Steps	Platform	Methods
Import raw data	Agilent 1/2-color Affimetrix Illumina Methylation	read.maimages from the limma R package[1] justRMA from the affy R package [2] read.metharray.exp from the minfi R package [3]
Quality Check	Agilent 1/2-color Affimetrix Illumina Methylation	arrayQualityMetrics [4] yaqc function from the yaqcaffy R package[5] shinyMethyl [6]
Probe Quality Estimation	Agilent 1/2-color Affimetrix Illumina Methylation	Custom function(); by using signal distribution of negative control probes - detectionP function from the minfi R package [3]
Diagnostics Plots	Agilent 1/2-color Affimetrix Illumina Methylation	Density plot, MDS plot, boxplot
normalisation	Agilent 1/2-color Affimetrix Illumina	Between array normalisation, quantile normalisation, VSN normalisation, cyclic loess normalisation available in the R limma package [1]; Handled by justRMA [2] ; preprocessRaw, SWAN normalisation [7], quantile normalisation [8], preprocessIllumina, Funnorm normalisation [9], Noob normalisation [10];
Batch Effect Estimation	Agilent 1/2-color Affimetrix Illumina Methylation	confounding plot [11]; prince plot[11]; plotMDS [1]
Batch Effect Mitigation	Agilent 1/2-color Affimetrix Illumina Methylation	Known ComBat function from the R sva package[12] Unknown sva function from the R sva package [13]; ComBat function from the R sva package [12]
Annotation	Agilent 1/2-color Affimetrix Illumina Methylation	specific annotation from eArray specific annotation from Brainarray specific annotation from Bioconductor
Differential Analysis	Agilent 1/2-color Affimetrix Illumina Methylation	fitting linear models by means of the limma R package [1]: Functions: makeContrasts, lmFit, eBayes, topTable
Data Exploration	Agilent 1/2-color Affimetrix Illumina Methylation	venn diagrams; upset plots; volcano plots; heatmaps

Table S1: Main steps of the eUTOPIA preprocessing pipelines. eUTOPIA pipelines are available for the Agilent 1-color, Agilent 2-color, Affymetrix gene expression platforms and Illumina Methylation platform.

		eUTOPIA	AGA	shinyMethyl	MeV	O-miner	Chipster	Babelomics
Quality Check		Yes	No	Yes	No	Yes	Yes	No
Probe Filtering		Yes	No	Yes	Yes	Yes	Yes	No
normalisation		Yes	Yes	Yes	Yes	Yes	Yes	Yes
Batch effect	Known variables	Yes	Automated	No	No	Yes	Yes	No
	Surrogate variables	Yes	Automated	No	No	No	Yes	No
Differential expression		Yes	Yes	No	No	Yes	Yes	Yes
Graphical User Interface		Yes	Yes	Yes	Yes	Yes	Yes	Yes
Reporting		Yes	Yes	Yes	Yes	Yes	Yes	Yes
Platforms	Agilent 2-color microarray	Yes	No	No	Yes	Yes	Yes	Yes
	Agilent 1-color microarray	Yes	No	No	Yes	Yes	Yes	Yes
	Affymetrix microarray	Yes	Yes	No	Yes	Yes	Yes	Yes
	Illumina methylation microarray	Yes	Yes	Yes	No	Yes	Yes	No
	Illumina expression microarray	No	No	No	No	Yes	Yes	No
	RNA-seq	No	Yes	No	Yes	Yes	Yes	No
Technology		R/Shiny	R/Shiny	R/Shiny	Java	HTML web service	Java	HTML

Table S2: Comparison of eUTOPIA to other existing tools.

Step	Method	More details
Data input	Import excel file by using the readxl R library	FunMappOne accepts gene symbols, Ensemble or Entrez gene ID for Human, Mouse and Rat.
Enrichment computation	R package gProfilerR [14]; R package gprofiler2 [15]	Enrichment is performed by means of the hypergeometric test. P-values can be corrected by means of the gSCS, FDR or Bonferroni methods [16].
FunMappone Creation	A hierarchical structure of the enriched term is used to organise and summarise them in the plot.	Summarisation and annotation of the enriched terms can be performed as the mean, median, minimum or maximum value of the provided modifications.
Distance matrix computation	Jaccard index (JI) Euclidean distance a combination of the two	If JI is used then the experimental conditions are clustered together based on shared enriched terms. If Euclidean distance is used then the clustering is driven by the modification of the shared enriched terms. A combination of the two methods will give results both based on the shared enriched terms, but also considering how similar are the enriched terms with respect to their enrichment p-value or summary statistic.
Clustering of experiments	Hierachical clustering with complete, single or ward linkage methods	Hierarchical clustering is applied on the FunMappOne matrix. The heat map will be plotted by arranging the columns according to the their clustering membership.

Table S3: Main steps implemented in FunMappOne.

Feature/Tool	**FunMappOne**	g:profiler	DAVID	clusterProfiler	Enrichr	ToppGene	Goplot	BACA
Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways	yes	yes	yes	yes	yes	yes		yes
Reactome pathways	yes	yes	yes	yes	yes	yes		
Gene Ontology (GO)	yes	yes	yes	yes	yes	yes	yes	yes
Graphical representation	yes			yes	yes	yes	yes	yes
Graphic User interface (GUI)	yes	yes	yes		yes			
Hierachical summarization	yes							
Comparison of multiple experiments	yes			yes				yes
Clustering based on functional redundancy	yes							
Mapping expression modification values on terms	yes						yes	

Table S4: Comparison of FunMappOne to other existing tools.

Step	**Methods**	**Description**
Compute correlation matrix	Pearson correlation Spearman correlation Kendall correlation Mutual information	Estimates pairwise gene correlations based on their corresponding gene expression values.
Infer Gene co-expression network	Aracne [17] CLR [18] MRNET [19] MRNETb [20]	From the correlation/mutual information matrices gene co-expression networks are inferred.
Combinations of inferred networks into a consensus network	BORDA [21]	The individual networks are combined into an ensembl network. Each networks edges are ranked based on their score and combined via BORDA [21]. From the combined ranking edges are selected until each node of the network has received at least one edge or the top n% of edges can be selected.
Community Detection	Walktrap [22] Springlass [23] Louvain [24] Greedy [25]	Communities on the constructed network are computed.
Community evaluation & annotation	Centrality scores Over-representation Jaccard Index	The detected communities are evaluated on their significance and a gene enrichment over GO terms is performed. Similarity scores between the annotated communities are computed.

