realization of a given phenotype. It was hypothesized that the nodes and the paths with significant centrality score can be considered as putative members of molecular pathways leading to the studied phenotype. To illustrate the method, the data from the miRNA loss-of-function screening and transcriptomic profiling of terminal muscle differentiation and from ‘druggable’ loss-of-function screening of the DNA repair process were analyzed with the method. masterPATH found known and new interesting components for both biological systems. The method is implemented in Java and the source code is available on GitHub page https://github.com/daggoo/masterPATH.

Also, the shortest path approach was studied on the human interactome. It was shown that the shortest paths between source and target points located close to each other on a canonical signaling pathway usually matched the canonical counterpart. It was not true for source and target points located far apart to each other. However, nodes and paths with centrality scores ≥2 in a subnetwork constructed from the shortest paths between source and target points belonged to the canonical counterpart for a large fraction of source and target points.

References

Alstott, J., Bullmore, E., & Plenz, D. (2014). Powerlaw: A python package for analysis of heavy-tailed distributions. *PLoS ONE*, *9*(1). https://doi.org/10.1371/journal.pone.0085777

Alzhanov, D. T., McInerney, S. F., & Rotwein, P. (2010). Long range interactions regulate Igf2 gene transcription during skeletal muscle differentiation. *Journal of Biological Chemistry*, *285*(50), 38969–38977. https://doi.org/10.1074/jbc.M110.160986

Amgalan, B., & Lee, H. (2014). WMAXC: A Weighted Maximum Clique method for identifying condition-specific sub-network. *PLoS ONE*, *9*(8). https://doi.org/10.1371/journal.pone.0104993

Arnau, V., Mars, S., & Marín, I. (2017). Iterative Cluster Analysis of Protein Interaction Data, *21*(3), 364–378. https://doi.org/10.1093/bioinformatics/bti021

Barrangou, R., Fremaux, C., Deveau, H., Richards, M., Boyaval, P., Moineau, S., … Horvath, P. (2007). CRISPR Provides Acquired Resistance Against Viruses in Prokaryotes. *Science*, *315*(March), 1709–1712. https://doi.org/10.1126/science.1138140

Bartel, D. P. (2004). MicroRNAs: Genomics, Biogenesis, Mechanism, and Function. *Cell*, *116*(2), 281–297. https://doi.org/10.1016/S0092-8674(04)00045-5

Bassik, M. C., Lebbink, R. J., Churchman, L. S., Ingolia, N. T., Patena, W., Leproust, E. M., … Mcmanus, M. T. (2009). Rapid Creation and Quantitative Monitoring of High Coverage shRNA Libraries, *6*(6), 443–445. https://doi.org/10.1038/nmeth.1330.Rapid

Beißbarth, T., & Speed, T. P. (2004). GOstat: Find statistically overrepresented Gene Ontologies with a group of genes. *Bioinformatics*, *20*(9), 1464–1465. https://doi.org/10.1093/bioinformatics/bth088

Beisser, D., Klau, G. W., Dandekar, T., Müller, T., & Dittrich, M. T. (2010). BioNet: An R-Package for the functional analysis of biological networks. *Bioinformatics*, *26*(8), 1129–1130. https://doi.org/10.1093/bioinformatics/btq089

Bernstein, E., Caudy, A. a, Hammond, S. M., & Hannon, G. J. (2001). Role for bidentate ribnuclease in the initiation site of RNA interference. *Nature*, *409*(1997), 363–366.

Bindea, G., Mlecnik, B., Hackl, H., Charoentong, P., Tosolini, M., Kirilovsky, A., … Galon, J. (2009). ClueGO: A Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. *Bioinformatics*, *25*(8), 1091–1093. https://doi.org/10.1093/bioinformatics/btp101

Birmingham, A., Anderson, E. M., Reynolds, A., Ilsley-Tyree, D., Leake, D., Fedorov, Y., … Khvorova, A. (2006). 3’ UTR seed matches, but not overall identity, are associated with RNAi off-targets. *Nature Methods*, *3*(3), 199–204. https://doi.org/10.1038/nmeth854

Boch, J. (2011). TALEs of genome targeting. *Nature Publishing Group*, *29*(2), 135–136. https://doi.org/10.1038/nbt0211-135

Bolotin, A., Quinquis, B., Sorokin, A., & Dusko Ehrlich, S. (2005). Clustered regularly interspaced short palindrome repeats (CRISPRs) have spacers of extrachromosomal origin. *Microbiology*, *151*(8), 2551–2561. https://doi.org/10.1099/mic.0.28048-0

Bonnet, E., Calzone, L., Rovera, D., Stoll, G., Barillot, E., & Zinovyev, A. (2013). BiNoM 2.0, a Cytoscape plugin for accessing and analyzing pathways using standard systems biology formats. *BMC Systems Biology*, *7*. https://doi.org/10.1186/1752-0509-7-18

Bourmoum, M., Charles, R., & Claing, A. (2016). The GTPase ARF6 controls ROS production to mediate angiotensin II-induced vascular smooth muscle cell proliferation. *PLoS ONE*, *11*(1), 1–17. https://doi.org/10.1371/journal.pone.0148097

Boutros, M., & Ahringer, J. (2008). The art and design of genetic screens: RNA interference. *Nature Reviews. Genetics*, *9*(7), 554–66. https://doi.org/10.1038/nrg2364

Boutros, M., Brás, L. P., & Huber, W. (2006). Analysis of cell-based RNAi screens. *Genome Biology*, *7*(7), R66. https://doi.org/10.1186/gb-2006-7-7-R66

Bradley, G., & Barrett, S. J. (2017). CausalR - extracting mechanistic sense from genome scale data. *Bioinformatics*, *33*(November), 3670–3672. https://doi.org/10.1093/bioinformatics/btx425

Breitling, R., Amtmann, A., & Herzyk, P. (2004). Iterative Group Analysis (iGA): a simple tool to enhance sensitivity and facilitate interpretation of microarray experiments. *BMC Bioinformatics*, *5*, 34. https://doi.org/10.1186/1471-2105-5-34

Bridge, A. J., Pebernard, S., Ducraux, A., Nicoulaz, A. L., & Iggo, R. (2003). Induction of an interferon response by RNAi vectors in mammalian cells. *Nature Genetics*, *34*(3), 263–264. https://doi.org/10.1038/ng1173

Bromberg, K. D., Ma'ayan, A., Neves, S. R., & Iyengar, R. (2009). Design Logic of a Cannabinoid Receptor Signaling Network That Triggers Neurite Outgrowth. *Pharmacology*, *320*(5878), 903–909. https://doi.org/10.1126/science.1152662.Design

Brouns, S. J. J., Jore, M. M., Lundgren, M., Westra, E. R., Slijkhuis, R. J. H., Snijders, A. P. L., … van der Oost, J. (2008). Small CRISPR RNAs Guide Antiviral Defense in Prokaryotes. *Science*, *331*(August), 1541–1545.

Buehler, E., Khan, A. a, Marine, S., Rajaram, M., Bahl, A., Burchard, J., & Ferrer, M. (2012). siRNA off-target effects in genome-wide screens identify signaling pathway members. *Scientific Reports*, *2*, 428. https://doi.org/10.1038/srep00428

Cabusora, L., Sutton, E., Fulmer, A., & Forst, C. V. (2005). Differential network expression during drug and stress response. *Bioinformatics*, *21*(12), 2898–2905. https://doi.org/10.1093/bioinformatics/bti440

Callis, T. E., Chen, J.-F., & Wang, D.-Z. (2007). MicroRNAs in skeletal and cardiac muscle development. *DNA and Cell Biology*, *26*(4), 219–25. https://doi.org/10.1089/dna.2006.0556

Carthew, Richard W. and Sontheimer, E. J. (2009). Origins and Mechanisms of miRNAs and siRNAs. *Cell*, *136*(4), 642–655. https://doi.org/10.1016/j.cell.2009.01.035.Origins

Chen, L., Xuan, J., Riggins, R. B., Wang, Y., & Clarke, R. (2013). Identifying protein interaction subnetworks by a bagging Markov random field-based method. *Nucleic Acids Research*, *41*(2), 1–13. https://doi.org/10.1093/nar/gks951

Chen, Y., Mccarthy, D., Robinson, M., & Smyth, G. K. (2015). edgeR : differential expression analysis of digital gene expression data User ' s Guide, (April).

Chindelevitch, L., Ziemek, D., Enayetallah, A., Randhawa, R., Sidders, B., Brockel, C., & Huang, E. S. (2012). Causal reasoning on biological networks: Interpreting transcriptional changes. *Bioinformatics*, *28*(8), 1114–1121. https://doi.org/10.1093/bioinformatics/bts090

Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D., & Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, *3*(140), 1–10. https://doi.org/10.1038/msb4100180

Cong, L., Ran, F. A., Cox, D., Lin, S., Barretto, R., Hsu, P. D., … Marraffini, L. a. (2013). Multiplex Genome Engineering Using CRISPR/VCas Systems. *Science*, *339*(6121), 819–823. https://doi.org/10.1126/science.1231143.Multiplex

Cornish, A. J., & Markowetz, F. (2014). SANTA: Quantifying the Functional Content of Molecular Networks. *PLoS Computational Biology*, *10*(9). https://doi.org/10.1371/journal.pcbi.1003808

Coux, O., Tanaka, K., & Goldberg, a L. (1996). Structure and functions of the 20S and 26S proteasomes. *Annual Review of Biochemistry*, *65*, 801–847. https://doi.org/10.1146/annurev.biochem.65.1.801

Crist, C. G., & Buckingham, M. (2009). microRNas gain magnitude in muscle. *Cell Cycle*, *8*(22), 3627–3628. https://doi.org/10.4161/cc.8.22.9960

Crosio, C., Fimia, G. M., Loury, R., Okano, Y., Zhou, H., Sen, S., … Sassone-corsi, P. (2002). Mitotic Phosphorylation of Histone H3 : Spatio-Temporal Regulation by Mammalian Aurora Kinases Mitotic Phosphorylation of Histone H3 : Spatio-Temporal Regulation by Mammalian Aurora Kinases. *Molecular and Cellular Biology*, *22*(3), 874–885. https://doi.org/10.1128/MCB.22.3.874

Daneholt, B. (2006). The discovery of RNA interference. *Karolinska Institute*, 1–10. https://doi.org/10.1038/nature07754

DeRisi, J., Penland, L., & Brown, P. O. (1996). Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nat Genet.*, *14*(3), 353–6. https://doi.org/10.1038/ng0496-417

Dion, V., Kalck, V., Horigome, C., Towbin, B. D., & Gasser, S. M. (2012). Increased mobility of double-strand breaks requires Mec1, Rad9 and the homologous recombination machinery. *Nature Cell Biology*, *14*(5), 502–509. https://doi.org/10.1038/ncb2465

Dittrich, M. T., Klau, G. W., Rosenwald, A., Dandekar, T., & Müller, T. (2008). Identifying functional modules in protein-protein interaction networks: An integrated exact approach. *Bioinformatics*, *24*(13), 223–231. https://doi.org/10.1093/bioinformatics/btn161

Doench, J. G., Petersen, C. P., & Sharp, P. a. (2003). siRNAs can function as miRNAs. *Genes and Development*, *17*(4), 438–442. https://doi.org/10.1101/gad.1064703

Doherty, K. R., Demonbreun, A. R., Wallace, G. Q., Cave, A., Posey, A. D., Heretis, K., … McNally, E. M. (2008). The endocytic recycling protein EHD2 interacts with myoferlin to regulate myoblast fusion. *Journal of Biological Chemistry*, *283*(29), 20252–20260. https://doi.org/10.1074/jbc.M802306200

Dutta, B., Azhir, A., Merino, L.-H., Guo, Y., Revanur, S., Madhamshettiwar, P. B., … Fraser, I. D. C. (2016). An interactive web-based application for Comprehensive Analysis of RNAi-screen Data. *Nature Communications*, *7*, 10578. https://doi.org/10.1038/ncomms10578

Elbashir, S. M., Harborth, J., Lendeckel, W., Yalcin, A., Weber, K., & Tuschl, T. (2001). Duplexes of 21-nucleotide RNAs mediate RNA interference in cultured mammalian cells, *411*(May), 1–5.

Elbashir, S. M., Lendeckel, W., & Tuschl, T. (2001). RNA interference is mediated 21- and 22-nucleotide RNAs. *Genes Dev.*, *15*, 188–200. https://doi.org/10.1101/gad.862301.vents

Elbashir, S. M., Martinez, J., Patkaniowska, A., Lendeckel, W., & Tuschl, T. (2001). Functional anatomy of siRNAs for mediating efficient RNAi in Drosophila melanogaster embryo lysate. *EMBO Journal*, *20*(23), 6877–6888. https://doi.org/10.1093/emboj/20.23.6877

Essaghir, A., Toffalini, F., Knoops, L., Kallin, A., van Helden, J., & Demoulin, J. B. (2010). Transcription factor regulation can be accurately predicted from the presence of target gene signatures in microarray gene expression data. *Nucleic Acids Research*, *38*(11). https://doi.org/10.1093/nar/gkq149

Fazekas, D., Koltai, M., Türei, D., Módos, D., Pálfy, M., Dúl, Z., … Korcsmáros, T. (2013). SignaLink 2 - a signaling pathway resource with multi-layered regulatory networks. *BMC Systems Biology*, *7*, 7. https://doi.org/10.1186/1752-0509-7-7

Fernández, A. M., Dupont, J., Farrar, R. P., Lee, S., Stannard, B., & Roith, D. Le. (2002). Muscle-specific inactivation of the IGF-I receptor induces compensatory hyperplasia in skeletal muscle. *Journal of Clinical Investigation*, *109*(3), 347–355. https://doi.org/10.1172/JCI200213503

Fire, a, Xu, S., Montgomery, M. K., Kostas, S. a, Driver, S. E., & Mello, C. C. (1998). Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans. *Nature*, *391*(6669), 806–811. https://doi.org/10.1038/35888

Fodor, S. P. A., Read, J. L., Pirrung, M. C., Stryer, L., Lu, T., & Solas, D. (1991). Light-Directed, Spatially Addressable Parallel Chemical Synthesis, (February), 767–774.

