The REACT Suite: A Software Toolkit for Microbial *RE*gulon *A*nnotation and Comparative *T*ranscriptomics

Peter Ricke and Thorsten Mascher

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/48040

1. Introduction

The ‘age of omics’ has provided a wealth of genomic and transcriptomic information that is readily available in public databases. In September 2011, GOLD (the Genomes OnLine Database)[1] (Liolios *et al.*, 2010) lists more than 2,600 finished microbial genome sequences and more than twice this number for ongoing and incomplete genome projects, not counting the plethora of metagenome projects, which provide even larger sequence compilations. Comparable numbers of datasets can be retrieved from the two major microarray databases, the Stanford Microarray Database (SMD)[2] (Demeter *et al.*, 2007), and the Gene Expression Omnibus (GEO database)[3] (Barrett *et al.*, 2011) hosted by the National Center for Biotechnology Information (NCBI), which in September 2011 together provide over 9,000 bacterial microarray datasets to the public.

This enormous amount of data provides a treasure chest of information ready to explore. In recent years, a number of powerful comparative genomics databases such as GenoList[4] (Lechat *et al.*, 2008), or MicrobesOnline[5] (Dehal *et al.*, 2010) have provided the community with toolkits to make use of this information.

Combining microarray data with genomic information is a particular powerful approach for identifying and predicting regulons, which are regulatory units consisting of a number of genes or operons under the control of specific transcription factors. Such studies require the

[1] URL for the GOLD database: http://genomesonline.org

[2] URL for the Stanford Microarray database: http://smd.stanford.edu

[3] URL for the Gene Expression Omnibus: http://www.ncbi.nlm.nih.gov/geo/

[4] URL for GenoList: http://genolist.pasteur.fr/

[5] URL for the MicrobesOnline database: http://www.microbesonline.org

identification of co-expressed genes (indicative of co-regulation) from in-depth comparative transcriptome profiling, combined with genomic information, including operon structure, genomic context conservation and the presence of specific regulator binding sites.

The major problem of combining genomic with transcriptomic data to ultimately extract meaningful regulon information is the lack of defined standard formats and software interfaces that allow a direct transfer of data sets derived from transcriptome analyses to comparative genomics databases and vice versa. The REACT suite was developed with the purpose in mind to facilitate such combinations of the different analysis steps outlined above in one intuitive and user-friendly environment. Transcriptome datasets from different sources can be integrated into REACT via a sophisticated import interface and are stored, together with the cognate genomic information, in a MySQL database. This database, together with the central part of the software toolkit and all interlinked third-party tools run on a central computer, which actually performs the analyses: the "REACT-server". It is accessed by the user-interface ("REACT-client") via inter- or intranet. The user will solely work with the corresponding client program, which can be installed on the personal computers or laptops of various users. While the installation of the REACT-server demands some technical knowledge, the client can be run easily on computers with a java runtime environment.

Taken together, the REACT suite provides users with a simple-to-use but powerful bioinformatics environment to perform regulon annotation and comparative genomics analyses based on microarray data and genome sequences. Both server and client software of the REACT suite are freely available from the corresponding author.

2. The basic concept of REACT

REACT was developed to enable users to perform the various steps of expression- and regulon analyses in a quick and intuitive manner. Tools are no longer separated entities demanding different and often incompatible data formats, but can be rather regarded as parts of a comprehensive, fully integrated unit. Data from a wide range of sources can be collected and analysed together. When working with REACT, the user has access to the various representations of the data as well as to the analysis tools via so-called "views" that are intuitively interlinked to enable an interactive flow of both data and analyses:

The "*GeneView*" displays gene-centric information including DNA- and amino acid sequences, links to a number of external databases, as well as the genomic context of a gene in the form of a simple genome browser.

The "*RegulonView*" lists all genes controlled by the same regulator as well as binding motif(s), individual binding sites, alternative promoters etc., based both on the information stored in curated public regulon databases and data added by REACT users.