Table S5: Description of INfORM steps and the implemented methods

Tool	**Language**	**Network inference from gene expression data**	**Ensemble strategy**	**Community detection**	**Annotation of communities**
INfORM [26]	R	***yes***	***yes***	***yes***	***yes***
WGCNA [27]	R	***yes***	no	***yes***	***yes***
CoExp [28]	R, ASP.NET	***yes***	no	***yes***	***yes***
GWENA [29]	R	***yes***	no	***yes***	***yes***

Table S6: Comparison of INFORM to other similar tools.

Table S7: Description of VOLTA modules and methodologies.

Module	Description	Example of implemented method
Distance & Similarity	This module contains 4 sub-modules, providing functions for distance/similarity measures between networks on a global and local scale. Combined the sub-modules contain 53 exposed functions.	On a global level networks can be for example characterised and compared based on their node/ edge centrality values as well as their overall similarity in contained nodes & edges. On a local level small sub-graphs, so called graphlets, can be computed and compared between networks.
Pipeline Wrappers	This module contains 6 sub-modules, which provide wrapper functions across the individual functions of the other modules in order to provide predefined analysis pipelines for co-expression networks.	
Simplification	This module can be used to reduce the complexity of large networks, by for example reducing the number of edges contained. The module contains 7 exposed functions.	Different methods for node and edge removal are implemented, such as edge removal based on edge weight, edge modularity or through the estimation of a spanning tree.
Community	This module provides several algorithms to detect community structures in a provided network as well as metrics to evaluate the partitioning. The module contains 29 exposed functions.	Algorithms for unweighted graphs, such as the walktrap algorithm, methods for weighted graphs, such as the Louvain algorithm and algorithms for overlapping communities, such as Angel are implemented.
Clustering	This module uses pairwise computed distances between a group of networks, such as can be estimated with the metrics implemented in the Distance & Similarity module to cluster a group of networks. The module contains 9 exposed functions.	For example different clustering algorithms, such as hierarchical clustering, k-mediod clustering or affinity clustering as well as consensus strategies and evaluation metrics are provided.
Common Sub-patterns	This module aims at identifying common sub-structures or statistical over-represented structures in a group of networks. The module contains 7 exposed functions.	This module aims at identifying common sub-structures or statistical over-represented structures in a group of networks. The module contains 7 exposed functions.
Plotting	This module provides different functions to visualise the networks, their community structures, the clustering results and other values estimated in the other modules. The module contains 7 exposed functions.	

Tool	**Language**	**Exposing of individual functions**	**File format restrictions**	**Community functions**	**Network similarities & Network clustering**	**Identification of common sub-structures**	**Network simplification**	**Network metrics**
VOLTA [30]	Python	***yes***	***no***	***yes***	***yes***	***yes***	***yes***	***yes***
NetworkX [31]	Python	***yes***	***no***	some	no	no	***yes***	***yes***
iGraph [32]	R / Python	***yes***	***no***	***yes***	no	no	***yes***	***yes***
CDLIB [33]	Python	***yes***	***no***	***yes***	no	no	no	no
BioNetStat [34]	R	no	yes	no	some	no	some	some
InfORM [26]	R	no	yes	some	no	no	some	some
CoNekT [35]	Python/ JavaScript	no	yes	some	some	some	N/A	some
CompNet [36]	Perl/ R	no	yes	some	***yes***	***yes***	N/A	some
NetSimile [37]	N/A	N/A	N/A	no	***yes***	no	no	***yes***
WGCNA [27]	R	***yes***	yes	***yes***	no	some	N/A	***yes***

Table S8: Comparison of VOLTA to other existing similar tools, packages & software applications.

Step	**Method**	**Description**

Data	ENM
	Drugs
	Chemicals
	Diseases
		from the Nanominer database [38].
		from the CMAP database [38]
		from the CTD database [39].
		from the CTD database [39].

Representation	ENM Drugs Diseases Chemicals	rank of genes rank of genes Sets of associated genes Sets of associated genes
Similarity	Jaccard Index GSEA Kendall Tau	Between sets of genes Between rank and set of genes Between ranks of genes
Other similarity	Drugs vs. Drugs Drugs vs. Diseases Drugs vs. Chemicals Diseases vs. Diseases Chemical vs. Chemical Diseases vs. Chemicals	Targets, Smiles Medical prescription Smiles Symptoms Smiles Downloaded from CTD [39]
Validation	Mantel test [40]	Used to compare transcriptomic based similarity and the other similarity
Integration	Similarity Network	Similarity scores where normalised by means of the cumulative function and used as edge weights to connect pairs of phenotypic entities
Contextualisation	Clique	Search all possible cliques, rank them by their association strength and prioritise those with known connections

Table S9: Method used in INSIdE NANO for the contextualisation of engineeredx nanomaterials (ENM)

Step	Method	More Details
Gene Filtering	Anova	Fit an analysis of variance model [41] as implemented in the stats R package [42]
	Trend test	The Mann-Kendall trend test [43, 44] is performed as implemented in the trend R package [45].
Model Fitting and Selection	Models	Linear, Polynomial (2nd and 3rd order), Hill (Kd=10, n=(0.5,1,2,3,4,5), Power (d=2,3,4), Exponential, Log-logistic*, Weibull*, Brain-Counsen*, Asymptotic*, Michaelis-Mentel* *Available from the drc R package [46]
	Lack-of-fit	Models with lack-of-fit pvalue lower than a threshold are filtered. Default: $p < 0.1$
	AIC	For the same gene, the optimal model is selected as the one with lowest Akaike Information Criteria [47] computed with the R stats package [42]
Doses estimation	BMD, BMDL, BMDU estimated from the optimal fitting model	The user need to specify: • the BMR factor • if the assumption of constant variance is true • the confidence interval used to compute BMDL and BMU
Functional Annotation	FunMappOne	The FunMappOne functionalities are embedded into the BMDx tool