Garneau, J. E., Dupuis, M. È., Villion, M., Romero, D. a., Barrangou, R., Boyaval, P., … Moineau, S. (2010). The CRISPR/cas bacterial immune system cleaves bacteriophage and plasmid DNA. *Nature*, *468*(7320), 67–71. https://doi.org/10.1038/nature09523

Gautier, L., Cope, L., Bolstad, B. M., & Irizarry, R. a. (2004). Affy - Analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, *20*(3), 307–315. https://doi.org/10.1093/bioinformatics/btg405

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., & Dudoit, S. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Uesc.Br*, *5*(10), R80. https://doi.org/10.1186/gb-2004-5-10-r80

Glaab, E., Baudot, A., Krasnogor, N., Schneider, R., & Valencia, A. (2012). EnrichNet: Network-based gene set enrichment analysis. *Bioinformatics*, *28*(18), 451–457. https://doi.org/10.1093/bioinformatics/bts389

Grimm, D., Streetz, K. L., Jopling, C. L., Storm, T. a., Pandey, K., Davis, C. R., … Kay, M. a. (2006). Fatality in mice due to oversaturation of cellular microRNA/short hairpin RNA pathways. *Nature*, *441*(7092), 537–541. https://doi.org/10.1038/nature04791

Grissa, I., Vergnaud, G., & Pourcel, C. (2007). The CRISPRdb database and tools to display CRISPRs and to generate dictionaries of spacers and repeats. *BMC Bioinformatics*, *8*, 1–10. https://doi.org/10.1186/1471-2105-8-172

Groenen, P. M. a., Bunschoten, A. E., Soolingen, D. Van, & Errtbden, J. D. A. va. (1993). Nature of DNA polymorphism in the direct repeat cluster of Mycobacterium tuberculosis; application for strain differentiation by a novel typing method. *Molecular Microbiology*, *10*(5), 1057–1065. https://doi.org/10.1111/j.1365-2958.1993.tb00976.x

Grüning, B. a., Fallmann, J., Yusuf, D., Will, S., Erxleben, A., Eggenhofer, F., … Backofen, R. (2017). The RNA workbench: Best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. *Nucleic Acids Research*, *45*(W1), W560–W566. https://doi.org/10.1093/nar/gkx409

Gu, J., Chen, Y., Li, S., & Li, Y. (2010). Identification of responsive gene modules by network-based gene clustering and extending: Application to inflammation and angiogenesis. *BMC Systems Biology*, *4*. https://doi.org/10.1186/1752-0509-4-47

Gunning, P. W., Ferguson, V., Brennan, K. J., & Hardeman, E. C. (2001). Alpha-skeletal actin induces a subset of muscle genes independently of muscle differentiation and withdrawal from the cell cycle. *Journal of Cell Science*, *114*(Pt 3), 513–24. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/11171321

Guyon, L., Lajaunie, C., Fer, F., Bhajun, R., Sulpice, E., Pinna, G., … Gidrol, X. (2015). Φ-score: A cell-to-cell phenotypic scoring method for sensitive and selective hit discovery in cell-based assays. *Scientific Reports*, *5*(1), 14221. https://doi.org/10.1038/srep14221

Hale, C. R., Zhao, P., Olson, S., & Duff, M. O. (2009). RNA-Guided RNA Cleavage by a CRISPR RNA-Cas Protein Complex, *139*(5), 945–956. https://doi.org/10.1016/j.cell.2009.07.040.RNA-Guided

Hammond, S. M., Bernstein, E., Beach, D., & Hannon, G. J. (2000). An RNA-directed nuclease mediates post-transcriptional gene silencing in Drosophila cells. *Nature*, *404*(6775), 293–296. https://doi.org/10.1038/35005107

Hauer, M. H., Seeber, A., Singh, V., Thierry, R., Sack, R., Amitai, A., … Gasser, S. M. (2017). Histone degradation in response to DNA damage enhances chromatin dynamics and recombination rates. *Nature Structural & Molecular Biology*, *24*(2), 99–107. https://doi.org/10.1038/nsmb.3347

HIPPIE Howto. (n.d.). Retrieved from http://cbdm-01.zdv.uni-mainz.de/~mschaefer/hippie/information.php

Hsu, S. Da, Lin, F. M., Wu, W. Y., Liang, C., Huang, W. C., Chan, W. L., … Huang, H. Da. (2011). MiRTarBase: A database curates experimentally validated microRNA-target interactions. *Nucleic Acids Research*, *39*(SUPPL. 1), 163–169. https://doi.org/10.1093/nar/gkq1107

Huang, D. W., Sherman, B. T., & Lempicki, R. a. (2009). Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, *4*(1), 44–57. https://doi.org/10.1038/nprot.2008.211

Hwang, W. Y., Fu, Y., Reyon, D., Maeder, M. L., Shengdar, Q., Sander, J. D., … Keith, J. (2013). Efficient In Vivo Genome Editing Using RNA-Guided Nucleases Woong. *Nature Biotechnology*, *31*(3), 227–229. https://doi.org/10.1038/nbt.2501.Efficient

Ideker, T., Ozier, O., Schwikowski, B., & Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics (Oxford, England)*, *18 Suppl 1*, S233–S240. https://doi.org/10.1093/bioinformatics/18.suppl_1.S233

Ishino, Y., Shinagawa, H., Makino, K., Amemura, M., & Nakata, a. (1987). Nucleotide sequence of the iap gene, responsible for alkaline phosphatase isozyme conversion in Escherichia coli, and identification of the gene product. *Journal of Bacteriology*, *169*(12), 5429–5433. https://doi.org/10.1128/jb.169.12.5429-5433.1987

Jackson, A. L., Bartz, S. R., Schelter, J., Kobayashi, S. V, Burchard, J., Mao, M., … Linsley, P. S. (2003). Expression profiling reveals off-target gene regulation by RNAi. *Nature Biotechnology*, *21*(6), 635–638.

Jansen, R., Van Embden, J. D. a, Gaastra, W., & Schouls, L. M. (2002). Identification of genes that are associated with DNA repeats in prokaryotes. *Molecular Microbiology*, *43*(6), 1565–1575. https://doi.org/10.1046/j.1365-2958.2002.02839.x

Jinek, M., Chylinski, K., Fonfara, I., Hauer, M., Doudna, J. a, & Charpentier, E. (2012). A Programmable Dual-RNA–Guided DNA Endonuclease in Adaptive Bacterial Immunity, *337*(August), 816–822. https://doi.org/10.1126/science.1225829

Kairov, U., Karpenyuk, T., Ramanculov, E., & Zinovyev, A. (2012). Network analysis of gene lists for finding reproducible prognostic breast cancer gene signatures. *Bioinformation*, *8*(16), 773–776. https://doi.org/10.6026/97320630008773

Kaplow, I. M., Singh, R., Friedman, A., Bakal, C., Perrimon, N., & Berger, B. (2009). RNAiCut: automated detection of significant genes from functional genomic screens. *Nature Methods*, *6*(7), 476–7. https://doi.org/10.1038/nmeth0709-476

Khan, A. a, Betel, D., Miller, M. L., Sander, C., Leslie, C. S., & Debora, S. (2009). Transfection of small RNAs globally perturbs gene regulation by endogenous microRNAs, *27*(6), 549–555. https://doi.org/10.1038/nbt.1543.Transfection

Khatri, P., Sirota, M., Butte, A. J., Packard, L., Road, W., & Alto, P. (2012). 10 Years of Pathway Analysis : Current Approaches and Outstanding Challenges - Table S2, 10.

Koonin, E. V., Makarova, K. S., & Zhang, F. (2017). Diversity, classification and evolution of CRISPR-Cas systems. *Current Opinion in Microbiology*, *37*, 67–78. https://doi.org/10.1016/j.mib.2017.05.008

Kropp, J., Degerny, C., Morozova, N., Pontis, J., Harel-Bellan, A., & Polesskaya, A. (2015). miR-98 delays skeletal muscle differentiation by down-regulating E2F5. *Biochemical Journal*, *466*(1), 85–93. https://doi.org/10.1042/BJ20141175

Kuleshov, M. V., Jones, M. R., Rouillard, A. D., Fernandez, N. F., Duan, Q., Wang, Z., … Ma'ayan, A. (2016). Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Research*, *44*(W1), W90–W97. https://doi.org/10.1093/nar/gkw377

Lagos-Quintana, M., Rauhut, R., Lendeckel, W., & Tuschl, T. (2001). Identification of novel genes Coding for RNAs of Small expressed RNAs. *Science*, *294*(5543), 853–858. https://doi.org/10.1126/science.1064921

Lau, N. C., Lim, L. P., Weinstein, E. G., & Bartel, D. P. (2001). An abundant class of tiny RNAs with probably regulatory roles in Caenorhabditis elegans. *Science*, *294*(October), 858–862.

Lee, R. C., & Ambros, V. (2001). An extensive class of small RNAs in Caenorhabditis elegans. *Science*, *294*(OCTOBER), 862–864.

Lee, R. C., Feinbaum, R. L., & Ambrost, V. (1993). The C. elegans Heterochronic Gene lin-4 Encodes Small RNAs with Antisense Complementarity to lin-14. *Cell*, *75*, 843–854. https://doi.org/10.1016/0092-8674(93)90529-Y

Lee, S.-J. (2004). Regulation of Muscle Mass By Myostatin. *Annual Review of Cell and Developmental Biology*, *20*(1), 61–86. https://doi.org/10.1146/annurev.cellbio.20.012103.135836

Lee, Y., Ahn, C., Han, J., Choi, H., Kim, J., Yim, J., … Kim, V. N. (2003). The nuclear RNase III Drosha initiates microRNA processing. *Nature*, *425*(6956), 415–419. https://doi.org/10.1038/nature01957

Lee, Y., Jeon, K., Lee, J.-T., Kim, S., & Kim, V. N. (2002). MicroRNA maturation: stepwisee processing and subcellular localization. *The EMBO Journal*, *21*(17), 4663–4670. https://doi.org/10.1093/emboj/cdf476

Li, W., Xu, H., Xiao, T., Cong, L., Love, M. I., Zhang, F., … Liu, X. S. (2014). MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. *Genome Biology*, *15*(12), 554. https://doi.org/10.1186/s13059-014-0554-4

Losada, A. (2014). Cohesin in cancer: chromosome segregation and beyond. *Nature Reviews Cancer*, *14*(6), 389–393. https://doi.org/10.1038/nrc3743

Love, M. I., Anders, S., Kim, V., & Huber, W. (2015). RNA-Seq workflow: gene-level exploratory analysis and differential expression. *F1000Research*, *4*, 1070. https://doi.org/10.12688/f1000research.7035.1

Lowe, R., Shirley, N., Bleackley, M., Dolan, S., & Shafee, T. (2017). Transcriptomics technologies. *PLoS Computational Biology*, *13*(5), 1–23. https://doi.org/10.1371/journal.pcbi.1005457

Lund, E., Güttinger, S., Calado, A., Dahlberg, J. E., & Kutay, U. (2004). Nuclear Export of MicroRNA. *Science*, *303*(January), 95–98. https://doi.org/10.1126/science.1090599

Ma, H., Schadt, E. E., Kaplan, L. M., & Zhao, H. (2011). COSINE: COndition-SpecIfic sub-NEtwork identification using a global optimization method. *Bioinformatics*, *27*(9), 1290–1298. https://doi.org/10.1093/bioinformatics/btr136

Ma, S., Jiang, T., & Jiang, R. (2014). Differential regulation enrichment analysis via the integration of transcriptional regulatory network and gene expression data. *Bioinformatics*, *31*(4), 563–571. https://doi.org/10.1093/bioinformatics/btu672

Maeder, M. L., Linder, S. J., Cascio, V. M., Fu, Y., Ho, Q. H., & Joung, J. K. (2014). CRISPR RNA-guided activation of endogenous human genes, *10*(10), 977–979. https://doi.org/10.1038/nmeth.2598.CRISPR

Makarova, K. S., Brouns, S. J. J., Horvath, P., Sas, D. F., & Wolf, Y. I. (2011). Evolution and classification of the CRISPR-Cas systems. *Nature Reviews …*, *9*(6), 467–477. https://doi.org/10.1038/nrmicro2577.Evolution

Makarova, K. S., Wolf, Y. I., Alkhnbashi, O. S., Costa, F., Shah, S. a, Saunders, S. J., … Garrett, R. a. (2015). An updated evolutionary classification of CRISPR–Cas systems, *13*(11), 722–736. https://doi.org/10.1038/nrmicro3569.An

Mali, P., Yang, L., Esvelt, K. M., Aach, J., Guell, M., DiCarlo, J. E., … Church, G. M. (2013). RNA-Guided Human Genome Engineering via Cas9 Prashant. *Science*, *339*(6121), 823–826. https://doi.org/10.1126/science.1232033.RNA-Guided

Marraffini, L. A., & Sontheimer, E. J. (2008). CRISPR Interference Limits Horizontal Gene Transfer in Staphylococci by Targeting DNA. *Science*, *322*, 1843–1845. https://doi.org/10.1016/j.pestbp.2011.02.012.Investigations

Martinez, J., Patkaniowska, A., Urlaub, H., Lührmann, R., & Tuschl, T. (2002). Single-stranded antisense siRNAs guide target RNA cleavage in RNAi. *Cell*, *110*(5), 563–574. https://doi.org/10.1016/S0092-8674(02)00908-X

Matzke, M. a, Primig, M., Trnovsky, J., & Matzke, a J. (1989). Reversible methylation and inactivation of marker genes in sequentially transformed tobacco plants. *The EMBO Journal*, *8*(3), 643–9.

McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, M. (n.d.). Online Mendelian Inheritance in Man, OMIM®. Retrieved from https://omim.org/

Meyer, S. U., Thirion, C., Polesskaya, A., Bauersachs, S., Kaiser, S., Krause, S., & Pfaffl, M. W. (2015). TNF-α and IGF1 modify the microRNA signature in skeletal muscle cell differentiation. *Cell Communication and Signaling*, *13*(1), 1–14. https://doi.org/10.1186/s12964-015-0083-0

Miles, L. A., Garippa, R. J., & Poirier, J. T. (2016). Design, execution, and analysis of pooled in vitro CRISPR/Cas9 screens. *FEBS Journal*, *283*, 3170–3180. https://doi.org/10.1111/febs.13770

Miné-Hattab, J., & Rothstein, R. (2012). Increased chromosome mobility facilitates homology search during recombination. *Nature Cell Biology*, *14*(5), 510–517. https://doi.org/10.1038/ncb2472

Mohr, S., Bakal, C., & Perrimon, N. (2010). Genomic Screening with RNAi: Results and Challenges, *16*(3), 387–393. https://doi.org/10.1016/j.cmet.2012.08.002.