The "*ArrayView*" allows both importing new microarray datasets and performing data analysis on existing datasets. REACT has a sophisticated interface for the import of array data in nearly every tabulated data format from individual proprietary formats up to GEO/SMD datasets. Data analysis includes one- and two-dimensional scatterplot analyses of signal or ratio-values, as well as gene- and array-clustering with various hierarchical clustering methods, distance methods and correction algorithms (normalization, gene-centering, log-transformation) of all or only selected (collected) subsets of the data.

The "*MotifView*" contains the information of all sequence motifs of known or putative regulator binding sites collected in the current REACT database. Moreover, it enables users to perform MEME analyses to discover new regulatory elements in the upstream regions of selected annotated genes or operons and MAST analyses of previously computed or imported motifs against pre-compiled upstream sequence datasets.

The concept of REACT includes an in-depth integration of the different views via links, enabling users to switch easily between different aspects of the data. Most views are flexible and can be extended with additional data fields to accommodate additional external links, allowing more individualized views and analyses of the data.

Moreover, wherever gene or array data are displayed, the user can easily collect them, thereby creating a data subset available as input for all other implemented analyses. During the various analysis steps, these collections can be continuously changed and expanded, again by selecting single genes and arrays or whole groups of them, such as groups of genes clustering together within a scatter plot analysis. All collected or "marked" arrays and genes are displayed throughout the various views of REACT in form of sortable lists. The items of these lists act as internal links to the corresponding detailed *Array*- or *GeneViews*. Current collections can be saved and opened again for later use, so that the user can easily switch between different data sets any time. In addition, the sequences of the selected genes can also be exported into external FASTA files.

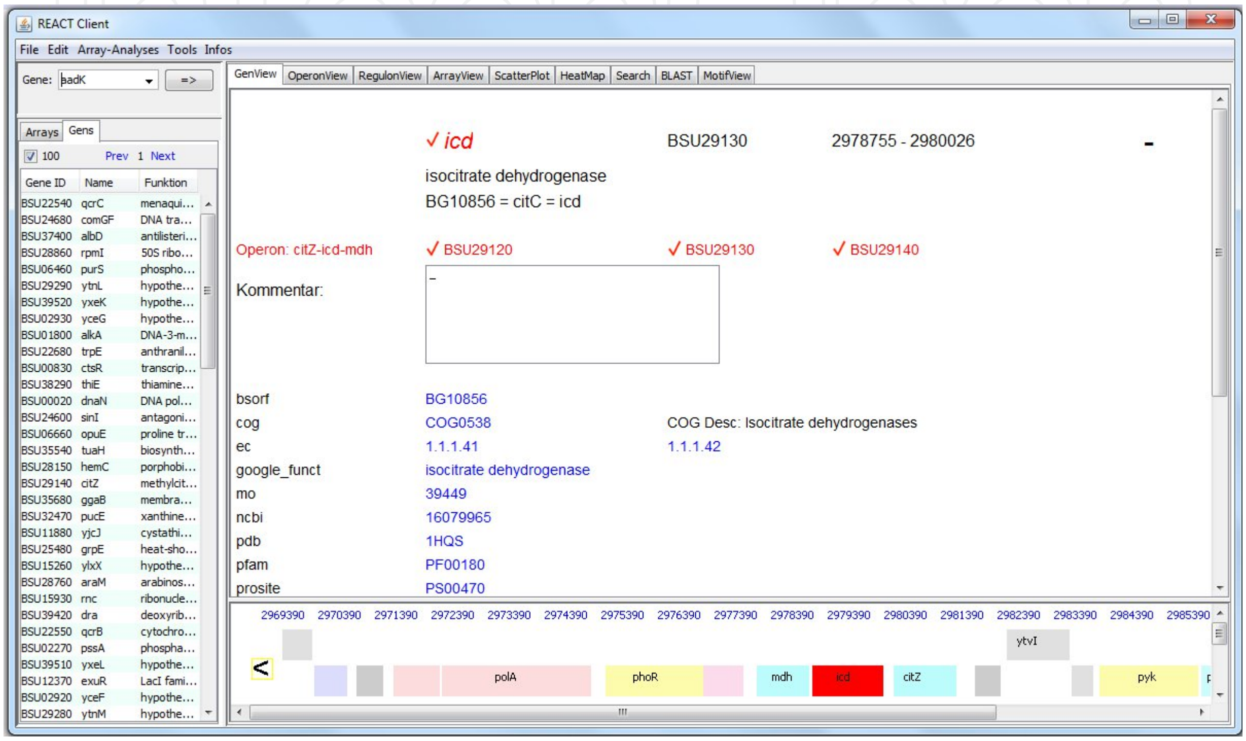
The implemented REACT-databases are organism-specific. In its current version, REACT contains two databases for the model bacteria *Escherichia coli* and *Bacillus subtilis*, but could also be extended to other microbial species. Each REACT-database is based on the detailed genomic data of the model organism, which will be described in the following paragraphs, as well as of an extendable amount of microarray data of this organism. Moreover, basic genomic information on related organisms, the so-called "reference organisms" is also integrated and can be included into some of the analyses. The list of reference genomes can easily be extended, to adjust a given database to the dynamics in genome sequencing.