Table S10: Methods implemented in the BMDx tool

Software	Platform	Models	Multiple Experiments	Transcriptomics based	Normalisation	Functional Analysis	Comparative Visualization
BMDx	R/Shiny	Linear Polynomial Hill Power Exponential Log-logistic* Weibull* Brain-Counsen* Asymptotic* Michaelis-Mentel*	Yes**	Yes	No***	Yes****	Yes
FastBMD	Web	Linear Polynomial Hill Power Exponential	No	Yes	Yes	Yes	Yes
BMDExpress2	Locally installed	Linear Polynomial Hill Power Exponential	Yes	Yes	No	Yes	Yes
DROmics	R, Web	Linear Hill Exponential Gauss-probit Log-Gauss-probit	No	Yes	Yes	No	No
BMDS	Locally installed	Linear Polynomial Hill Power Exponential	No	No	No	No	No

Table S11: Comparison of BMDx to other existing tools
*: available from the R drc package
**: Specifically designed for the comparisons of multiple chemical exposures and/or multiple time-points
***: assumes that the data are already pre-processed and normalised. This can be done with the eUTOPIA tool
****: The FunMappOne tool is included into the BMDx graphical interface

Step	Method	More Details
Gene Filtering	Anova	Fit an analysis of variance model [41] as implemented in the stats R package [42]
Model Fitting and selection	Models	Linear, 2nd and 3rd order polynomial
	ANOVA	The best-fitting model is selected performing a nested model hypothesis test [48]
Responsive Area	Contour Plots	The selected model is used to predict the gene deregulation values in a a continuous map
Active region	Thresholding	The active region is identified on the contour map as the one above a certain activity threshold selected by the user.
POD estimation		Based on the position of the active region on the map, the gene is labelled as an early, middle or late, with respect to time, and as sensitive, intermediate or resilient, with respect to the dose.

Table S12: Methods implemented in the TinderMIX tool

Module	**Method**	**Description**
Fuzzy patterns	R package DFP [49]	This module is used to extract the fuzzy patterns from the input gene expression data.
Feature Prioritisation	Random Forest [50]	This module is used to train a Random Forest model on the fuzzy patterns data
Feature Ranking	Mean Decrease Accuracy	This module is used to compute the retrieve the most important variables estimated by the Random Forest

Table S13: Main steps of the FPRF method

Tool	**Language**	**Features**	**Feature Importance**	**Model**
FPRF [51]	R	Fuzzy patterns	Feature permutation	Random Forest
Boruta [52]	R	Input features	Feature permutation	Random Forest
varSelRF [53]	R	Input features	Recursive elimination	Random Forest
PIMP [54]	R	Input features	Label permutation	Random Forest
GRRF [55]	R	Input features	Regularised Information Gain	Regularised Random Forest
L1-eSVM [56]	-	Linear Kernel	Bootstrapped stability score	Support Vector Machines

Table S14: Comparison of FPRF to other existing tools.

Operator	**Description**
Chromosome of variable length	This allows encoding admissible solutions more efficiently compared to the fixed binary representation. This reduces computation requirements during evolution
Dynamic genetic operators	Mutation, selection, and cross-over operators adjust their behaviours dynamically to enforce gene elitism and gene set size optimisation. Adjustments are performed by a fuzzy logic controller which reacts to the different situations that happen during evolution
Isolated populations and migration	To allow the emergence of multiple equally optimal solutions, multiple isolated populations are generated throughout evolution. However, very fit individuals have a chance to migrate to other populations, increasing the chances of reproduction of the high ranking individuals
Fitness evaluation	The fitness of each individual is scored using a 3-fold cross-validated performances of a random forest trained on the feature set encoded in the chromosome of the individuals

Table S15: Key points of the GARBO method

Tool	**Language**	**Method**	**Quality evaluation**	**Multiple features sets**	**Enforce minimal sets size**
GARBO [57]	Python	GA (wrapper)	predictive performance	yes	yes
OptSelect [58]	R	PSO (wrapper)	predictive performance	no	no
Wang et al. [59]	MATLAB	Markov Blanket (wrapper)	predictive performance	no	no
Sun et al. [60]	MATLAB	Lagrange Multipliers (filter)	minimum redundancy maximum relevance	no	yes
Saeys et al. [61]	Java	Ensemble (filter, embedded)	Symmetric uncertainty + RF variable importance	no	no

Table S16: Comparison of GARBO to similar tools.

Module	**Description**
Multi-niche population	The population of candidate descriptors is partitioned into a number of disjoint sets that evolve independently to allow an efficient exploration of the solution space. Regular interactions among the niches ensures that the components of the best solutions are represented in the population.
Genetic operators	Mutation and cross-over genetic operators perturb the most promising solutions to explore different descriptor sets trading-off exploration and exploitation of the current best.The non-dominant selection operator finds at each iteration the individuals with the best trade-offs among the multiple objective criteria to be optimised
Fitness functions	Multiple criteria are evaluated to estimate the fitness of each individual. These criteria encompass many QSAR quality assessment measures including such as $RMSE$, R^2,Q^2, $Q^2_{F_1}$,$Q^2_{F_2}$, $Q^2_{F_3}$, CCC [62]. Moreover, the number of molecular descriptors used is considered, and the applicability domain of the fitted model. Applicability domain is computed by means of the Williams Plot [63]
Model training	Different categories of trainable models are offered including regression model [64], support vector regression [65] and k-nearest neighbours (kNN) for regression [66]

Table S17: Key points of the MANGA methodology.

Tool	**Language**	**Features**	**Feature Selection**	**Model**	**Validation**	**Applicability domain**
MaNGA [62]	Python	Molecular descriptors	Multi-objective Genetic Algorithm	Linear, kNN, SVM	Internal / external	Williams plot
hyQSAR [67]	R	Molecular descriptors, mechanism of action	LASSO	Linear, non-linear with feature transforms	Internal / external	Williams plot
CORAL [68]	-	SMILES based optimal descriptors	-	Univariate, Linear	-	-
QSAR-Co [69]	Java	Molecular descriptors	Filter, Genetic Algorithm	Linear, Random Forest	Internal / external	Standardisation method, Confidence estimation
QSARINS [70]	-	Molecular descriptors	Filter, Genetic Algorithm, Forward feature selection	Linear	Internal / external	Williams plot

Table S18: Comparison of QSAR tools.