Mohr, S. E., Smith, J. A., Shamu, C. E., Neumüller, R. a., & Perrimon, N. (2014). RNAi screening comes of age: improved techniques and complementary approaches. *Nature Reviews Molecular Cell Biology*, *15*(9), 591–600. https://doi.org/10.1038/nrm3860

Mojica, F. J. M., Díez-Villaseñor, C., García-Martínez, J., & Soria, E. (2005). Intervening sequences of regularly spaced prokaryotic repeats derive from foreign genetic elements. *Journal of Molecular Evolution*, *60*(2), 174–182. https://doi.org/10.1007/s00239-004-0046-3

Mojica, F. J. M., Díez-Villaseñor, C., Soria, E., & Juez, G. (2000). Biological significance of a family of regularly spaced repeats in the genomes of Archaea, Bacteria and

mitochondria. *Molecular Microbiology*, *36*(1), 244–246. https://doi.org/10.1046/j.1365-2958.2000.01838.x

Mojica, F. J. M., Ferrer, C., Juez, G., & Rodríguez-Valera, F. (1995). Long stretches of short tandem repeats are present in the largest replicons of the Archaea Haloferax mediterranei and Haloferax volcanii and could be involved in replicon partitioning. *Molecular Microbiology*, *17*(1), 85–93. https://doi.org/10.1111/j.1365-2958.1995.mmi_17010085.x

Mojica, F. J. M., Juez, G., & Rodriguez-Valera, F. (1993). Transcription at different salinities of Haloferax mediterranei sequences adjacent to partially modified PstI sites. *Molecular Microbiology*, *9*(3), 613–621. https://doi.org/10.1111/j.1365-2958.1993.tb01721.x

Montgomery, M. K., Xu, S., & Fire, a. (1998). RNA as a target of double-stranded RNA-mediated genetic interference in Caenorhabditis elegans. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(26), 15502–7. https://doi.org/10.1073/pnas.95.26.15502

Nakamura, M., Hachiya, T., Saito, Y., Sato, K., & Sakakibara, Y. (2012). An efficient algorithm for de novo predictions of biochemical pathways between chemical compounds. *BMC Bioinformatics*, *13 Suppl 1*(Suppl 17), S8. https://doi.org/10.1186/1471-2105-13-S17-S8

Nakao, R., Hirasaka, K., Goto, J., Ishidoh, K., Yamada, C., Ohno, a., … Nikawa, T. (2009). Ubiquitin Ligase Cbl-b Is a Negative Regulator for Insulin-Like Growth Factor 1 Signaling during Muscle Atrophy Caused by Unloading. *Molecular and Cellular Biology*, *29*(17), 4798–4811. https://doi.org/10.1128/MCB.01347-08

Napoli, C. (1990). Introduction of a Chimeric Chalcone Synthase Gene into Petunia Results in Reversible Co-Suppression of Homologous Genes in trans. *The Plant Cell Online*, *2*(4), 279–289. https://doi.org/10.1105/tpc.2.4.279

Nicholas, G., Thomas, M., Langley, B., Somers, W., Patel, K., Fred Kemp, C., … Kambadur, R. (2002). Titin-cap associates with, and regulates secretion of, myostatin. *Journal of Cellular Physiology*, *193*(1), 120–131. https://doi.org/10.1002/jcp.10158

Odho, Z., Southall, S. M., & Wilson, J. R. (2010). Characterization of a novel WDR5-binding site that recruits RbBP5 through a conserved motif to enhance methylation of histone H3 lysine 4 by mixed lineage leukemia protein-1. *Journal of Biological Chemistry*, *285*(43), 32967–32976. https://doi.org/10.1074/jbc.M110.159921

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., & Kanehisa, M. (1999). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, *27*(1), 29–34. https://doi.org/10.1093/nar/27.1.29

Panier, S., & Boulton, S. J. (2013). Double-strand break repair: 53BP1 comes into focus. *Nature Reviews Molecular Cell Biology*, *15*(1), 7–18. https://doi.org/10.1038/nrm3719

Parrish, S., Fleenor, J., Xu, S., Mello, C., & Fire, a. (2000). Functional anatomy of a dsRNA trigger: differential requirement for the two trigger strands in RNA interference. *Molecular Cell*, *6*(5), 1077–1087. https://doi.org/S1097-2765(00)00106-4 [pii]

Perez-Pinera, P., Kocak, D. D., Vockley, C. M., Adler, A. F., Kabadi, M., Polstein, L. R., … Gersbach, C. a. (2013). Transcription Factors. *Nature Methods*, *10*(10), 973–976. https://doi.org/10.1038/nmeth.2600.RNA-guided

Peri, S., Navarro, J. D., Amanchy, R., Kristiansen, T. Z., Jonnalagadda, C. K., Surendranath, V., … Pandey, A. (2003). Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research*, *13*(10), 2363–2371. https://doi.org/10.1101/gr.1680803

Persengiev, S. P., Zhu, X., & Green, M. R. (2004). Nonspecific, concentration-dependent stimulation and repression of mammalian gene expression by small interfering RNAs (siRNAs). *Rna*, *10*(1), 12–8. https://doi.org/10.1261/rna5160904.RNAi

Polesskaya, A., Degerny, C., Pinna, G., Maury, Y., Kratassiouk, G., Mouly, V., … Harel-Bellan, A. (2013). Genome-Wide Exploration of miRNA Function in Mammalian Muscle Cell Differentiation. *PLoS ONE*, *8*(8), 1–10. https://doi.org/10.1371/journal.pone.0071927

Pourcel, C., Salvignol, G., & Vergnaud, G. (2005). CRISPR elements in Yersinia pestis acquire new repeats by preferential uptake of bacteriophage DNA, and provide additional tools for evolutionary studies. *Microbiology*, *151*(3), 653–663. https://doi.org/10.1099/mic.0.27437-0

Rauscher, B., Heigwer, F., Breinig, M., Winter, J., & Boutros, M. (2016). GenomeCRISPR - a database for high-throughput CRISPR/Cas9 screens. *Genome Biology and Evolution*, *45*(October 2016), 1–8. https://doi.org/10.1093/gbe/evw245

Reinhart, B. J., Slack, F. J., Basson, M., Pasquinelll, A. E., Bettlnger, J. C., Rougvle, A. E., … Ruvkun, G. (2000). The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans. *Nature*, *403*(6772), 901–906. https://doi.org/10.1038/35002607

Rieber, N., Knapp, B., Eils, R., & Kaderali, L. (2009). RNAither, an automated pipeline for the statistical analysis of high-throughput RNAi screens. *Bioinformatics*, *25*(5), 678–679. https://doi.org/10.1093/bioinformatics/btp014

Robinson, S., Nevalainen, J., Pinna, G., Campalans, A., Radicella, J. P., & Guyon, L. (2017). Incorporating interaction networks into the determination of functionally related hit genes in genomic experiments with Markov random fields. *Bioinformatics*, *33*(14), i170–i179. https://doi.org/10.1093/bioinformatics/btx244

Romano, N., & Macino, G. (1992). Quelling: transient inactivation of gene expression in Neurospora crassa by transformation with homologous sequences. *Molecular Microbiology*, *6*(22), 3343–3353. https://doi.org/10.1111/j.1365-2958.1992.tb02202.x

Ruthenburg, A. J., Wang, W., Graybosch, D. M., Li, H., Allis, C. D., Patel, D. J., & Verdine, G. L. (2016). Histone H3 recognition and presentation by the WDR5 module of the MLL1 complex, *13*(8), 704–712. https://doi.org/10.1038/nsmb1119.Histone

Schaefer, M. H., Fontaine, J. F., Vinayagam, A., Porras, P., Wanker, E. E., & Andrade-Navarro, M. a. (2012). Hippie: Integrating protein interaction networks with experiment based quality scores. *PLoS ONE*, *7*(2), 1–8. https://doi.org/10.1371/journal.pone.0031826

Schultz, D. C., Ayyanathan, K., Negorev, D., Maul, G. G., & Rauscher Iii, F. J. (2002). SETDB1 : a novel KAP - 1 - associated histone H3 , lysine 9 - specific methyltransferase that contributes to HP1 - mediated silencing of euchromatic genes by KRAB zinc - finger proteins. *Genes and Development*, *16*, 919–932. https://doi.org/10.1101/gad.973302

Semizarov, D., Frost, L., Sarthy, a, Kroeger, P., Halbert, D., & Fesik, S. (2003). Specificity of short interfering RNA determined through gene expression signatures. *Proc Natl Acad Sci U S A*, *100*(11), 6347–6352. https://doi.org/10.1073/pnas.1131959100

Shalem, O., Sanjana, N. E., Zhang, F., & Sciences, C. (2015). High-throughput functional genomics using CRISPR-Cas9, *16*(5), 299–311. https://doi.org/10.1038/nrg3899.High-throughput

Shamir, R., Maron-Katz, A., Tanay, A., Linhart, C., Steinfeld, I., Sharan, R., … Elkon, R. (2005). EXPANDER--an integrative program suite for microarray data analysis. *BMC Bioinformatics*, *6*, 232. https://doi.org/10.1186/1471-2105-6-232

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., … Ideker, T. (2003). Cytoscape: A software Environment for integrated models of biomolecular interaction networks. *Genome Research*, *13*(11), 2498–2504. https://doi.org/10.1101/gr.1239303

Sharan, R., Ulitsky, I., & Shamir, R. (2007). Network-based prediction of protein function. *Molecular Systems Biology*, *3*(88). https://doi.org/10.1038/msb4100129

Sharma, S., & Rao, A. (2009). RNAi screening: tips and techniques. *Nat. Immunnol.*, *10*(8), 799–804. https://doi.org/10.1038/ni0809-799.RNAi

Shu, F., Guo, S., Dang, Y., Qi, M., Zhou, G., Guo, Z., … Yu, L. (2003). Human Aurora-B binds to a proteasome α-subunit HC8 and undergoes degradation in a proteasome-dependent manner. *Molecular and Cellular Biochemistry*, *254*(1-2), 157–162. https://doi.org/10.1023/A:1027317014159

Sigoillot, F. D., & King, R. W. (2011). Vigilance and Validation: Keys to Success in RNAi Screening, *6*(1), 47–60. https://doi.org/10.1021/cb100358f.Vigilance

Sigoillot, F. D., Lyman, S., Huckins, J. F., Adamson, B., Quattrochi, B., & King, R. W. (2012). A Bioinformatics Method Identifies Prominent Off-targeted Transcripts in RNAi Screens, *9*(4), 363–366. https://doi.org/10.1038/nmeth.1898.A

Sims, D., Mendes-Pereira, A. M., Frankum, J., Burgess, D., Cerone, M.-A., Lombardelli, C., … Root, D. (2011). High-throughput RNA interference screening using pooled shRNA libraries and next generation sequencing. *Genome Biology*, *12*(10), R104. https://doi.org/10.1186/gb-2011-12-10-r104

Sledz, C. a., Holko, M., De Veer, M. J., Silverman, R. H., & Williams, B. R. G. (2003). Activation of the interferon system by short-interfering RNAs. *Nature Cell Biology*, *5*(9), 834–839. https://doi.org/10.1038/ncb1038

Spirin, V., Mirny, L. A., Nature, A. W., & C, C. (2003). Protein complexes and functional modules in molecular networks, *100*(21), 12123–12128.

Strzyz, P. (2017). DNA repair: Histones have got to go. *Nature Reviews Molecular Cell Biology*, *18*(3), 138–139. https://doi.org/10.1038/nrm.2017.2

Subramanian, a., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. a., … Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, *102*(43), 15545–15550. https://doi.org/10.1073/pnas.0506580102

Surdo, P. Lo, Calderone, A., Cesareni, G., & Perfetto, L. (2017). SIGNOR: A database of causal relationships between biological entities-a short guide to searching and browsing.

Current Protocols in Bioinformatics, *2017*(June), 8.23.1–8.23.16. https://doi.org/10.1002/cpbi.28

Tachibana, I., & Hemler, M. E. (1999). Role of transmembrane 4 superfamily (TM4SF) proteins CD9 and CD81 in muscle cell fusion and myotube maintenance. *Journal of Cell Biology*, *146*(4), 893–904. https://doi.org/10.1083/jcb.146.4.893

Tarca, A. L., Draghici, S., Khatri, P., Hassan, S. S., Mittal, P., Kim, J. S., … Romero, R. (2009). A novel signaling pathway impact analysis. *Bioinformatics*, *25*(1), 75–82. https://doi.org/10.1093/bioinformatics/btn577

Tian, L. F., Li, H. Y., Jin, B. F., Pan, X., Man, J. H., Zhang, P. J., … Zhang, X. M. (2006). MDM2 interacts with and downregulates a sarcomeric protein, TCAP. *Biochemical and Biophysical Research Communications*, *345*(1), 355–361. https://doi.org/10.1016/j.bbrc.2006.04.108

Tuschl, T., Zamore, P. D., Lehmann, R., Bartel, D. P., & Sharp, P. a. (1999). Targeted mRNA degradation by double-stranded RNA in vitro. *Genes and Development*, *13*(24), 3191–3197. https://doi.org/10.1101/gad.13.24.3191

Urnov, F. D., Rebar, E. J., Holmes, M. C., Zhang, H. S., & Gregory, P. D. (2010). Genome editing with engineered zinc finger nucleases. *Nature Reviews Genetics*, *11*(9), 636–646. https://doi.org/10.1038/nrg2842

Van der Krol, a. R. (1990). Flavonoid Genes in Petunia: Addition of a Limited Number of Gene Copies May Lead to a Suppression of Gene Expression. *The Plant Cell Online*, *2*(4), 291–299. https://doi.org/10.1105/tpc.2.4.291

Van Soolingen, D., De Haas, P. E. W., Hermans, P. W. M., Groenen, P. M. a, & Van Embden, J. D. a. (1993). Comparison of various repetitive DNA elements as genetic markers for strain differentiation and epidemiology of Mycobacterium tuberculosis. *Journal of Clinical Microbiology*, *31*(8), 1987–1995.