3. Description of the individual views of the REACT suite

The information stored in REACT databases can be accessed via so-called views that display the data, allow their selection and provide functional links between different types of data for their interactive analysis. In the following sections, we will describe the major views of REACT, to provide an overview of their features.

3.1. The *GeneView*

The "*GeneView*" bundles all available information about individual genes (Fig. 1). On top of the page, the gene identifiers are displayed, accordingly to the existing genomic nomenclature. The first identifier is the gene name, e.g. "*icd*" in case of the *B. subtilis* isocitrate dehydrogenase. If more than one gene name exists for a given gene, the nomenclature applied by REACT is derived from the genome annotation stored in the NCBI genome database[6]. Here, as in other views of the REACT suite, active features working as internal links are highlighted by red letters (Fig. 1). In case of gene names, a double click would bring the user to the corresponding *GeneView*, while a single click would mark the gene (= add it to the gene collection in the left panel) for further analyses.



☑ 100 Prev 1 Next

Gene ID	Name	Funktion
BSU22540	qcrC	menaqui...
BSU24680	comGF	DNA tra...
BSU37400	albD	antilisteri...
BSU28860	rpmI	50S ribo...
BSU06460	purS	phospho...
BSU29290	ytnL	hypothe...
BSU39520	yxeK	hypothe...
BSU02930	yceG	hypothe...
BSU01800	alkA	DNA-3-m...
BSU22680	trpE	anthranil...
BSU00830	ctsR	transcrip...
BSU38290	thiE	thiamine...
BSU00020	dnaN	DNA pol...
BSU24600	sinI	antagoni...
BSU06660	opuE	proline tr...
BSU35540	tuaH	biosynth...
BSU28150	hemC	porphobi...
BSU29140	citZ	methylcit...
BSU35680	ggaB	membra...
BSU32470	pucE	xanthine...
BSU11880	yjcJ	cystathi...
BSU25480	grpE	heat-sho...
BSU15260	ylxX	hypothe...
BSU28760	araM	arabinos...
BSU15930	rnc	ribonucle...
BSU39420	dra	deoxyrib...
BSU22550	qcrB	cytochro...
BSU02270	pssA	phospha...
BSU39510	yxeL	hypothe...
BSU12370	exuR	LacI fami...
BSU02920	yceF	hypothe...
BSU29280	ytnM	hypothe...

✓ icd BSU29130 2978755 - 2980026 -

isocitrate dehydrogenase
BG10856 = citC = icd

Operon: citZ-icd-mdh ✓ BSU29120 ✓ BSU29130 ✓ BSU29140

Kommentar: -

bsorf	BG10856		
cog	COG0538	COG Desc: Isocitrate dehydrogenases	
ec	1.1.1.41	1.1.1.42	
google_funct	isocitrate dehydrogenase		
mo	39449		
ncbi	16079965		
pdb	1HQS		
pfam	PF00180		
prosite	PS00470		

Figure 1. The *GeneView*. Exemplary screenshot for the gene *icd*, encoding the isocitrate dehydrogenase. See text for details.