Module	**Description**
Feature transformation	In some cases, a power transformation of the input features may help improve the correlation with the response variable. For this reason, hyQSAR allows the estimation of a power transform if it improves the error estimates of the model.
Parameter Tuning	Feature transform parameters are tuned during training using a grid search based on a random split validation algorithm in which a fraction of the samples are repeatedly selected at random as a validation set while the rest is used to train the model.
Model Training	The main assumption of hyQSAR is that the response variable is dependent on a reduced set of features, therefore the model trained is a LASSO regression of the response variable in which the vector of coefficients is constrained to have as few non-zero entries as possible.
Model Validation	Both measures based on the predictive error of the model, as well as measures based on the applicability domain of the models, are verified to assess the quality of the model fit.

Table S19: Description of the hyQSAR methodology.

Step	**Description**
Preprocessing	Omic features with low variance are removed from each data layer
Omic feature Clustering	To reduce the dimensionality of the data, the omic features are clustered in each data layer. For each cluster, a prototype is identified and used for further steps. Clustering algorithms available are: Pvclust [71], SOM [72], hierarchical clustering with Ward's method,[73] K-means [74], Partitional Around Medoids [75] and Spectral clustering [76]. Clustering goodness is evaluated employing a score (ranging from 0 to 1) that considers sample correlation cohesion, the number of singletons, and the compression rate.
Prototype Selection	In the case of the supervised approach, the cluster prototypes are ranked based on their class separability performances. Feature ranking can be performed using the CAT-score [77] and/or the Mean Decreasing Accuracy index calculated by Random Forests [50].
Sample Clustering	In each data layer, samples are clustered by using only the cluster prototype from previous steps. Clustering algorithms available are the same as the Omic feature clustering.
Multi-view clustering	The sample clusterings from the individual data layer are merged in a late integration fashion. Two integrative approaches are available: a matrix factorisation strategy [78] and a general model for multi-view integration [79]

Table S20: Description of the MVDA methodology.

Tool	Language	Integration type	Integration Method	Learning type
MVDA [80]	R	Late integration	Matrix factorization	Unsupervised or semi-supervised
SNF [81]	R, Matlab	Intermediate integration	Network fusion	Unsupervised
mixKernel [82]	R	Intermediate integration	Multiple kernel learning	Unsupervised
iCluster [83]	R	Intermediate integration	Matrix factorization	Unsupervised

Table S21: Comparison of MVDA to other existing tools.

Module	Description
Network Generation	This module is used to generate a regulatory network comprised of mRNA, transcription factors and miRNAs
Experiment Simulation	This module is used to simulate a transcriptomics experiment given a generated regulatory network. Two levels of noise are added to the experiment to simulate real experimental data.

Table S22: Description of the MOSIM methodology

Tool	Language	Network Topologies	Network Motifs	Regulation type	Regulators interaction	Modelled Entities	Network Simulation
MOSIM [84]	R	hierarchical modular	yes	activation, inhibition	cooperation, synergy, antagonism	mRNA, miRNA, TF	Built-in
AGN [85]	C++, SAS, PERL	small-world, scale-free, random network	no	activation	synergy	mRNA, TF	External tool
SynTReN [86]	JAVA	Sub-sample of real network	yes	activation, inhibition	cooperation, synergy, antagonism	mRNA, TF	External tool
netsim [87]	R	hierarchical modular	yes	activation, inhibition	cooperation, synergy, antagonism	mRNA, TF	Built-in
GRENDEL [88]	C++	scale-free	no	activation, inhibition	synergy	mRNA, TF	External tool
RENCO [89]	C++	scale-free	no	activation	cooperation, synergy	mRNA, TF	External tool
GeNGe [90]	Web application	scale-free, random network, modular	yes	activation, inhibition	cooperation, synergy, antagonism	mRNA, TF	External tool

Table S23: Comparison of MOSIM to other existing tools.

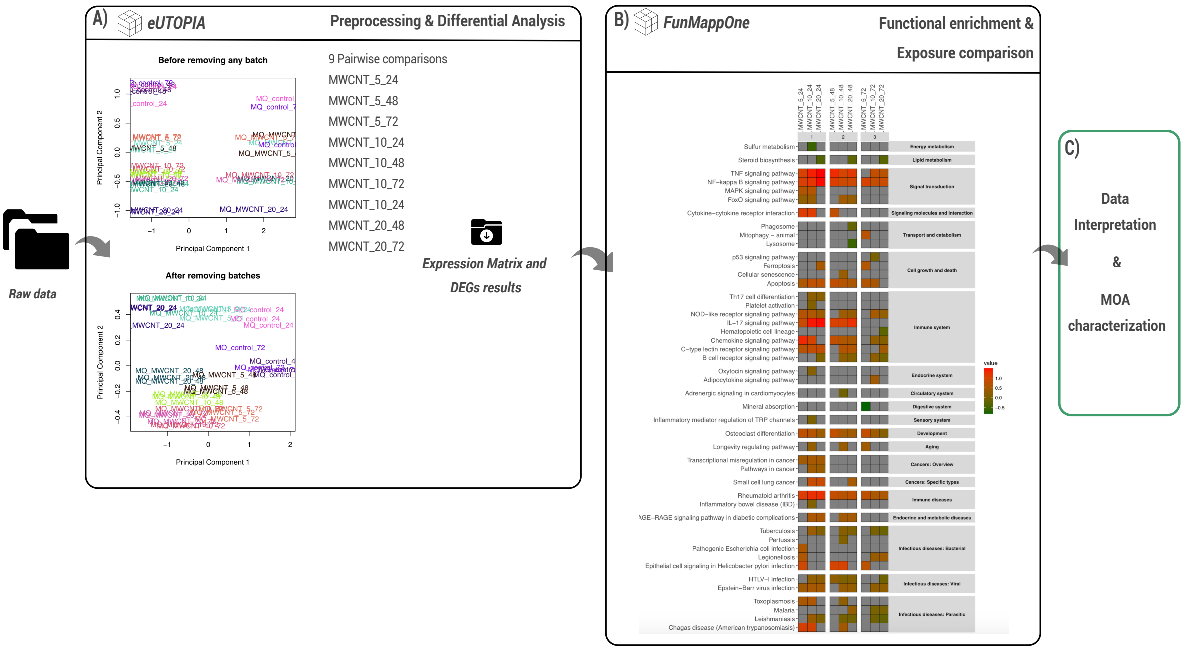

Figure S1: **Example application of the characterisation of the MWCNT MOA employing FunMappOne.** (A) eUTOPIA was used to preprocessing the raw data and to perform differential analysis. The spreadsheet file, containing the lists of differentially expressed genes, can be exported from eUTOPIA in a format that is ready-to-use for FunMappOne. (B) The enrichment analysis with respect to the KEGG human pathways was performed. The FunMappOne map is coloured by the genes modification value (their log-fold-change) aggregating by means of the mean function. (C) The output was interpreted for MOA characterisation of MWCNT exposures at different doses and time points.