Vasyutina, E., Martarelli, B., Brakebusch, C., Wende, H., & Birchmeier, C. (2009). The small G-proteins Rac1 and Cdc42 are essential for myoblast fusion in the mouse. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(22), 8935–40. https://doi.org/10.1073/pnas.0902501106

Wang, J., Lu, M., Qiu, C., & Cui, Q. (2009). TransmiR: A transcription factor microRNA regulation database. *Nucleic Acids Research*, *38*(SUPPL.1), 119–122. https://doi.org/10.1093/nar/gkp803

Wang, L., Tu, Z., & Sun, F. (2009). A network-based integrative approach to prioritize reliable hits from multiple genome-wide RNAi screens in Drosophila. *BMC Genomics*, *10*, 220. https://doi.org/10.1186/1471-2164-10-220

Wang, Y., Szczesna-Cordary, D., Craig, R., Diaz-Perez, Z., Guzman, G., Miller, T., & Potter, J. D. (2007). Fast skeletal muscle regulatory light chain is required for fast and slow skeletal muscle development. *The FASEB Journal*, *21*(9), 2205–2214. https://doi.org/10.1096/fj.06-7538com

Wassenegger, M., Heimes, S., Riedel, L., & Sanger, H. L. (1994). RNA-directed De-Novo methylation of genomic sequences in plants. *Cell*, *76*(3), 567–576. https://doi.org/10.1016/0092-8674(94)90119-8

Watrin, E., & Peters, J.-M. (2009). The cohesin complex is required for the DNA damage-induced G2/M checkpoint in mammalian cells. *The EMBO Journal*, *28*(17), 2625–2635. https://doi.org/10.1038/emboj.2009.202

Wightman, B., Ha, I., & Ruvkun, G. (1993). Posttranscriptional regulation of the heterochronic gene lin-14 by lin-4 mediates temporal pattern formation in C. elegans. *Cell*, *75*(5), 855–862. https://doi.org/10.1016/0092-8674(93)90530-4

Winbanks, C. E., Wang, B., Beyer, C., Koh, P., White, L., Kantharidis, P., & Gregorevic, P. (2011). TGF-β regulates miR-206 and miR-29 to control myogenic differentiation

through regulation of HDAC4. *Journal of Biological Chemistry*, *286*(16), 13805–13814. https://doi.org/10.1074/jbc.M110.192625

Yi, R., Qin, Y., Macara, I. G., & Cullen, B. R. (2003). Exportin-5 mediates the nuclear export of pre-microRNAs and short hairpin RNAs. *Genes and Development*, *17*(24), 3011–3016. https://doi.org/10.1101/gad.1158803

Yongmin Yan, D. W. (2013). The CRISPR-Cas9 System: A Powerful Tool for Genome Engineering and Regulation. *Advancements in Genetic Engineering*, *02*(03), 3–5. https://doi.org/10.4172/2169-0111.1000e103

Yuan, F., Zhang, Y. H., Kong, X. Y., & Cai, Y. D. (2017). Identification of Candidate Genes Related to Inflammatory Bowel Disease Using Minimum Redundancy Maximum Relevance, Incremental Feature Selection, and the Shortest-Path Approach. *BioMed Research International*, *2017*. https://doi.org/10.1155/2017/5741948

Zamore, P. D., Tuschl, T., Sharp, P. a, & Bartel, D. P. (2000). RNAi: double-stranded RNA directs the ATP-dependent cleavage of mRNA at 21 to 23 nucleotide intervals. *Cell*, *101*(1), 25–33. https://doi.org/10.1016/S0092-8674(00)80620-0

Zappia, M. P., & Frolov, M. V. (2016). E2F function in muscle growth is necessary and sufficient for viability in Drosophila. *Nat Commun*, *7:10509*, 1–16. https://doi.org/10.1038/ncomms10509

Zeng, Y., & Cullen, B. R. (2003). Sequence requirements for micro RNA processing and function in human cells. *RNA (New York, N.Y.)*, *9*(1), 112–23. https://doi.org/10.1261/rna.2780503.known

Zhang, L., Li, X., Tai, J., Li, W., & Chen, L. (2012). Predicting candidate genes based on combined network topological features: A case study in coronary artery disease. *PLoS ONE*, *7*(6). https://doi.org/10.1371/journal.pone.0039542

Zheng, S., & Zhao, Z. (2009). GenRev: exploring functional relevance of genes in molecular networks, *27*(4), 339–351. https://doi.org/10.1016/j.neuron.2009.10.017.A

Appendix 1 Application: ARP2/3 RNAi loss-of-function screening

Application: human ARP2/3 complex RNAi loss-of-function screening

Hit list: ACSM2A, AGAP2, AIM1L, AMBRA1, ANKRD33, ANO4, ANXA11, AQP11, CLMP, ATP6AP2, BAGE2, BPIFB2, C14orf180, FAM210A, URI1, GFOD1, C9, CASS4, CBX1, CCDC6, CD1A, CDK5RAP2, CFC1, CFDP1, CHAF1A, CLEC16A, CLPTM1, CNOT10, CNRIP1, COX16, CRYGA, CYP19A1, DCP2, DCST1, DDRGK1, FAM71F1, FAM193B, C7orf69, FRMD8, FZR1, GNAQ, GOLM1, HIST1H2AM, HNRNPUL1, HTR3D, JMJD7, CWC22, KLHL10, KRT1, KRTAP11-1, LIMCH1, LTB, LY6K, MAGEF1, MAGOH, MAPRE2, MGA, NATD1, MRPL19, NPAT, OR1F1, OR2L3, OR2W3, PALMD, PCDHA1, PFDN2, IPCEF1, POMC, PPP4R2, ERI3, PSG3, RAB32, RICTOR, RPL15, SGTB, SH2D4B, SLC2A7, SLC5A9, SLC9A7, SNX10, SPOP, STAG2, STMN4, STRBP, SUMO2, SYT3, SYT5, TEX2, TMC5, NDC1, TNFSF12, TOM1L1, TOMM20L, TP53BP1, TYMS, UFM1, UTP14C, VPS39, TBC1D31, WTAP, YY1, ZC3H13, ZDHHC13, ZNF226, ZNF253, ZNF614.

“Final implementers”: RB1, CDKN1A.

The analysis was performed in protein-protein network.

Figure 1. The shortest paths network for ARP2/3 complex – LIMCH1 protein

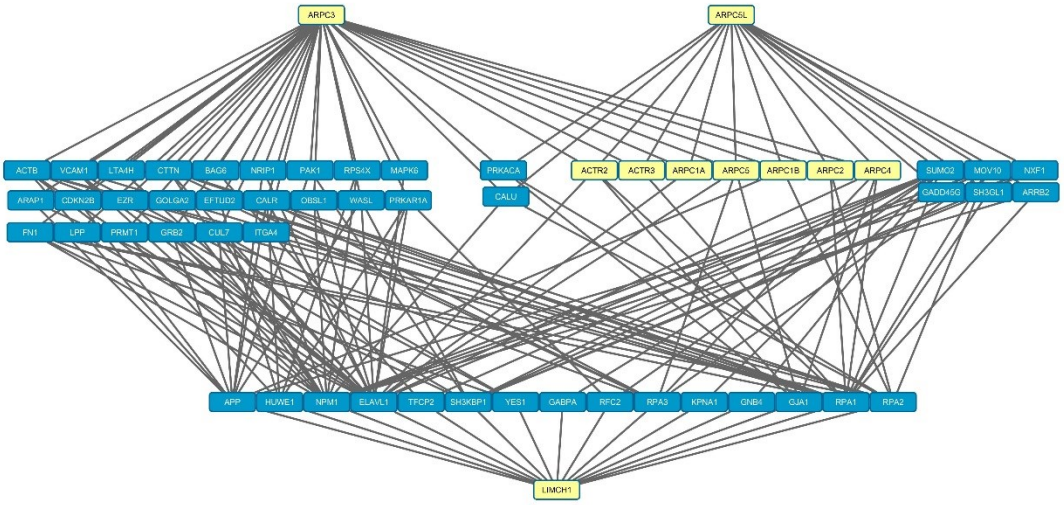


Figure 2. The shortest paths network for LIMCH1 – p21 proteins.

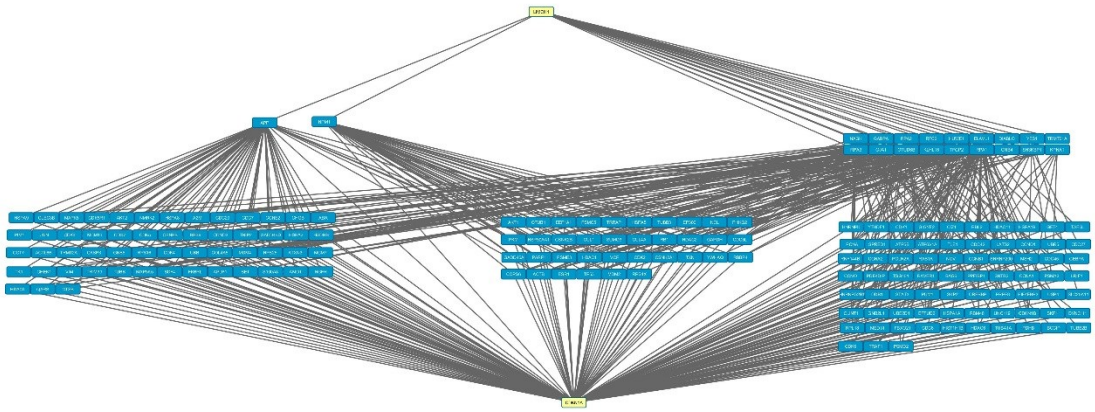


Figure 3. The shortest paths network for LIMCH1 – RB1 proteins.

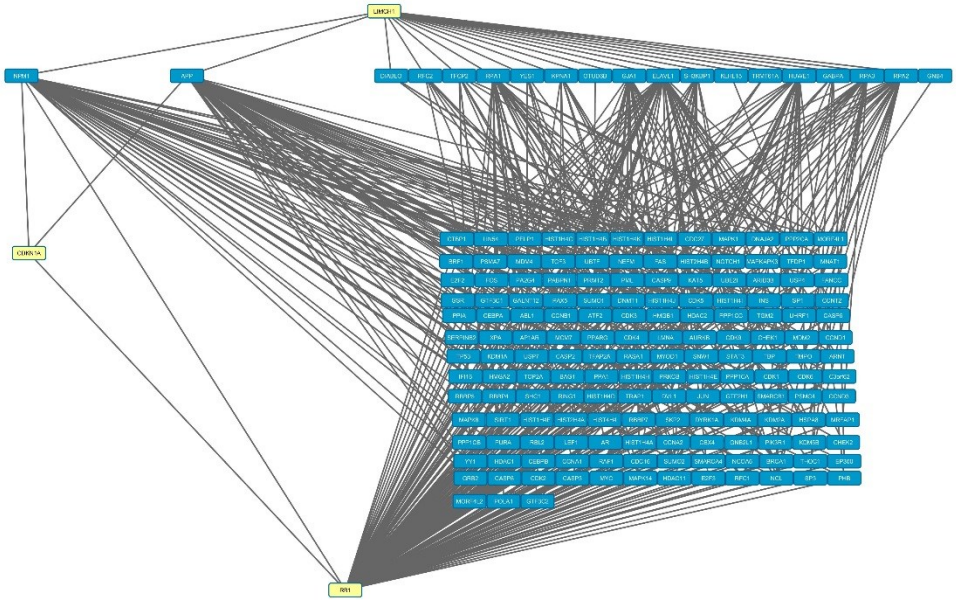


Figure 4. The shortest paths network for RAC complex – ARPIN protein.

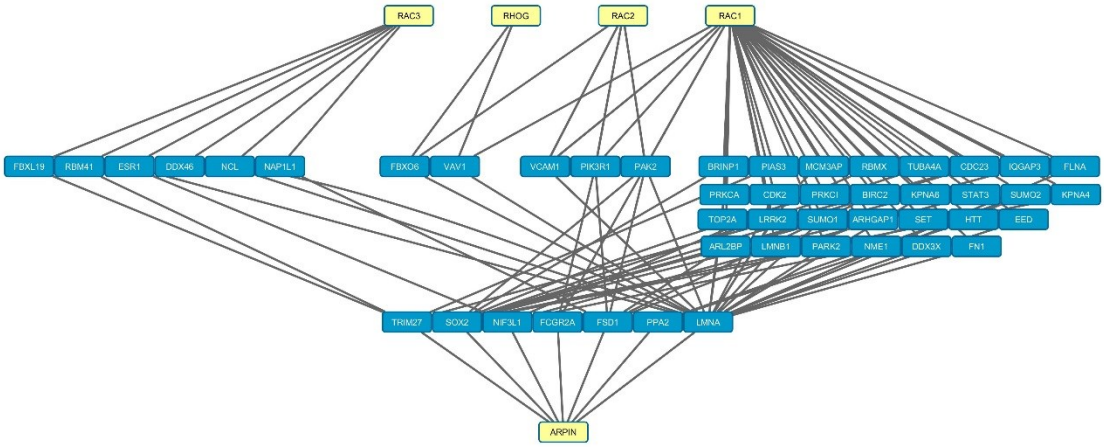


Figure 5. A path with high centrality score.

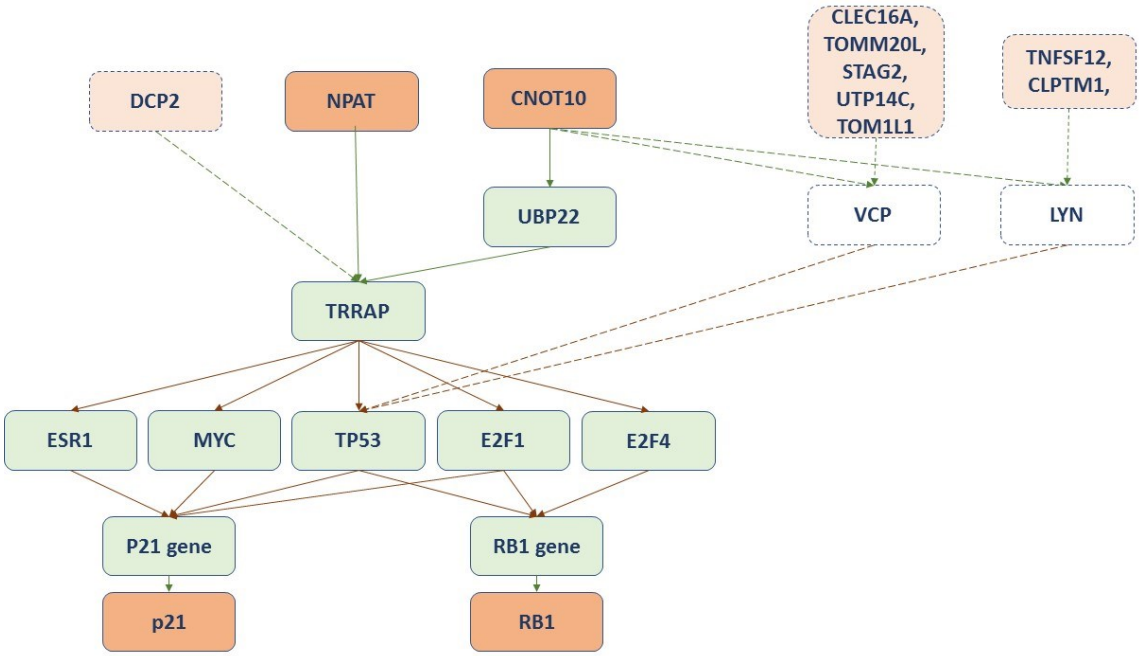


Figure 6. A path with high centrality score.