In addition to the name, each gene has a unique gene-ID or gene number, which consists of an abbreviation for the organism and a number of the gene (based on the chromosomal position). For example, the identifier of the *B. subtilis* isocitrate dehydrogenase is "BSU29130" (Fig. 1). Gene names and numbers are the major identifiers that are used throughout the several displays of the REACT suite. The gene numbers cannot be modified by the users to ensure the integrity of the database. The putative or known functions of the encoded proteins are shown below the gene name, including synonyms and alternative descriptions (if present).

The central part of the *GeneView* are the external links and descriptive data fields. For each genome database implemented in the REACT suite, links to important public databases are already predefined for each gene. This include links to the COGs (Cluster of Orthologous Genes) database[7] (Tatusov *et al.*, 1997), hosted again by the NCBI, the Enzyme Nomenclature

[6] URL for the NCBI genome database: http://www.ncbi.nlm.nih.gov/sites/genome
[7] URL for the COGs database: http://www.ncbi.nlm.nih.gov/**COG**

database[8] (Bairoch, 2000), the already mentioned MicrobesOnline comparative genomics database, the NCBI Protein database[9], the protein data bank PDB[10], a collection of protein structures and structure-related information (Rose *et al.*, 2011), the Pfam[11] (Finn *et al.*, 2010), Prosite[12] (Sigrist *et al.*, 2010) and SMART[13] (Letunic *et al.*, 2009) databases, all of which are dedicated to the definition, maintenance and easy identification of protein domains and families. Further predefined links include a link to Pubmed and to Google. In addition to these general sites, the *GeneView* page also links to organism-specific databases and genome resources, such as BSORF[14] or SubtiList[15] in case of *B. subtilis.*

For all of the above, the links in the REACT databases are gene-specific and directly connect the user with the cognate gene/protein-specific page of the external database. Depending on the type of the external database and the information available for the displayed gene, zero to many external hits will be provided as links. If no such specific database identifier exists, as in the case of Google`s search engine, a gene-related term (e.g. the gene name) has been chosen as the link parameter. REACT is highly adjustable to the individual users' needs. Hence, the external links are not limited to those preimplemented in the existing REACT databases for *E. coli* and *B. subtilis*. (see section 5.3 "Modifying REACT: the administrator mode" for details).

In addition to the links and data fields, the *GeneView* also displays the DNA and amino acid sequences of the current gene, which are linked to the BLASTn and BLASTp tools[16] at NCBI. The user is therefore able to directly search for similar sequences in the public domain. Moreover, a genome browser is implemented at the bottom of the *GeneView* for a quick glance on the genomic environment of the current gene (Fig. 1). The gene icons are coloured according to the COG-functional classes assigned to each gene and serve as links to the corresponding *GeneViews*.

Two additional functions are available in the *GeneView*. First, the user can retrieve the upstream genomic region of the gene via a specific dialog box, based on user-provided information, such as upstream region length, inclusion of start codon, or choice between the upstream region of the current gene or the first gene of its operon. The latter function is very useful for collecting upstream regions for motif searches (see section 3.5). Second, expression data of the active gene can directly be retrieved from the REACT microarray database. For this, the user can choose either all or only selected microarray datasets, and limit the set of extracted values by a certain threshold expression ratio level. The *GeneView* is therefore not only the central platform for all gene-centric data, but is also directly linked to all other views described in the following sections.