1 Supplementary Methods

1.1 Interoperability of Nextcast data formats with external tools

Gene expression matrices obtained from eUTOPIA are saved as tab-delimited text files with the genes in the rows and the samples in the columns; These matrices can be easily uploaded into the MORPHEUS tool (`https://software.broadinstitute.org/morpheus`) for dynamic visualisation and clustering. Moreover, they can be imported into R or python for more advanced data visualisation such as data projection with t-SNE [91] and UMAP [92] or to identify coherent groups of genes or samples employing clustering analysis. The results of the differential expression analysis can be exported as a spreadsheet file. The results are provided in the format adopted by the Bioconductor limma package [1]. The file contains a list of genes, log2-fold changes, and p-values computed for each one of the comparisons performed during the analysis. These lists can be easily provided to online tools for functional annotation such as WebGestalt [93] or Enrichr [94]. Moreover, network-based enrichment analysis can be performed by passing the set of genes to online tools like PathwAX [95]. The output of the enrichment analysis performed with FunMappOne can be exported in a spreadsheet format. The file contains as many sheets as the number of experimental conditions compared in the FunMappOne module. When performing the enrichment with respects to the Gene Ontology database, the resulting lists of enriched terms and their associated p-values resulting can be provided as input to the online REVIGO tool (`http://revigo.irb.hr/`) for summarization and to study their interactions. The networks generated by INfORM can be exported in multiple standardised formats, such as a text file containing a tab-delimited adjacency matrix or edge list, or as GraphML format. These are commonly accepted format for many network visualisation and analysis tools such as Cytoscape [96] and Gephy [97]. The list of differentially expressed genes identified with eUTOPIA or the sets of genes in specific modules of co-expression networks identified with INfORM, can be further investigated through the Ingenuity Pathway Analysis tool as shown in [98, 99]. Moreover, the relationships between the genes can also be studied by using the STRING analysis engine and retrieving connections between the proteins whose genes are mapped to [100].

1.2 Example of application of the Nextcast pipelines on real data

1.2.1 Preprocessing and differential analysis with eUTOPIA

First, the raw data were uploaded to eUTOPIA together with the metadata table describing both experimental and technical variables relevant for the process. The preprocessing was then performed following the guided interface of the tool [101]. Shortly, probes with intensities higher than the 75% quantile of the negative control probes in at least 85% of the samples were retained for further preprocessing. The data were then quantile-normalised between arrays. Observed batch effects, particularly those associated with variables "dye", "slide", and "row" were corrected using the ComBat method [102]. The probes were then annotated to gene symbols and aggregated by the median expression values of the probes mapped to the same entity. Finally, differential expression between sample groups was evaluated with limma [1] using the corrected batches as covariates for the model. The combination of three exposure doses (5, 10, and 20 μg) and three time points (24, 48 and 72 hours) yields a total of nine comparisons when each dose-time point combination is compared to its corresponding control samples. Genes were considered significantly differentially expressed with an absolute log2-fold change > 0.58 and Benjamini & Hochberg adjusted p-value < 0.05. The preprocessed and aggregated gene expression data and the results of the differential analysis were exported from eUTOPIA. Additionally, the data were exported in a format that is ready to use for the basic usage of the FunMappOne and BMDx tools. The following sections will describe the further analytical pipelines using this data in the context of the suggested pipelines in Figure 2 and Figure 3, respectively.

1.2.2 MOA characterisation with INfORM and FunMappOne

In order to select the most relevant genes for each MWCNT exposure, individual gene co-expression networks for each pairwise comparison resulting from the differential expression analysis were generated. This experimental design required the conversion of the expression matrix and differential expression analysis results into nine different inputs for the analysis with INfORM. This transformation was performed by using a custom R script. To make a fair comparison between the 9 different sets of genes and to maximise the differences across the exposures, we imposed all the networks to share the same nodes and ranked the genes based on how many times they were differentially expressed across the nine conditions. Only the top 1,000 genes of this rank were used to build the gene co-expression networks, so that all the exposures shared the same genes while preserving the more frequently deregulated ones. The networks were inferred using default parameters in INfORM. The individual gene-rank scores were used for prioritisation. Lastly, to fully characterise and compare the MOA of each exposure, the top 200 genes of each network rank were functionally annotated with FunMappOne. The selection of the top ranked genes in each of the nine network and their combination in the expected format by FunMappOne was performed by means of a custom script. KEGG enrichment analysis was performed with FunMappOne by using all default parameters. The summarised results were plotted as a heatmap and annotated with the second level of the KEGG hierarchy. The R scripts used in this example are available at `https://github.com/fhaive/nextcast`.

1.2.3 Benchmark dose analysis with BMDx

The preprocessed and aggregated gene expression matrix and the metadata table were exported from eUTOPIA in a format compatible with BMDx. The BMD analysis was then performed under the assumption of constant variance with the benchmark response (also called BMR factor) of 1.349. The confidence interval was set as 0.95 and the lack-of-fit threshold as 0.1. Further filters for the results were defined based on the exposure doses. Namely, genes with BMD or BMDU values extrapolated higher than the highest exposure dose ($20\mu g$) were removed. Also, genes whose predicted values have high ratio (BMD/BMDL$>$ 20, BMDU/BMD$>$ 20, and BMDU/BMDL$>$ 40) were filtered from the analysis. Finally, multiple models (linear, quadratic, power2, exponential, hill05, hill1,hill2) were fit to each gene and the optimal model was selected based on the combination of the Akaike Information Criterion and the filtering criteria defined above. The genes that fit a model according to these criteria were considered dose-dependent. The results can be investigated at the level of individual genes, and their independent BMD values can be explored visually, or in the spreadsheet files that can be exported for further analysis and interpretation. However, BMDx contains an integrated implementation of the functional enrichment tool FunMappOne. Hence, the functional enrichment of the results can be performed and investigated as part of the same run. Alternatively, the BMD analysis results in the spreadsheet file can be easily transformed into the input for the stand-alone version of FunMappOne, allowing the user to come back to the results of the functional enrichment at another time. Here, the functional enrichment was performed against KEGG pathways (p-values were FDR corrected and considered significant when lower than 0.001 using all annotated genes as the background). The colour of the cell represents the mean BMD values of the genes enriching the pathway. Alternatively, the cell can be coloured based on the minimum BMD value of the genes resulting in a more conservative estimation. The results of the functional enrichment can be downloaded as a spreadsheet file containing all enriched terms and the genes enriching them at each time point. Additionally, the enrichment heatmap can be downloaded as a PDF file.