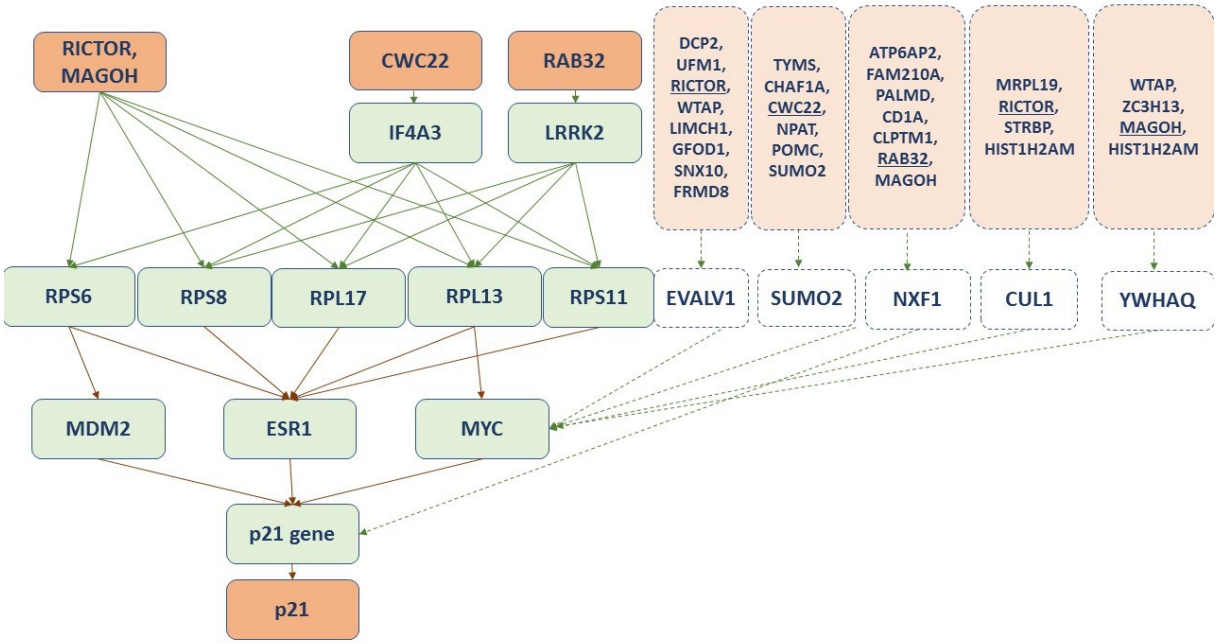


Table 1. Paths with centrality scores.

Centrality score	Path				p value
4	pMYC-pRPL13	pMYC-CDKN1A	CDKN1A-pCDKN1A		0.004599
4	pESR1-pRPL13	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.004599
4	pCDK5RAP3-pHSPA4	pTP53-pHSPA4			0.011598
4	pCDKN2A-pCDK5RAP3	pCDKN2A-pTP53			0.011598
4	pCREBBP-pCDK5RAP3	pCREBBP-pTP53			0.011598
4	pCREBBP-pCDK5RAP3	pCREBBP-pSP1			0.011598
4	pCREBBP-pCDK5RAP3	pCREBBP-pATF2			0.0012
4	pRELA-pCDK5RAP3	pRELA-pSP1			0.011598
4	pATF2-pSHMT2	pATF2-RB1	RB1-pRB1		0.046391
4	pRPL23-pTP53	pTP53-RB1	RB1-pRB1		0.036793
4	pATF2-pSHMT2	pATF2-RB1			0.046391
4	pRPL23-pTP53	pTP53-RB1			0.036793
4	pACSM2A-cpd:C00136	cpd:C00136-pHADHB	pATF2-pHADHB		0.011398
3	pMDM2-pRPS6	pMDM2-CDKN1A	CDKN1A-pCDKN1A		0.026795
3	pESR1-pRPS6	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.026795
3	pESR1-pRPS8	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.023595
3	pESR1-pRPL17	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.018796
3	pESR1-pRPS11	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.029594
3	pRELA-pCDK5RAP3	pRELA-HDAC1	HDAC1-pHDAC1		0.0024
3	pRELA-pCDK5RAP3	pRELA-CCND1	CCND1-pCCND1		0.0024
3	pRELA-pCDK5RAP3	pRELA-TP53	TP53-pTP53		0.0024
3	pACSM2A-cpd:C00136	cpd:C00136-pHADHB	pMYC-pHADHB		0.011398
3	pESR1-pILF2	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.039992
3	pESR1-pRPL15	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.0014
3	pTP53-pMAP1B	pTP53-CDKN1A	CDKN1A-pCDKN1A		0.04939
3	pTP53-pPPP4C	pTP53-RB1	RB1-pRB1		0.046791
2	pTRRAP-pTP53	pTP53-RB1	RB1-pRB1		0.04879
2	pTRRAP-pE2F1	pE2F1-RB1	RB1-pRB1		0.04879
2	pTRRAP-pE2F4	pE2F4-RB1	RB1-pRB1		0.04879
2	pTRRAP-pE2F1	pE2F1-CDKN1A	CDKN1A-pCDKN1A		0.029794
2	pTRRAP-pTP53	pTP53-CDKN1A	CDKN1A-pCDKN1A		0.029794
2	pTRRAP-pMYC	pMYC-CDKN1A	CDKN1A-pCDKN1A		0.029794
2	pTRRAP-pESR1	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.029794
2	pAKAP13-pESR1	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.044391
2	pATF2-pFUBP1	pATF2-RB1	RB1-pRB1		0.044591
2	pBRCC3-pTP53	pTP53-RB1	RB1-pRB1		0.026795
2	pBRE-pTP53	pTP53-RB1	RB1-pRB1		0.042791
2	pCDK5R1-pCTNNB1	pCTNNB1-CDKN1A	CDKN1A-pCDKN1A		0.042392
2	pESR1-pRPL21	pESR1-CDKN1A	CDKN1A-pCDKN1A		0.020996
2	pKPNA6-pNOTCH1	pNOTCH1-CDKN1A	CDKN1A-pCDKN1A		0.046791
2	pMDM2-pRYBP	pMDM2-CDKN1A	CDKN1A-pCDKN1A		0.031394
2	pNFATC2-pFOXP3	pFOXP3-RB1	RB1-pRB1		0.040392
2	pRNF14-pAR	pAR-RB1	RB1-pRB1		0.031394
2	pRYBP-pE2F3	pE2F3-CDKN1A	CDKN1A-pCDKN1A		0.031394
2	pRYBP-pTP53	pTP53-CDKN1A	CDKN1A-pCDKN1A		0.031394
2	pRYBP-pTP53	pTP53-RB1	RB1-pRB1		0.027594
2	pSTAP1-pAR	pAR-RB1	RB1-pRB1		0.045591
2	pTP53-pPPP4C	pTP53-CDKN1A	CDKN1A-pCDKN1A		0.045391
2	pTP53-pRPL7A	pTP53-RB1	RB1-pRB1		0.030994
2	pRELA-pCDK5RAP3	pRELA-MYC	MYC-pMYC	pMYC-pRB1	0.002799

Appendix 2 masterPATH JavaDoc

Below is Java Doc for masterPATH method.

masterPATH

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 masterPATH.CentralityManager Class Reference

Public Member Functions

- void **calculate_centrality_scores_for_nodes** (**Network** all, Map< String, String[ ]> hugo_by_id, **NetworkManager** nutils, String f_shortest_pathways, String output_file) throws FileNotFoundException, IOException
- void **calculate_centrality_scores_for_paths** (**Network** all, String inf, String outf, int min_length, int max↩ _length) throws FileNotFoundException, IOException
- void **calculate_centrality_scores_for_paths_ppi** (**Network** all, String inf, String outf, int min_length, int max_length) throws FileNotFoundException, IOException
- void **add_hgnc_symbols_to_paths** (String inf, String outf, **Network** all, Map< String, String[ ]> hugo_↩ by_id) throws IOException
- void **add_experimental_scores_and_pvalues_to_nodes** (String nodes_file, String gw_file, String conv_↩ table, String out) throws FileNotFoundException, IOException
- void **add_experimnetal_scores_and_aggregated_pvalues_to_paths** (String unique_over, String gw_file, String conv_table, String out) throws FileNotFoundException, IOException
- void **calculate_paths_degree** (String unique_over, String path_con, String out) throws FileNotFound↩ Exception, IOException
- Map< String, I1S1 > **find_unique_paths_for_list_of_pathways_ppi** (int max_length, int min_length, List< String > all_paths) throws IOException
- void **filter_paths** (String foundf, String outname, **Network** all, Map< String, String > hg, Map< String, String > fpl, int min_occ, String mask) throws FileNotFoundException, IOException
- HashMap< String, Float[ ]> **load_screening_data** (String p_file, String n_file) throws IOException
- HashMap< String, List< Float > > **return_permutation_distribution** (String p_file, String n_file, int number_of_permutations) throws FileNotFoundException, IOException
- Map< String, I1S1 > **find_unique_paths_for_list_of_pathways** (int max_length, int min_length, List< String > all_paths) throws IOException
- void **compare_two_lists_of_nodes** (String f1, String f2) throws FileNotFoundException, IOException
- String **returnPathType** (**Network** all, String s)

Static Public Member Functions

- static String **return_symmetric_path** (String part)

2.1.1 Detailed Description

CentralityManager (p. 3) class contains methods to calculate centrality scores

Author

Natalia Rubanova

2.1.2 Member Function Documentation

2.1.2.1 add_experimental_scores_and_pvalues_to_nodes()

```
void masterPATH.CentralityManager.add_experimental_scores_and_pvalues_to_nodes (
            String nodes_file,
            String gw_file,
            String conv_table,
            String out ) throws FileNotFoundException, IOException
```

Add scores and p-values from GW screen to file with nodes and centrality scores

nodes_file	File with nodes and centrality scores
gw_file	File with screening data
conv_table	Conversion table for genes ids
out	Output file

2.1.2.2 add_experimnetal_scores_and_aggregated_pvalues_to_paths()

```
void masterPATH.CentralityManager.add_experimnetal_scores_and_aggregated_pvalues_to_paths (
            String unique_over,
            String gw_file,
            String conv_table,
            String out ) throws FileNotFoundException, IOException
```

Add aggregated p-values from GW screen to file with paths and centrality scores

nodes_file	File with nodes and centrality scores
gw_file	File with screening data
conv_table	Conversion table for genes ids
out	Output file

2.1.2.3 add_hgnc_symbols_to_paths()

```
void masterPATH.CentralityManager.add_hgnc_symbols_to_paths (
            String inf,
            String outf,
             Network all,
            Map< String, String[]> hugo_by_id ) throws IOException
```

Add approved gene symbols to file with paths and centrality scores

inf	Input file with paths
outf	Output file
all	**Network** (p. 14) all
hugo_by_id	Map of approved gene symbols and gene IDs from **DBManager** (p. 9) class

2.1.2.4 calculate_centrality_scores_for_nodes()

```
void masterPATH.CentralityManager.calculate_centrality_scores_for_nodes (
             Network all,
            Map< String, String[]> hugo_by_id,
             NetworkManager nutils,
            String f_shortest_pathways,
            String output_file ) throws FileNotFoundException, IOException
```

Find nodes and calculate centrality scores

all	**Network** (p. 14)
hugo_by_id	Map with HGNC ids
nutils	Link to **NetworkManager** (p. 16) object
f_shortest_pathways	File with pathways
output_file	Output file

2.1.2.5 calculate_centrality_scores_for_paths()

```
void masterPATH.CentralityManager.calculate_centrality_scores_for_paths (
             Network all,
            String inf,
```

```
            String outf,
            int min_length,
            int max_length ) throws FileNotFoundException, IOException
```

Find short linear paths in a mixed direct/indirect subnetwork and calculate centrality scores

all	**Network** (p. 14)
inf	List of pathways to create subnetwork
outf	Output file
min_length	minimum length of a path
max_length	maximum length of a path

2.1.2.6 calculate_centrality_scores_for_paths_ppi()

```
void masterPATH.CentralityManager.calculate_centrality_scores_for_paths_ppi (
             Network all,
            String inf,
            String outf,
            int min_length,
            int max_length ) throws FileNotFoundException, IOException
```

Find short linear paths in a indirect subnetwork and calculate centrality scores

all	**Network** (p. 14)
inf	List of pathways to create subnetwork
outf	Output file
min_length	Minimum path length
max_length	Maximum path length

2.1.2.7 calculate_paths_degree()

```
void masterPATH.CentralityManager.calculate_paths_degree (
            String unique_over,
            String path_con,
            String out ) throws FileNotFoundException, IOException
```

Add information about degree to file with paths and centrality scores

unique_over	File with paths
path_con	File with pathway connectivity
out	Output file

2.1.2.8 compare_two_lists_of_nodes()

```
void masterPATH.CentralityManager.compare_two_lists_of_nodes (
            String f1,
            String f2 ) throws FileNotFoundException, IOException
```

Compare two lists of nodes

f1	File 1
f2	File 2

2.1.2.9 filter_paths()

```
void masterPATH.CentralityManager.filter_paths (
            String foundf,
            String outname,
             Network all,
            Map< String, String > hg,
            Map< String, String > fpl,
            int min_occ,
            String mask ) throws FileNotFoundException, IOException
```

Filter paths

foundf	File with paths
outname	Output file
all	**Network** (p. 14)
hg	Hit genes
fpl	Final Players
min_occ	Minimum occurrence filter
mask	Mask to include only paths with miRNAs