[8] URL for the ENZYME database: http://enzyme.expasy.org/enzyme_ref.html
[9] URL for the NCBI Protein database: http://www.ncbi.nlm.nih.gov/protein
[10] URL for PDB: http://www.rcsb.org/
[11] URL for the Pfam database: http://pfam.sanger.ac.uk/
[12] URL for the Prosite database: http://prosite.expasy.org/
[13] URL for the SMART database: http://smart.embl-heidelberg.de/
[14] URL for the BSORF site: http://bacillus.genome.ad.jp/
[15] URL for SubtiList: http://genolist.pasteur.fr/SubtiList/
[16] URL for BLASTn and BLASTp: http://blast.ncbi.nlm.nih.gov/Blast.cgi

3.2. The *OperonView*

Operons are transcriptional units consisting of two or more neighbouring genes that are co-expressed. If a gene has been assigned to an operon and annotated accordingly in the REACT database, a link from the *GeneView* leads to the *OperonView*. Both views are organized in a similar fashion and the *OperonView* also contains a genome browser. It is identical to the *GeneView*'s with the exception that here the current operon is highlighted.

The operon identifier is again immutable, since it is used by REACT as internal reference. The operon name by default consists of the concatenated names of the genes within this operon. When displayed outside of the *OperonView*, it functions as an internal link, enabling the user to jump directly to the corresponding *OperonView*. From within the *OperonView*, it can be used as a link to an external database containing additional information regarding this operon.

In addition to providing a direct link to all corresponding *GeneView* pages, the *OperonView* also provides a list of and links to all regulons, to which the current operon belongs. They are represented by their REACT-internal regulon identifier, the name of the corresponding transcription factor and a brief description of the regulon (see the following section for details).

3.3. The *RegulonView*

The next higher level of genetic units is the regulon, which consists of a number of genes or operons under the direct control of a specific transcription factor. Regulons are displayed within the REACT suite in the *RegulonView*, which is subdivided into two panels. The first section ("All regulons") contains a tabulated list of all regulons currently defined in the REACT database, which includes the most important information such as the regulon-ID, the main description of the regulon and the associated transcription factor. The second part displays the detailed view of a specific regulon selected from the first list ("Act. regulon"). This view is organized similar to the *GeneView* or *OperonView*. It contains regulon identifier, a link to the corresponding transcription factor (if known) and the sequence motif of its cognate DNA-binding site, which is found upstream of the regulated operons or genes comprising this regulon.

The regulon-ID is derived from the gene-ID of the corresponding transcription factor and marked by the extension "_R". It is implemented as an active link that directly connects to an external regulon database. In case of *B. subtilis*, this is primarily DBTBS[17], the database of transcriptional regulation in *B. subtilis* (Sierro *et al.*, 2008), but BSORF or SubtiList have also been used for the initial regulon annotation. For *E. coli*, the regulon information has been extracted from RegulonDB[18] (Gama-Castro *et al.*, 2011). Additional regulon definitions can be added at any time, including putative regulons with only rudimentary information, such

[17] URL for the DBTBS database: http://dbtbs.hgc.jp/

[18] URL for RegulonDB: http://regulondb.ccg.unam.mx/

as sets of co-expressed genes. If the regulator DNA-binding site is known and defined, it will also be displayed as a so-called SequenceLogo (see section 3.5 for details). Below the general regulon-associated information and, if available, the sequence motif overview, additional data fields and external links are displayed.

The central part of the *RegulonView* provides individual information on all associated operons and genes, including the name, the first gene in case of operons, the corresponding σ factor and the position and sequence of the putative binding site in front of the regulated transcriptional units. This information enables the user to get a quick first impression of the regulated genes of this regulon.

3.4. The *ArrayView*

All three views explained so far are highly similar to one another and strongly integrated, not only regarding the information provided but also in the way the user can navigate from one view to the next. They all provide a gene-centric view on the REACT data and invariantly rely on the genomic sequence as a reference. Regulons consist of operons, which are made up of individual genes with a defined position on the chromosome. The same is true for regulator binding sites.