References

[1] M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, G. K. Smyth, limma powers differential expression analyses for RNA-sequencing and microarray studies, Nucleic Acids Research 43 (2015) e47.

[2] L. Gautier, L. Cope, B. M. Bolstad, R. A. Irizarry, affy—analysis of affymetrix genechip data at the probe level, Bioinformatics 20 (2004) 307–315.

[3] M. J. Aryee, A. E. Jaffe, H. Corrada-Bravo, C. Ladd-Acosta, A. P. Feinberg, K. D. Hansen, R. A. Irizarry, Minfi: a flexible and comprehensive bioconductor package for the analysis of infinium dna methylation microarrays, Bioinformatics 30 (2014) 1363–1369.

[4] A. Kauffmann, R. Gentleman, W. Huber, arrayqualitymetrics–a bioconductor package for quality assessment of microarray data, Bioinformatics 25 (2009) 415–6.

[5] L. Gatto, yaqcaffy: Affymetrix expression data quality control and reproducibility analysis, R package version 1 (2017).

[6] J.-P. Fortin, E. Fertig, K. Hansen, shinymethyl: interactive quality control of illumina 450k dna methylation arrays in r, F1000Research 3 (2014).

[7] J. Maksimovic, L. Gordon, A. Oshlack, Swan: Subset-quantile within array normalization for illumina infinium humanmethylation450 beadchips, Genome biology 13 (2012) 1–12.

[8] N. Touleimat, J. Tost, Complete pipeline for infinium® human methylation 450k beadchip data processing using subset quantile normalization for accurate dna methylation estimation, Epigenomics 4 (2012) 325–341.

[9] J.-P. Fortin, A. Labbe, M. Lemire, B. W. Zanke, T. J. Hudson, E. J. Fertig, C. M. Greenwood, K. D. Hansen, Functional normalization of 450k methylation array data improves replication in large cancer studies, Genome biology 15 (2014) 1–17.

[10] J. Halvardson, A. Zaghlool, L. Feuk, Exome rna sequencing reveals rare and novel alternative transcripts, Nucleic acids research 41 (2013) e6–e6.

[11] M. Lauss, swamp: Visualization, Analysis and Adjustment of High-Dimensional Data in Respect to Sample Annotations, 2019. URL: `https://CRAN.R-project.org/package=swamp`, r package version 1.5.1.

[12] W. E. Johnson, C. Li, A. Rabinovic, Adjusting batch effects in microarray expression data using empirical bayes methods, Biostatistics 8 (2007) 118–127.

[13] J. T. Leek, J. D. Storey, Capturing heterogeneity in gene expression studies by surrogate variable analysis, PLoS genetics 3 (2007) e161.

[14] J. Reimand, R. Kolde, T. Arak, gProfileR: Interface to the 'g:Profiler' Toolkit, 2019. URL: `https://CRAN.R-project.org/package=gProfileR`, r package version 0.7.0.

[15] L. Kolberg, U. Raudvere, I. Kuzmin, J. Vilo, H. Peterson, gprofiler2– an r package for gene list functional enrichment analysis and namespace conversion toolset g:profiler, F1000Research 9 (ELIXIR) (2020). R package version 0.2.1.

[16] J. Reimand, M. Kull, H. Peterson, J. Hansen, J. Vilo, g: Profiler—a web-based toolset for functional profiling of gene lists from large-scale experiments, Nucleic acids research 35 (2007) W193–W200.

[17] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, A. Califano, Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, in: BMC bioinformatics, volume 7, Springer, 2006, pp. 1–15.

[18] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, T. S. Gardner, Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles, PLoS biology 5 (2007) e8.

[19] P. E. Meyer, K. Kontos, F. Lafitte, G. Bontempi, Information-theoretic inference of large transcriptional regulatory networks, EURASIP journal on bioinformatics and systems biology 2007 (2007) 1–9.

[20] P. Meyer, D. Marbach, S. Roy, M. Kellis, Information-theoretic inference of gene networks using backward elimination., in: BioComp, Citeseer, 2010, pp. 700–705.

[21] M. G. Schimek, E. Budinská, K. G. Kugler, V. Švendová, J. Ding, S. Lin, Topklists: a comprehensive r package for statistical inference, stochastic aggregation, and visualization of multiple omics ranked lists, Statistical Applications in Genetics and Molecular Biology 14 (2015) 311–316.

[22] P. Pons, M. Latapy, Computing communities in large networks using random walks, in: International symposium on computer and information sciences, Springer, 2005, pp. 284–293.

[23] M. E. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical review E 69 (2004) 026113.

[24] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of statistical mechanics: theory and experiment 2008 (2008) P10008.

[25] A. Clauset, M. E. Newman, C. Moore, Finding community structure in very large networks, Physical review E 70 (2004) 066111.

[26] V. S. Marwah, P. A. S. Kinaret, A. Serra, G. Scala, A. Lauerma, V. Fortino, D. Greco, Inform: Inference of network response modules, Bioinformatics 34 (2018) 2136–2138.

[27] P. Langfelder, S. Horvath, Wgcna: an r package for weighted correlation network analysis, BMC bioinformatics 9 (2008) 1–13.

[28] S. García-Ruiz, A. L. Gil-Martínez, A. Cisterna, F. Jurado-Ruiz, R. H. Reynolds, M. R. Cookson, J. Hardy, M. Ryten, J. A. Botía, et al., Coexp: A web tool for the exploitation of co-expression networks, Frontiers in genetics 12 (2021) 218.

[29] G. G. Lemoine, M.-P. Scott-Boyer, B. Ambroise, O. Périn, A. Droit, Gwena: gene co-expression networks analysis and extended modules characterization in a single bioconductor package, BMC bioinformatics 22 (2021) 1–20.

[30] A. Pavel, A. Federico, G. Del Giudice, A. Serra, D. Greco, Volta: advanced molecular network analysis, Bioinformatics (2021).