2.1.2.10 find_unique_paths_for_list_of_pathways()

```
Map<String, I1S1> masterPATH.CentralityManager.find_unique_paths_for_list_of_pathways (
            int max_length,
            int min_length,
            List< String > all_paths ) throws IOException
```

Find all unique paths in a mixed direct/indirect subnetwork

max_length	Minimum length
min_length	Maximum length
all_paths	List of pathways to create subnetwork

2.1.2.11 find_unique_paths_for_list_of_pathways_ppi()

```
Map<String, I1S1> masterPATH.CentralityManager.find_unique_paths_for_list_of_pathways_ppi (
            int max_length,
            int min_length,
            List< String > all_paths ) throws IOException
```

Find all unique paths in a indirect subnetwork

max_length	Maximum length
min_length	Minimum length
all_paths	Pathways to create subnetwork

2.1.2.12 load_screening_data()

```
HashMap<String, Float[]> masterPATH.CentralityManager.load_screening_data (
            String p_file,
            String n_file ) throws IOException
```

Load experimental data

p_file	Experimental data
n_file	File for symbol conversion

2.1.2.13 return_permutation_distribution()

```
HashMap<String, List<Float> > masterPATH.CentralityManager.return_permutation_distribution (
            String p_file,
            String n_file,
            int number_of_permutations ) throws FileNotFoundException, IOException
```

Calculate empirical distribution for Fisher's statistic

p_file	Experimental data
n_file	File for symbol conversion
number_of_permutations	Number of permutations

2.1.2.14 return_symmetric_path()

```
static String masterPATH.CentralityManager.return_symmetric_path (
             String part )  [static]
```

Return symmetric pathway

part	Pathway

2.1.2.15 returnPathType()

```
String masterPATH.CentralityManager.returnPathType (
              Network all,
             String s )
```

Check type of the pathway

all	**Network** (p. 14)
s	pathway

The documentation for this class was generated from the following file:

- CentralityManager.java

2.2 masterPATH.DBManager Class Reference

Public Member Functions

- **DBManager** () throws IOException
- void **loadHUGO** () throws FileNotFoundException, IOException
- void **loadHPRD** () throws FileNotFoundException, IOException
- void **loadHIPPIE** (boolean high) throws FileNotFoundException, IOException

- void **loadSignor** () throws FileNotFoundException, IOException
- void **loadSignalink** () throws FileNotFoundException, IOException
- void **loadtFacts** () throws FileNotFoundException, IOException
- void **loadKEGG** () throws FileNotFoundException, IOException
- void **loadTransmir** () throws FileNotFoundException, IOException
- void **loadmirTarBase** () throws FileNotFoundException, IOException

2.2.1 Detailed Description

DBManager (p. 9) class contains methods to load databases

Author

Natalia Rubanova

2.2.2 Constructor & Destructor Documentation

2.2.2.1 DBManager()

```
masterPATH.DBManager.DBManager ( ) throws IOException
```

Constructor

2.2.3 Member Function Documentation

2.2.3.1 loadHIPPIE()

```
void masterPATH.DBManager.loadHIPPIE (
            boolean high ) throws FileNotFoundException, IOException
```

Load HIPPIE database

high	load all or only high confidence interactions

2.2.3.2 loadHPRD()

```
void masterPATH.DBManager.loadHPRD ( ) throws FileNotFoundException, IOException
```

Load HPRD database

2.2.3.3 loadHUGO()

```
void masterPATH.DBManager.loadHUGO ( ) throws FileNotFoundException, IOException
```

Loads HGNC nomenclature

2.2.3.4 loadKEGG()

```
void masterPATH.DBManager.loadKEGG ( ) throws FileNotFoundException, IOException
```

Load KEGG database

2.2.3.5 loadmirTarBase()

```
void masterPATH.DBManager.loadmirTarBase ( ) throws FileNotFoundException, IOException
```

Load mitTarBase database

2.2.3.6 loadSignalink()

```
void masterPATH.DBManager.loadSignalink ( ) throws FileNotFoundException, IOException
```

Load Signalink database

2.2.3.7 loadSignor()

```
void masterPATH.DBManager.loadSignor ( ) throws FileNotFoundException, IOException
```

Load Signor database

2.2.3.8 loadtFacts()

```
void masterPATH.DBManager.loadtFacts ( ) throws FileNotFoundException, IOException
```

Load tFacts database

2.2.3.9 loadTransmir()

```
void masterPATH.DBManager.loadTransmir ( ) throws FileNotFoundException, IOException
```

Load TransMir database

The documentation for this class was generated from the following file:

- DBManager.java

2.3 masterPATH.FoldersPaths Class Reference

2.3.1 Detailed Description

FolderPaths class contains system paths to databases files, files with pathways, paths, nodes, etc

Author

Natalia Rubanova

The documentation for this class was generated from the following file:

- FoldersPaths.java

2.4 masterPATH.Interaction Class Reference

Public Member Functions

- **Interaction** (String id, List< String > other_ids, **Node** int1, **Node** int2, String type, String sourcedb, List< String > sourcedbentry, String quality, String dir)
- String **toString** ()
- void **print** ()

2.4.1 Detailed Description

Interaction (p. 12) class describes an **Interaction** (p. 12) object

Author

Natalia Rubanova

2.4.2 Constructor & Destructor Documentation

2.4.2.1 Interaction()

```
masterPATH.Interaction.Interaction (
            String id,
            List< String > other_ids,
             Node int1,
             Node int2,
            String type,
            String sourcedb,
            List< String > sourcedbentry,
            String quality,
            String dir )
```

Constructor

id	id of the interaction
other_ids	ids from the source databases
int1	first interactor
int2	second interactor
type	type of the interaction
sourcedb	source database
sourcedbentry	entry in the source database
quality	confidence parameter of the interaction
dir	direct or indirect interaction

2.4.3 Member Function Documentation

2.4.3.1 print()

```
void masterPATH.Interaction.print ( )
```

Print all information about interaction in one line

The documentation for this class was generated from the following file:

- Interaction.java

2.5 masterPATH.MasterPATH Class Reference

Static Public Member Functions

- static void **main** (String[ ] args) throws FileNotFoundException, IOException, InterruptedException

2.5.1 Detailed Description

Main class

Author

Natalia Rubanova

2.5.2 Member Function Documentation

2.5.2.1 main()

```
static void masterPATH.MasterPATH.main (
            String [] args ) throws FileNotFoundException, IOException, InterruptedException
[static]
```

main method

args	

The documentation for this class was generated from the following file:

- MasterPATH.java

2.6 masterPATH.Network Class Reference

Public Member Functions

- **Network** ()
- void **loadNetworkfromfile** (String nodef, String intf) throws FileNotFoundException, IOException
- void **saveNetworktofile** (String nodef, String intf) throws IOException
- void **removeAll** ()
- void **loadNetwork2** (**NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DB↩ Manager** dbutils) throws IOException

2.6.1 Detailed Description

Network (p. 14) class describes an **Network** (p. 14) object

Author

Natalia Rubanova

2.6.2 Constructor & Destructor Documentation

2.6.2.1 Network()

```
masterPATH.Network.Network ( )
```

Constructor

2.6.3 Member Function Documentation

2.6.3.1 loadNetwork2()

```
void masterPATH.Network.loadNetwork2 (
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Load network from databases

nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.6.3.2 loadNetworkfromfile()

```
void masterPATH.Network.loadNetworkfromfile (
            String nodef,
            String intf ) throws FileNotFoundException, IOException
```

Load network from two files ∗

nodef	file with information about nodes
intf	file with information about interactions

2.6.3.3 removeAll()

```
void masterPATH.Network.removeAll ( )
```

Remove all interactions and nodes from network

2.6.3.4 saveNetworktofile()

```
void masterPATH.Network.saveNetworktofile (
            String nodef,
            String intf ) throws IOException
```

Save all interactions and nodes in the network to the two output files

nodef	node file
intf	interaction file

The documentation for this class was generated from the following file:

- Network.java

2.7 masterPATH.NetworkManager Class Reference

Classes

- class **Depth_List_of_PathUnit**
- class **PathUnit**
- class **PathUnitwFP**

Public Member Functions

- **Network merge_list_of_networks** (List< **Network** > n)
- void **build_map_of_neighbors** (**Network** nw)
- void **load_hitlist_and_finalimpl** (String hgf, String fpf, Map< String, String[ ]> hugo) throws FileNotFound↩ Exception, IOException
- List< PathUnit > **find_pathway_BF_algorithm** (**Network** network, List< String > n1, List< String > n2, int max, String prefix, FileWriter out) throws IOException
- List< PathUnit > **find_pathway_BF_algorithm_ppi** (**Network** network, List< String > n1, List< String > n2, int max, String prefix, BufferedWriter out) throws IOException
- void **find_pathway_for_list_BF_algorithm** (**Network** nw, Map< String, String[ ]> hugo, int max, String outf, String prefix) throws IOException
- void **find_pathway_for_list_BF_algorithm_ppi** (**Network** nw, Map< String, String[ ]> hugo, int max, String outf, String prefix) throws IOException
- void **add_missing_genes_and_products** (**Network** nw)

2.7.1 Detailed Description

NetworkManager (p. 16) class contains methods to deal with networks

Author

Natalia Rubanova

2.7.2 Member Function Documentation

2.7.2.1 add_missing_genes_and_products()

```
void masterPATH.NetworkManager.add_missing_genes_and_products (
              Network nw )
```

Add transcription information

nw	**Network** (p. 14)

2.7.2.2 build_map_of_neighbors()

```
void masterPATH.NetworkManager.build_map_of_neighbors (
              Network nw )
```

Create links to all neighbors for every node in the network

nw	**Network** (p. 14)

2.7.2.3 find_pathway_BF_algorithm()

```
List<PathUnit> masterPATH.NetworkManager.find_pathway_BF_algorithm (
              Network network,
             List< String > n1,
             List< String > n2,
             int max,
             String prefix,
             FileWriter out ) throws IOException
```

Find length-bound pathways between two lists of nodes

network	
n1	List of Hit genes nodes
n2	List of Final implementers nodes
max	Maximum length
prefix	Prefix for interaction id
out	Output file

tmp_visited_nodes.containsKey(n)

2.7.2.4 find_pathway_BF_algorithm_ppi()

```
List<PathUnit> masterPATH.NetworkManager.find_pathway_BF_algorithm_ppi (
              Network network,
             List< String > n1,
             List< String > n2,
             int max,
             String prefix,
             BufferedWriter out ) throws IOException
```

Find a pathway between two lists of nodes in a PPI network

network	

n1	List of Hit genes nodes
n2	List of Final implementers nodes
max	Maximum length
prefix	Prefix for interaction id
out	output file

Arrays.equals(visited_nodes.get(n), visited_nodes.get(curn.current_seed)) || || path_type >= 1

Arrays.equals(visited_nodes.get(n), visited_nodes.get(curn.current_seed)) || || path_type >= 1

tmp_visited_nodes.containsKey(n)

2.7.2.5 find_pathway_for_list_BF_algorithm()

```
void masterPATH.NetworkManager.find_pathway_for_list_BF_algorithm (
              Network nw,
             Map< String, String[]> hugo,
             int max,
             String outf,
             String prefix ) throws IOException
```

Wrapper (p. 38) for finding pathways

nw	**Network** (p. 14)
hugo	Link to HGNC nomenclature
max	Maximum length
outf	Output file
prefix	Prefix for interaction id

2.7.2.6 find_pathway_for_list_BF_algorithm_ppi()

```
void masterPATH.NetworkManager.find_pathway_for_list_BF_algorithm_ppi (
              Network nw,
             Map< String, String[]> hugo,
             int max,
             String outf,
             String prefix ) throws IOException
```

Wrapper (p. 38) for finding pathways in a PPI network

nw	**Network** (p. 14)
hugo	Link to HGNC nomenclature

max	Maximum length
outf	Output file
prefix	Prefix for interaction id

2.7.2.7 load_hitlist_and_finalimpl()

```
void masterPATH.NetworkManager.load_hitlist_and_finalimpl (
            String hgf,
            String fpf,
            Map< String, String[]> hugo ) throws FileNotFoundException, IOException
```

Load hit genes and final implementers files

hgf	Hit genes file
fpf	Final players file
hugo	Link to the HGNC nomenclature

2.7.2.8 merge_list_of_networks()

```
Network masterPATH.NetworkManager.merge_list_of_networks (
            List<  Network > n )
```

Merge list of networks

n	List of networks

The documentation for this class was generated from the following file:

- NetworkManager.java

2.8 masterPATH.NetworkTopology Class Reference

Public Member Functions

- void **calculate_average_clustering_coefficient_ppi** (**Network** all)
- void **calculate_average_clustering_coefficient_direct** (**Network** all)

- void **calculate_number_of_connected_components_ppi** (**Network** all)
- void **calculate_number_of_connected_components_direct** (**Network** all)
- void **calculate_diameter_ppi** (**Network** all)
- void **calculate_diameter_direct** (**Network** all)
- void **get_degree_distribution_ppi** (**Network** all, String outfile) throws IOException
- void **get_degree_distribution_direct** (**Network** all, String outfile) throws IOException
- void **get_subnetwork_statistics** (**Network** all, String fname) throws FileNotFoundException, IOException

2.8.1 Detailed Description

NetworkTopology (p. 19) class contains methods to calculate network's topological properties

Author

Natalia Rubanova

2.8.2 Member Function Documentation

2.8.2.1 calculate_average_clustering_coefficient_direct()

```
void masterPATH.NetworkTopology.calculate_average_clustering_coefficient_direct (
              Network all )
```

Calculate average clustering coefficient for direct network

all	**Network** (p. 14)

2.8.2.2 calculate_average_clustering_coefficient_ppi()

```
void masterPATH.NetworkTopology.calculate_average_clustering_coefficient_ppi (
              Network all )
```

Calculate average clustering coefficient for PPI network

all	**Network** (p. 14)

2.8.2.3 calculate_diameter_direct()

```
void masterPATH.NetworkTopology.calculate_diameter_direct (
            Network all )
```

Calculate diameter of a direct network

all	**Network** (p. 14)

2.8.2.4 calculate_diameter_ppi()

```
void masterPATH.NetworkTopology.calculate_diameter_ppi (
            Network all )
```

Calculate diameter

all	**Network** (p. 14)

2.8.2.5 calculate_number_of_connected_components_direct()