[31] A. Hagberg, P. Swart, D. S Chult, Exploring network structure, dynamics, and function using NetworkX, Technical Report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[32] G. Csardi, T. Nepusz, et al., The igraph software package for complex network research, InterJournal, complex systems 1695 (2006) 1–9.

[33] G. Rossetti, L. Milli, R. Cazabet, Cdlib: a python library to extract, compare and evaluate communities from complex networks, Applied Network Science 4 (2019) 1–26.

[34] V. C. Jardim, S. d. S. Santos, A. Fujita, M. S. Buckeridge, Bionetstat: A tool for biological networks differential analysis, Frontiers in genetics 10 (2019) 594.

[35] S. Proost, M. Mutwil, Conekt: an open-source framework for comparative genomic and transcriptomic network analyses, Nucleic acids research 46 (2018) W133–W140.

[36] B. K. Kuntal, A. Dutta, S. S. Mande, Compnet: a gui based tool for comparison of multiple biological interaction networks, BMC bioinformatics 17 (2016) 1–11.

[37] M. Berlingerio, D. Koutra, T. Eliassi-Rad, C. Faloutsos, Netsimile: A scalable approach to size-independent network similarity, arXiv preprint arXiv:1209.2684 (2012).

[38] L. Kong, S. Tuomela, L. Hahne, H. Ahlfors, O. Yli-Harja, B. Fadeel, R. Lahesmaa, R. Autio, Nanominer—integrative human transcriptomics data resource for nanoparticle research, PloS one 8 (2013) e68414.

[39] A. P. Davis, C. J. Grondin, K. Lennon-Hopkins, C. Saraceni-Richards, D. Sciaky, B. L. King, T. C. Wiegers, C. J. Mattingly, The comparative toxicogenomics database's 10th year anniversary: update 2015, Nucleic acids research 43 (2015) D914–D920.

[40] N. Mantel, The detection of disease clustering and a generalized regression approach, Cancer research 27 (1967) 209–220.

[41] J. M. Chambers, A. E. Freeny, R. M. Heiberger, Analysis of variance; designed experiments, in: Statistical models in S, Routledge, 2017, pp. 145–193.

[42] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2021. URL: `https://www.R-project.org/`.

[43] K. W. Hipel, A. I. McLeod, Time series modelling of water resources and environmental systems, Elsevier, 1994.

[44] C. Libiseller, A. Grimvall, Performance of partial mann–kendall tests for trend detection in the presence of covariates, Environmetrics: The official journal of the International Environmetrics Society 13 (2002) 71–84.

[45] T. Pohlert, trend: Non-Parametric Trend Tests and Change-Point Detection, 2020. URL: `https://CRAN.R-project.org/package=trend`, r package version 1.1.4.

[46] C. Ritz, F. Baty, J. C. Streibig, D. Gerhard, Dose-response analysis using r, PLOS ONE 10 (2015).

[47] Y. Sakamoto, M. Ishiguro, G. Kitagawa, Akaike information criterion statistics, Dordrecht, The Netherlands: D. Reidel 81 (1986) 26853.

[48] T. Teräsvirta, I. Mellin, Model selection criteria and model selection tests in regression models, Scandinavian Journal of Statistics (1986) 159–171.

[49] D. Glez-Peña, R. Álvarez, F. Díaz, F. Fdez-Riverola, Dfp: A bioconductor package for fuzzy profile identification and gene reduction of microarray data, BMC bioinformatics 10 (2009) 1–8.

[50] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

[51] V. Fortino, P. Kinaret, N. Fyhrquist, H. Alenius, D. Greco, A robust and accurate method for feature selection and prioritization from multi-class omics data, PloS one 9 (2014) e107801.

[52] M. B. Kursa, W. R. Rudnicki, et al., Feature selection with the boruta package, J Stat Softw 36 (2010) 1–13.

[53] R. Díaz-Uriarte, S. A. De Andres, Gene selection and classification of microarray data using random forest, BMC bioinformatics 7 (2006) 1–13.

[54] A. Altmann, L. Toloşi, O. Sander, T. Lengauer, Permutation importance: a corrected feature importance measure, Bioinformatics 26 (2010) 1340–1347.

[55] H. Deng, G. Runger, Gene selection with guided regularized random forest, Pattern Recognition 46 (2013) 3483–3489.

[56] M. Moon, K. Nakai, Stable feature selection based on the ensemble l 1-norm support vector machine for biomarker discovery, BMC genomics 17 (2016) 65–74.

[57] V. Fortino, G. Scala, D. Greco, Feature set optimization in biomarker discovery from genome-scale data, Bioinformatics 36 (2020) 3393–3400.

[58] E. K. Lee, K. Uppal, Optselect: An algorithm for ensemble feature selection and stability assessment, in: 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2020, pp. 1979–1986.

[59] A. Wang, N. An, J. Yang, G. Chen, L. Li, G. Alterovitz, Wrapper-based gene selection with markov blanket, Computers in biology and medicine 81 (2017) 11–23.

[60] S. Sun, Q. Peng, X. Zhang, Global feature selection from microarray data using lagrange multipliers, Knowledge-Based Systems 110 (2016) 267–274.

[61] Y. Saeys, T. Abeel, Y. Van de Peer, Robust feature selection using ensemble feature selection techniques, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2008, pp. 313–325.

[62] A. Serra, S. Önlü, P. Festa, V. Fortino, D. Greco, Manga: a novel multi-niche multi-objective genetic algorithm for qsar modelling, Bioinformatics 36 (2020) 145–153.

[63] P. Gramatica, Principles of qsar models validation: internal and external, QSAR & combinatorial science 26 (2007) 694–701.

[64] D. A. Freedman, Statistical models: theory and practice, cambridge university press, 2009.

[65] D. Basak, S. Pal, D. Ch, R. Patranabis, Support vector regression. 2007, Neural Information Processing Letters and Reviews (????) 203–224.

[66] Z.-H. Zhou, M. Li, et al., Semi-supervised regression with co-training., in: IJCAI, volume 5, 2005, pp. 908–913.

[67] A. Serra, S. Önlü, P. Coretto, D. Greco, An integrated quantitative structure and mechanism of action-activity relationship model of human serum albumin binding, Journal of cheminformatics 11 (2019) 1–10.