```
void masterPATH.NetworkTopology.calculate_number_of_connected_components_direct (
            Network all )
```

Calculate number of connected components for direct network

all	**Network** (p. 14)

2.8.2.6 calculate_number_of_connected_components_ppi()

```
void masterPATH.NetworkTopology.calculate_number_of_connected_components_ppi (
            Network all )
```

Calculate number of connected components PPI network

all	**Network** (p. 14)

2.8.2.7 get_degree_distribution_direct()

```
void masterPATH.NetworkTopology.get_degree_distribution_direct (
              Network all,
             String outfile ) throws IOException
```

Build degree distribution for direct network

all	**Network** (p. 14)
outfile	Output file

2.8.2.8 get_degree_distribution_ppi()

```
void masterPATH.NetworkTopology.get_degree_distribution_ppi (
              Network all,
             String outfile ) throws IOException
```

Build degree distribution for PPI network

all	**Network** (p. 14)
outfile	Output file

2.8.2.9 get_subnetwork_statistics()

```
void masterPATH.NetworkTopology.get_subnetwork_statistics (
              Network all,
             String fname ) throws FileNotFoundException, IOException
```

Calculate subnetwork properties

all	**Network** (p. 14)
fname	File with pathways to create subnetwork

The documentation for this class was generated from the following file:

- NetworkTopology.java

2.9 masterPATH.Node Class Reference

Public Member Functions

- **Node** (String id, String type, String id_type, String db_flag, List< String[ ]> ids)
- String **toString** ()
- void **print** ()

2.9.1 Detailed Description

Node (p. 23) class describes a **Node** (p. 23) object

Author

Natalia Rubanova

2.9.2 Constructor & Destructor Documentation

2.9.2.1 Node()

```
masterPATH.Node.Node (
            String id,
            String type,
            String id_type,
            String db_flag,
            List< String[]> ids )
```

Constructor

id	id of the **Node** (p. 23)
type	type of the **Node** (p. 23)
id_type	type of the id(nomenclature)
db_flag	flag of the source database
ids	ids from other nomenclatures

2.9.3 Member Function Documentation

2.9.3.1 print()

```
void masterPATH.Node.print ( )
```

Print node information in one line

The documentation for this class was generated from the following file:

- Node.java

2.10 masterPATH.PathwayManager Class Reference

Public Member Functions

- void **find_the_shortest_paths** (String foundf, String outname, **Network** all, Map< String, String > hg, Map< String, String > fpl, int d_ppi, int d_tf, int d_mirna, int d_kegg) throws FileNotFoundException, IO↩ Exception
- void **find_the_strongest_pathways** (String foundf, String outfile) throws IOException
- void **find_miRNAs_on_pathways** (String foundf, String resf, **Network** mirtarbase, Map< String, String > hg, Map< String, String > fpl, int length, int min, String mask) throws FileNotFoundException, IOException
- void **find_miRNAs_on_paths** (String foundf, String resf, **Network** mirtarbase, **Network** transmir, Map< String, String > hg, Map< String, String > fpl, int length, int min_occ, String mask) throws FileNotFound↩ Exception, IOException
- void **create_cyto_for_paths** (String foundf, **Network** all, Map< String, String > hg, Map< String, String > fpl) throws FileNotFoundException, IOException
- void **create_cyto_for_pathways** (String path_file, String filename, String out_dir, **Network** all, Map< String, String > hg, Map< String, String > fpl) throws FileNotFoundException, IOException
- void **filterPathways_by_length** (String inf, String outf, int min, int max) throws IOException
- void **filterPathways_by_centrality_score** (String inf, String outf, int min, int max) throws IOException
- void **filterPathways** (String foundf, String outname, **Network** all, Map< String, String > hg, Map< String, String > fpl, String mask, String gene, String type) throws FileNotFoundException, IOException
- void **filterPathways_2** (String foundf, String outname, **Network** all, Map< String, String > hg, Map< String, String > fpl, String gene, int l1, int l2, int l3, int l4) throws FileNotFoundException, IOException
- void **add_connectivity_to_pathways** (**Network** all, String infile, String outfile) throws IOException
- void **filter_pathways_by_connectivity** (String in, String out, String out_for_overrreps, int in_min, int in_max, int out_min, int out_max) throws FileNotFoundException, IOException
- void **compare_two_paths_files** (**Network** all, Map< String, String[ ]> hugo_by_id, String f1, String f2, String out) throws FileNotFoundException, IOException
- void **compare_two_nodes_files** (**Network** all, Map< String, String[ ]> hugo_by_id, String f1, String f2, String out) throws FileNotFoundException, IOException
- void **find_hitgenes_on_paths** (**Network** all, **NetworkManager** nutils, Map< String, String[ ]> hugo_by_id, String f1, String out) throws FileNotFoundException, IOException
- void **add_aggregated_pvalues_to_pathways** (**Network** all, String f_pathways, String gw_file, String conv_table, String out, Map< String, String[ ]> hugo_by_id, **CentralityManager** centralityManager) throws IOException

2.10.1 Detailed Description

PathwayManager (p. 24) class contains methods to deal with pathways

Author

Natalia Rubanova

2.10.2 Member Function Documentation

2.10.2.1 add_aggregated_pvalues_to_pathways()

```
void masterPATH.PathwayManager.add_aggregated_pvalues_to_pathways (
              Network all,
             String f_pathways,
             String gw_file,
             String conv_table,
             String out,
             Map< String, String[]> hugo_by_id,
              CentralityManager centralityManager ) throws IOException
```

all	the value of all
f_pathways	the value of f_pathways
gw_file	the value of gw_file
conv_table	the value of conv_table
out	the value of out
hugo_by_id	the value of hugo_by_id
centralityManager	the value of centralityManager

Exceptions

IOException	

2.10.2.2 add_connectivity_to_pathways()

```
void masterPATH.PathwayManager.add_connectivity_to_pathways (
              Network all,
             String infile,
             String outfile ) throws IOException
```

Calculate pathways connectivity

all	**Network** (p. 14)
infile	Input file
outfile	Output file

2.10.2.3 compare_two_nodes_files()

```
void masterPATH.PathwayManager.compare_two_nodes_files (
              Network all,
             Map< String, String[]> hugo_by_id,
             String f1,
             String f2,
             String out ) throws FileNotFoundException, IOException
```

Compare two files with nodes

all	**Network** (p. 14)
hugo_by↩ _id	HGNC ids map
f1	File 1
f2	File 2
out	Output file

2.10.2.4 compare_two_paths_files()

```
void masterPATH.PathwayManager.compare_two_paths_files (
              Network all,
             Map< String, String[]> hugo_by_id,
             String f1,
             String f2,
             String out ) throws FileNotFoundException, IOException
```

Compare two files with paths

all	**Network** (p. 14)
hugo_by↩ _id	Map with HGNC ids
f1	File1
f2	File 2
out	OUtput file

2.10.2.5 create_cyto_for_paths()

```
void masterPATH.PathwayManager.create_cyto_for_paths (
            String foundf,
             Network all,
            Map< String, String > hg,
            Map< String, String > fpl ) throws FileNotFoundException, IOException
```

Create a file for Cytoscape software from the shortest paths

foundf	File with paths
all	**Network** (p. 14)
hg	Hit genes list
fpl	Final players list

Exceptions

FileNotFoundException	
IOException	

2.10.2.6 create_cyto_for_pathways()

```
void masterPATH.PathwayManager.create_cyto_for_pathways (
            String path_file,
            String filename,
            String out_dir,
             Network all,
            Map< String, String > hg,
            Map< String, String > fpl ) throws FileNotFoundException, IOException
```

Creates a file for Cytoscape software for a list of pathways by pathways ids

path_file	File with pathways
filename	File with list of pathways by id
out_dir	Output folder
all	**Network** (p. 14)
hg	Hit genes list
fpl	Final players list

2.10.2.7 filter_pathways_by_connectivity()

```
void masterPATH.PathwayManager.filter_pathways_by_connectivity (
             String in,
             String out,
             String out_for_overrreps,
             int in_min,
             int in_max,
             int out_min,
             int out_max ) throws FileNotFoundException, IOException
```

Filter pathways by connectivity

in	Input file
out	Output file
out_for_overrreps	Output file only for pathway without additional information
in_min	Minimum inward connectivity
in_max	Maximum inward connectivity
out_min	Minimum outward connectivity
out_max	Maximum outward connectivity

Exceptions

FileNotFoundException	
IOException	

2.10.2.8 filterPathways()

```
void masterPATH.PathwayManager.filterPathways (
             String foundf,
             String outname,
              Network all,
             Map< String, String > hg,
             Map< String, String > fpl,
             String mask,
             String gene,
             String type ) throws FileNotFoundException, IOException
```

Filter pathways by node names \ type

foundf	Input file
outname	Output file
all	**Network** (p. 14)
hg	Hit genes list
fpl	Final players list
mask	Mask
gene	**Node** (p. 23) name
type	**Node** (p. 23) type

2.10.2.9 filterPathways_2()

```
void masterPATH.PathwayManager.filterPathways_2 (
            String foundf,
            String outname,
             Network all,
            Map< String, String > hg,
            Map< String, String > fpl,
            String gene,
            int l1,
            int l2,
            int l3,
            int l4 ) throws FileNotFoundException, IOException
```

Filter paths on condition ppi [3,4] & miRNA [6/5] or ppi [3,4] or miRNA [6/5]

foundf	File with pathways
outname	Output file
all	**Network** (p. 14)
hg	Hit genes
fpl	Final implementers
gene	Gene name
l1	min length for ppi path
l2	max length for ppi path
l3	min length for miRNA path
l4	max length for miRNA path

2.10.2.10 filterPathways_by_centrality_score()

```
void masterPATH.PathwayManager.filterPathways_by_centrality_score (
            String inf,
            String outf,
            int min,
            int max ) throws IOException
```

Filter pathways by length

inf	Input file
outf	Output file
min	Minimum length
max	Maximum length

2.10.2.11 filterPathways_by_length()

```
void masterPATH.PathwayManager.filterPathways_by_length (
            String inf,
            String outf,
            int min,
            int max ) throws IOException
```

Filter pathways by length

inf	Input file
outf	Output file
min	Minimum length
max	Maximum length

2.10.2.12 find_hitgenes_on_paths()

```
void masterPATH.PathwayManager.find_hitgenes_on_paths (
             Network all,
             NetworkManager nutils,
            Map< String, String[]> hugo_by_id,
            String f1,
            String out ) throws FileNotFoundException, IOException
```

Compare a file with paths with a hit list

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
hugo_by↩ _id	HGNC ids map
f1	File with paths
out	OUtput file

2.10.2.13 find_miRNAs_on_paths()

```
void masterPATH.PathwayManager.find_miRNAs_on_paths (
            String foundf,
            String resf,
```

```
 Network mirtarbase,
 Network transmir,
Map< String, String > hg,
Map< String, String > fpl,
int length,
int min_occ,
String mask ) throws FileNotFoundException, IOException
```

Find and rank miRNAs inside paths

foundf	Common paths file
resf	Output file
mirtarbase	Link to mirTarBAse network
transmir	Link to transMir network
hg	Hit genes list
fpl	Final players List
length	Maximum length
min_occ	Minimum occurrence
mask	Mask

2.10.2.14 find_miRNAs_on_pathways()

```
void masterPATH.PathwayManager.find_miRNAs_on_pathways (
            String foundf,
            String resf,
             Network mirtarbase,
            Map< String, String > hg,
            Map< String, String > fpl,
            int length,
            int min,
            String mask ) throws FileNotFoundException, IOException
```

Find and rank miRNAs inside pathways

foundf	File with common paths
resf	Output file
mirtarbase	Link to mirTarBase network
hg	Hit genes list
fpl	Final implementer list
length	Maximum length
min	
mask	

2.10.2.15 find_the_shortest_paths()

```
void masterPATH.PathwayManager.find_the_shortest_paths (
            String foundf,
            String outname,
             Network all,
            Map< String, String > hg,
            Map< String, String > fpl,
            int d_ppi,
            int d_tf,
            int d_mirna,
            int d_kegg ) throws FileNotFoundException, IOException
```

Find the shortest pathways

foundf	Input file
outname	Output file
all	**Network** (p. 14)
hg	Hit genes list
fpl	Final players list
d_ppi	Length gap for protein-protein interactions
d_tf	Length gap for transcriptional interactions
d_mirna	Length gap for miRNA-mRNA interactions
d_kegg	Length gap for metabolic interactions

2.10.2.16 find_the_strongest_pathways()

```
void masterPATH.PathwayManager.find_the_strongest_pathways (
            String foundf,
            String outfile ) throws IOException
```

Find the strongest pathways

foundf	Input file
outname	Output file

The documentation for this class was generated from the following file:

- PathwayManager.java

2.11 masterPATH.RandomManager Class Reference

Public Member Functions

- void **permute_phenotype_label** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, String folder, String prefix, int count, List< String > hit↩ _list, Map< String, String[ ]> hugo_by_id, String fplayers) throws IOException
- void **calculate_p_values_phenotype_label_permutation** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, String random_paths_file, String ovrreps_paths_file, String hitlists_file, int hitlist_length, int position_of_overreps_count, Map< String, String[ ]> hugo_by_id) throws IOException
- void **calculate_p_values_phenotype_label_permutation_nodes** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, String random_paths_file, String ovrreps_paths_file, String hitlists_file, int hitlist_length, Map< String, String[ ]> hugo_by_id) throws IO↩ Exception
- void **calculate_p_values_paths_random_networks** (**Network** all, **NetworkManager** nutils, **Pathway**↩ **Manager** putils, **CentralityManager** outils, **DBManager** dbutils, String random_paths_file, String ovrreps↩ _paths_file, String hitlists_file, int hitlist_length, int position_of_overreps_count, Map< String, String[ ]> hugo_by_id) throws IOException
- void **calculate_p_values_phenotype_label_permutation_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, String random_paths_file, String ovrreps_paths_file, String hitlists_file, int hitlist_length, int position_of_overreps_count, Map< String, String[ ]> hugo_by_id) throws IOException
- **Network create_random_network** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils)
- void **calculate_p_values_nodes_random_networks** (**Network** all, **NetworkManager** nutils, **Pathway**↩ **Manager** putils, **CentralityManager** outils, **DBManager** dbutils, String file_name_random, int count, String hubs_file, String overreps_file) throws IOException
- void **calculate_p_values_nodes_random_networks_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, String file_name_random, int count, String hubs_file, String overreps_file) throws IOException

2.11.1 Detailed Description

RandomManager (p. 32) contains methods to calculate p-values

Author

Natalia Rubanova

2.11.2 Member Function Documentation

2.11.2.1 calculate_p_values_nodes_random_networks()