[68] E. Benfenati, A. A. Toropov, A. P. Toropova, A. Manganaro, R. Gonella Diaza, Coral software: Qsar for anticancer agents, Chemical biology & drug design 77 (2011) 471–476.

[69] P. Ambure, A. K. Halder, H. Gonzalez Diaz, M. N. D. Cordeiro, Qsar-co: An open source software for developing robust multitasking or multitarget classification-based qsar models, Journal of chemical information and modeling 59 (2019) 2538–2544.

[70] P. Gramatica, N. Chirico, E. Papa, S. Cassani, S. Kovarich, Qsarins: A new software for the development, analysis, and validation of qsar mlr models, 2013.

[71] R. Suzuki, H. Shimodaira, Pvclust: an r package for assessing the uncertainty in hierarchical clustering, Bioinformatics 22 (2006) 1540–1542.

[72] J. Vesanto, E. Alhoniemi, Clustering of the self-organizing map, IEEE Transactions on neural networks 11 (2000) 586–600.

[73] J. H. Ward Jr, Hierarchical grouping to optimize an objective function, Journal of the American statistical association 58 (1963) 236–244.

[74] J. A. Hartigan, M. A. Wong, Algorithm as 136: A k-means clustering algorithm, Journal of the royal statistical society. series c (applied statistics) 28 (1979) 100–108.

[75] L. Kaufman, P. J. Rousseeuw, Clustering by means of medoids. statistical data analysis based on the l1–norm and related methods, edited by y. dodge, 1987.

[76] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in neural information processing systems, 2002, pp. 849–856.

[77] M. Ahdesmäki, K. Strimmer, Feature selection in omics prediction problems using cat scores and false nondiscovery rate control, The Annals of Applied Statistics 4 (2010) 503–519.

[78] D. Greene, P. Cunningham, A matrix factorization approach for integrating multiple data views, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2009, pp. 423–438.

[79] B. Long, P. S. Yu, Z. Zhang, A general model for multiple view unsupervised learning, in: Proceedings of the 2008 SIAM international conference on data mining, SIAM, 2008, pp. 822–833.

[80] A. Serra, M. Fratello, V. Fortino, G. Raiconi, R. Tagliaferri, D. Greco, Mvda: a multi-view genomic data integration methodology, BMC bioinformatics 16 (2015) 1–13.

[81] B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe-Kains, A. Goldenberg, Similarity network fusion for aggregating data types on a genomic scale, Nature methods 11 (2014) 333–337.

[82] J. Mariette, N. Villa-Vialaneix, Unsupervised multiple kernel learning for heterogeneous data integration, Bioinformatics 34 (2018) 1009–1015.

[83] R. Shen, A. B. Olshen, M. Ladanyi, Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis, Bioinformatics 25 (2009) 2906–2912.

[84] M. Fratello, A. Serra, V. Fortino, G. Raiconi, R. Tagliaferri, D. Greco, A multi-view genomic data simulator, BMC bioinformatics 16 (2015) 1–15.

[85] P. Mendes, W. Sha, K. Ye, Artificial gene networks for objective comparison of analysis algorithms, Bioinformatics 19 (2003) ii122–ii129.

[86] T. Van den Bulcke, K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, K. Marchal, Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms, BMC bioinformatics 7 (2006) 1–12.

[87] B. Di Camillo, G. Toffolo, C. Cobelli, A gene network simulator to assess reverse engineering algorithms, Annals of the New York Academy of Sciences 1158 (2009) 125–142.

[88] B. C. Haynes, M. R. Brent, Benchmarking regulatory network reconstruction with grendel, Bioinformatics 25 (2009) 801–807.

[89] S. Roy, M. Werner-Washburne, T. Lane, A system for generating transcription regulatory networks with combinatorial control of transcription, Bioinformatics 24 (2008) 1318–1320.

[90] H. Hache, C. Wierling, H. Lehrach, R. Herwig, Genge: systematic generation of gene regulatory networks, Bioinformatics 25 (2009) 1205–1207.

[91] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (2008).

[92] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018).

[93] Y. Liao, J. Wang, E. J. Jaehnig, Z. Shi, B. Zhang, WebGestalt 2019: gene set analysis toolkit with revamped UIs and APIs, Nucleic Acids Research 47 (2019) W199–W205.

[94] E. Y. Chen, C. M. Tan, Y. Kou, Q. Duan, Z. Wang, G. V. Meirelles, N. R. Clark, A. Ma'ayan, Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool, BMC Bioinformatics 14 (2013).

[95] C. Ogris, T. Helleday, E. L. Sonnhammer, PathwAX: a web server for network crosstalk based pathway annotation, Nucleic Acids Research 44 (2016) W105–W109.

[96] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, T. Ideker, Cytoscape: a software environment for integrated models of biomolecular interaction networks, Genome research 13 (2003) 2498–2504.

[97] M. Bastian, S. Heymann, M. Jacomy, Gephi: An open source software for exploring and manipulating networks (2009).

[98] P. A. Kinaret, J. Ndika, M. Ilves, H. Wolff, G. Vales, H. Norppa, K. Savolainen, T. Skoog, J. Kere, S. Moya, et al., Toxicogenomic profiling of 28 nanomaterials in mouse airways, Advanced Science 8 (2021) 2004588.

[99] H. Suojalehto, J. Ndika, I. Lindström, L. Airaksinen, K. Karvala, P. Kauppi, A. Lauerma, S. Toppila-Salmi, P. Karisola, H. Alenius, Transcriptomic profiling of adult-onset asthma related to damp and moldy buildings and idiopathic environmental intolerance, International Journal of Molecular Sciences 22 (2021) 10679.

[100] D. Szklarczyk, A. L. Gable, K. C. Nastou, D. Lyon, R. Kirsch, S. Pyysalo, N. T. Doncheva, M. Legeay, T. Fang, P. Bork, L. J. Jensen, C. von Mering, The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets, Nucleic Acids Research 49 (2020) D605–D612.

[101] V. S. Marwah, G. Scala, P. A. S. Kinaret, A. Serra, H. Alenius, V. Fortino, D. Greco, eutopia: solution for omics data preprocessing and analysis, Source code for biology and medicine 14 (2019) 1–12.

[102] J. T. Leek, W. E. Johnson, H. S. Parker, E. J. Fertig, A. E. Jaffe, Y. Zhang, J. D. Storey, L. C. Torres, sva: Surrogate Variable Analysis, 2020. R package version 3.38.0.