```
void masterPATH.RandomManager.calculate_p_values_nodes_random_networks (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
             String file_name_random,
             int count,
             String hubs_file,
             String overreps_file ) throws IOException
```

Calculate p-values for nodes on random network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object
file_name_random	Name for file to store information about pathways built on the random network
count	Number of shuffling steps
hubs_file	File with nodes and centrality scores
overreps_file	File with paths and centrality scores

2.11.2.2 calculate_p_values_nodes_random_networks_ppi()

```
void masterPATH.RandomManager.calculate_p_values_nodes_random_networks_ppi (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
             String file_name_random,
             int count,
             String hubs_file,
             String overreps_file ) throws IOException
```

Calculate p-values on for nodes random ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object
file_name_random	File name to store information about pathways built on the random network
count	Number of shuffling steps
hubs_file	File with nodes and centrality scores
overreps_file	File with paths and centrality scores

2.11.2.3 calculate_p_values_paths_random_networks()

```
void masterPATH.RandomManager.calculate_p_values_paths_random_networks (
              Network all,
              NetworkManager nutils,
```

```
                PathwayManager putils,
                CentralityManager outils,
                DBManager dbutils,
               String random_paths_file,
               String ovrreps_paths_file,
               String hitlists_file,
               int hitlist_length,
               int position_of_overreps_count,
               Map< String, String[]> hugo_by_id ) throws IOException
```

all	
nutils	
putils	
outils	
dbutils	
random_paths_file	
ovrreps_paths_file	
hitlists_file	
hitlist_length	
position_of_overreps_count	
hugo_by_id	

Exceptions

IOException	

2.11.2.4 calculate_p_values_phenotype_label_permutation()

```
void masterPATH.RandomManager.calculate_p_values_phenotype_label_permutation (
                Network all,
                NetworkManager nutils,
                PathwayManager putils,
                CentralityManager outils,
                DBManager dbutils,
               String random_paths_file,
               String ovrreps_paths_file,
               String hitlists_file,
               int hitlist_length,
               int position_of_overreps_count,
               Map< String, String[]> hugo_by_id ) throws IOException
```

Calculate p-values for paths for shuffled phenotype label

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object

outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object
random_paths_file	File name for pathways from shuffled hit genes
ovrreps_paths_file	File with paths
hitlists_file	File with hit genes
hitlist_length	Length of hit list
position_of_overreps_count	Position of the centrality count in the paths file
hugo_by_id	HGNC ids map

2.11.2.5 calculate_p_values_phenotype_label_permutation_nodes()

```
void masterPATH.RandomManager.calculate_p_values_phenotype_label_permutation_nodes (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
             String random_paths_file,
             String ovrreps_paths_file,
             String hitlists_file,
             int hitlist_length,
             Map< String, String[]> hugo_by_id ) throws IOException
```

Calculate p-values for nodes for shuffled phenotype label

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object
random_paths_file	File name for pathways from shuffled hit genes
ovrreps_paths_file	File with nodes
hitlists_file	File with hit genes
hitlist_length	Length of hit list
hugo_by_id	HGNC ids map

2.11.2.6 calculate_p_values_phenotype_label_permutation_ppi()

```
void masterPATH.RandomManager.calculate_p_values_phenotype_label_permutation_ppi (
              Network all,
```

```
                NetworkManager nutils,
                PathwayManager putils,
                CentralityManager outils,
                DBManager dbutils,
               String random_paths_file,
               String ovrreps_paths_file,
               String hitlists_file,
               int hitlist_length,
               int position_of_overreps_count,
               Map< String, String[]> hugo_by_id ) throws IOException
```

Calculate p-values for paths for shuffled phenotype label for ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object
random_paths_file	File name for pathways from shuffled hit genes
ovrreps_paths_file	File with paths
hitlists_file	File with hit genes
hitlist_length	Length of hit list
position_of_overreps_count	Position of the centrality count in the paths file
hugo_by_id	HGNC ids map

2.11.2.7 create_random_network()

```
Network masterPATH.RandomManager.create_random_network (
                Network all,
                NetworkManager nutils,
                PathwayManager putils,
                CentralityManager outils,
                DBManager dbutils )
```

Create random network with the same degree distribution

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.11.2.8 permute_phenotype_label()

```
void masterPATH.RandomManager.permute_phenotype_label (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
             String folder,
             String prefix,
             int count,
             List< String > hit_list,
             Map< String, String[]> hugo_by_id,
             String fplayers ) throws IOException
```

Permute phenotype label

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object
folder	Folder to store information about pathways
prefix	Prefix for pathways ids
count	Number of shuffling steps
hit_list	Hit List
hugo_by_id	HGNC ids map
fplayers	Final implementers

The documentation for this class was generated from the following file:

- RandomManager.java

2.12 masterPATH.Wrapper Class Reference

Public Member Functions

- **Network ogg1_add_connections** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, Map< String, String[]> hugo) throws IOException
- void **ogg1_part1_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **ogg1_part2_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **ogg1_random_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException
- void **ogg1_part4_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException

- void **ogg1_random2_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **Centrality↩ Manager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException
- void **ogg1_part1** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **ogg1_part2** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **ogg1_part3** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **ogg1_part4** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **ogg1_random** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException
- void **arp_part1_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **arp_part2_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **arp_random_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **Centrality↩ Manager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException
- **Network arp_add_connections** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, Map< String, String[ ]> hugo, String hitlist) throws IO↩ Exception
- void **arp_part1_LIMCH1** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **Centrality↩ Manager** outils, **DBManager** dbutils) throws IOException
- void **arp_part1** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **arp_part2** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **arp_random** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException
- void **arp_part3** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **arp_filter** (**Network** all) throws FileNotFoundException, IOException
- void **adenocarcinoma_part1** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **adenocarcinoma_part2** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **adenocarcinoma_random** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException
- void **adenocarcinoma_part1_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **adenocarcinoma_part2_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **adenocarcinoma_random_ppi** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException
- void **mirna63Sys_part1** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **Centrality↩ Manager** outils, **DBManager** dbutils) throws IOException
- void **mirna63Sys_part2** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **Centrality↩ Manager** outils, **DBManager** dbutils) throws IOException
- void **mirna63Sys_random** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **Centrality↩ Manager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException
- void **mirna63Sys_part3** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **Centrality↩ Manager** outils, **DBManager** dbutils) throws IOException
- void **case2_part1** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **case2_part2** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException

- void **case2_part3** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **case2_part4** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- void **case2_random** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException
- void **case2_random2** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils, **RandomManager** rand) throws IOException

Static Public Member Functions

- static void **mirna63Sys_part3_3** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- static void **mirna63Sys_part4** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **CentralityManager** outils, **DBManager** dbutils) throws IOException
- static void **case2_part2_2** (**Network** all, **NetworkManager** nutils, **PathwayManager** putils, **Centrality↩ Manager** outils, **DBManager** dbutils) throws IOException

2.12.1 Detailed Description

Wrapper (p. 38) class contains wrappers implementing the whole pipeline

Author

Natalia Rubanova

2.12.2 Member Function Documentation

2.12.2.1 adenocarcinoma_part1()

```
void masterPATH.Wrapper.adenocarcinoma_part1 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 1 of pipeline for adenocarcinoma screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.2 adenocarcinoma_part1_ppi()

```
void masterPATH.Wrapper.adenocarcinoma_part1_ppi (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 1 of pipeline for adenocarcinoma screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.3 adenocarcinoma_part2()

```
void masterPATH.Wrapper.adenocarcinoma_part2 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 2 of pipeline for adenocarcinoma screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.4 adenocarcinoma_part2_ppi()

```
void masterPATH.Wrapper.adenocarcinoma_part2_ppi (
              Network all,
```

```
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 2 of pipeline for adenocarcinoma screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.5 adenocarcinoma_random()

```
void masterPATH.Wrapper.adenocarcinoma_random (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'p-values' of pipeline for adenocarcinoma screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.6 adenocarcinoma_random_ppi()

```
void masterPATH.Wrapper.adenocarcinoma_random_ppi (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'p-values' of pipeline for adenocarcinoma screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.7 arp_add_connections()

```
Network masterPATH.Wrapper.arp_add_connections (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
             Map< String, String[]> hugo,
             String hitlist ) throws IOException
```

Add low confident interactions for LIMCH1 protein

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.8 arp_filter()

```
void masterPATH.Wrapper.arp_filter (
              Network all ) throws FileNotFoundException, IOException
```

Filter pathways for arp2/3 screen in mixed network

all	**Network** (p. 14)

2.12.2.9 arp_part1()

```
void masterPATH.Wrapper.arp_part1 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 1 of pipeline for arp2/3 screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.10 arp_part1_LIMCH1()

```
void masterPATH.Wrapper.arp_part1_LIMCH1 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 1 of pipeline for arp2/3 screen in mixed network Hit list only LMCH1 gene

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.11 arp_part1_ppi()

```
void masterPATH.Wrapper.arp_part1_ppi (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 1 of pipeline for arp2/3 screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.12 arp_part2()

```
void masterPATH.Wrapper.arp_part2 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 2 of pipeline for arp2/3 screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.13 arp_part2_ppi()

```
void masterPATH.Wrapper.arp_part2_ppi (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 2 of pipeline for arp2/3 screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.14 arp_part3()

```
void masterPATH.Wrapper.arp_part3 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 3 of pipeline for arp2/3 screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.15 arp_random()

```
void masterPATH.Wrapper.arp_random (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'p-values' of pipeline for arp2/3 screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.16 arp_random_ppi()

```
void masterPATH.Wrapper.arp_random_ppi (
              Network all,
```

```
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'p-values' of pipeline for arp2/3 screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.17 case2_part1()

```
void masterPATH.Wrapper.case2_part1 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 1 of pipeline for transcriptome profiling muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.18 case2_part2()

```
void masterPATH.Wrapper.case2_part2 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 2 of pipeline for transcriptome profiling muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.19 case2_part2_2()

```
static void masterPATH.Wrapper.case2_part2_2 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException  [static]
```

Part 2 + filter of pipeline for transcriptome profiling muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.20 case2_part3()

```
void masterPATH.Wrapper.case2_part3 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 3 of pipeline for transcriptome profiling muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.21 case2_part4()

```
void masterPATH.Wrapper.case2_part4 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 4 of pipeline for transcriptome profiling muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.22 case2_random()

```
void masterPATH.Wrapper.case2_random (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'p-values' of pipeline for transcriptome profiling muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.23 case2_random2()

```
void masterPATH.Wrapper.case2_random2 (
              Network all,
```

```
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'p-values on random networks' of pipeline for transcriptome profiling muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.24 mirna63Sys_part1()

```
void masterPATH.Wrapper.mirna63Sys_part1 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 1 of pipeline for miRNA muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.25 mirna63Sys_part2()

```
void masterPATH.Wrapper.mirna63Sys_part2 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 2 of pipeline for miRNA muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.26 mirna63Sys_part3()

```
void masterPATH.Wrapper.mirna63Sys_part3 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 3 of pipeline for miRNA muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.27 mirna63Sys_part3_3()

```
static void masterPATH.Wrapper.mirna63Sys_part3_3 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException  [static]
```

Part 3 + filter of pipeline for miRNA muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.28 mirna63Sys_part4()

```
static void masterPATH.Wrapper.mirna63Sys_part4 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException  [static]
```

Part 4 of pipeline for miRNA muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.29 mirna63Sys_random()

```
void masterPATH.Wrapper.mirna63Sys_random (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'p-values' of pipeline for miRNA muscle differentiation screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.30 ogg1_add_connections()

```
Network masterPATH.Wrapper.ogg1_add_connections (
              Network all,
```

```
            NetworkManager nutils,
            PathwayManager putils,
            CentralityManager outils,
            DBManager dbutils,
           Map< String, String[]> hugo ) throws IOException
```

Add low confident connections for OGG1 protein

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object
hugo	HGNC ids map

2.12.2.31 ogg1_part1()

```
void masterPATH.Wrapper.ogg1_part1 (
            Network all,
            NetworkManager nutils,
            PathwayManager putils,
            CentralityManager outils,
            DBManager dbutils ) throws IOException
```

Part 1 of pipeline for DNA repair screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.32 ogg1_part1_ppi()

```
void masterPATH.Wrapper.ogg1_part1_ppi (
            Network all,
            NetworkManager nutils,
            PathwayManager putils,
            CentralityManager outils,
            DBManager dbutils ) throws IOException
```

Part 1 of pipeline for DNA repair screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.33 ogg1_part2()

```
void masterPATH.Wrapper.ogg1_part2 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 2 of pipeline for DNA repair screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.34 ogg1_part2_ppi()

```
void masterPATH.Wrapper.ogg1_part2_ppi (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 2 of pipeline for DNA repair screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.35 ogg1_part3()

```
void masterPATH.Wrapper.ogg1_part3 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 3 of pipeline for DNA repair screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.36 ogg1_part4()

```
void masterPATH.Wrapper.ogg1_part4 (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 4 of pipeline for DNA repair screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.37 ogg1_part4_ppi()

```
void masterPATH.Wrapper.ogg1_part4_ppi (
              Network all,
```

```
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils ) throws IOException
```

Part 4 of pipeline for DNA repair screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.38 ogg1_random()

```
void masterPATH.Wrapper.ogg1_random (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'p-values' of pipeline for DNA repair screen in mixed network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.39 ogg1_random2_ppi()

```
void masterPATH.Wrapper.ogg1_random2_ppi (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'random network' of pipeline for DNA repair screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

2.12.2.40 ogg1_random_ppi()

```
void masterPATH.Wrapper.ogg1_random_ppi (
              Network all,
              NetworkManager nutils,
              PathwayManager putils,
              CentralityManager outils,
              DBManager dbutils,
              RandomManager rand ) throws IOException
```

Part 'pvalues' of pipeline for DNA repair screen in ppi network

all	**Network** (p. 14)
nutils	**NetworkManager** (p. 16) object
putils	**PathwayManager** (p. 24) object
outils	**CentralityManager** (p. 3) object
dbutils	**DBManager** (p. 9) object

The documentation for this class was generated from the following file:

- Wrapper.java

Index