In contrast, the *ArrayView* provides access to the second central data pool stored within the REACT database, the microarray datasets. Array data exist in a great variety of different formats. Especially data sets from the early years of transcriptome studies often are available only in form of simple tables or excel spread sheets without any defined data format, distributed over numerous journal homepages or webpages of individual research groups, making their implementation into comparative transcriptome analysis very difficult. As a result, public databases, such as the already mentioned GEO or SMD, have been developed for the storage and description of microarray datasets that comply to the MIAME (Minimal Information About a Microarray Experiment)[19] standard (Brazma *et al.*, 2001). Unfortunately, these databases still contain only a fraction of the published microarray datasets. The biggest challenge for a comparative transcriptome database is therefore to organize and import microarray data from diverse sources into a compatible format.

3.4.1. Organizing microarray data in the REACT suite

A complete microarray dataset contains at least three types of information. (i) A list of all genes represented by a given DNA microarray, which is linked to the corresponding expression values, either expressed as (ii) raw fluorescence values for the reference and experimental condition, or as (iii) the respective ratio (or fold-change) between the two conditions. Within the REACT suite, such a data collection is called an "Array". Obviously, the Array is only useful if additional descriptive information (meta-information) is available. This can be a short description of the specific experimental set-up or a link where this information is stored. Often, a group of array datasets are related to each other and

[19] URL for the description of the MIAME standard: http://www.mged.org/Workgroups/MIAME/miame.html

described in a single format, e.g. as a result of one experiment. This is reflected by the REACT data format “Array Set”.

The *ArrayView* is split into four sub-views. The first sub-view, “All Arrays” contains a tabulated view of all array sets of the current REACT database. If one array set is selected, all arrays of this set are displayed below the upper window, again in tabulated format. Both tables contain some basic meta-information on either the array or array set, respectively.

Selection of one array or array set leads to the next sub-view “Act. Array”, which provides the detailed information, including the ID, name, a description of the underlying experiment, the source of the data, available literature, and external links. The “Array Set” subview lists all individual arrays within the set, which can be marked separately for further analysis. The most important feature of this sub-view is a tabulated, sortable list of all genes, for which data are available within this array. It contains information on the gene name, the signal value, the control value, the ratio of signal to control, the number of replicates that were combined, the arithmetic mean and error of the values. This data is normally directly derived from the original data sets. Two additional columns indicate which genes are currently marked and if their value can be trusted. The trust value is a simple way to allow users to flag single values as untrusted, thereby automatically excluding them from subsequent analyses. Trust values can be easily set for marked genes within the current subview.

The data table is sortable based on any column, e.g. high or low signals or ratio values. Genes of interest can be collected as “marked genes” for inclusion into follow-up analyses. Each gene-specific data row of the table functions as an internal link to the corresponding *GeneView*, thereby providing a direct connection between the array-centric data of the *ArrayView* and the gene-centric data of the *Gene/Operon/RegulonViews*.

An additional feature of the *ArrayView* is the “Similar gene” function. For each array displayed in the dialog, the user can define ratio-thresholds similar to the ratio of the current gene. REACT then automatically retrieves a list of all genes fulfilling the user defined criteria. These genes can then be marked for subsequent analyses. The “Similar gene” function therefore provides a simple but efficient and direct way to find genes with similar expression characteristics from the available microarray database.

3.4.2. Importing microarray data into the REACT database

As already mentioned, one of the major problems in comparative transcriptome analyses is the lack of a mandatory gold standard for array datasets, especially from the early, pre-MIAME era. But even ten years after this standard has been introduced, this problem is still far from be solved, and the number of microarray datasets not complying to these standards is still rising (Brazma, 2009).

Even implementing the minimum amount of information needed to integrate an array data set into the REACT database – a two-column table, with one column containing the gene identifiers and the second containing either the signal values or expression ratios between signal and control – can be daunting. Gene identifiers are either not used consistently (as synonyms often exist), or the DNA microarray might not contain all genes, or duplication of some. Likewise, signals can be represented as raw fluorescence values, either as mean or average values, in which case control values need to be provided or defined. Alternatively, a table might provide ratio of signal to control, which can be either expressed as log-values or as fold-changes. To facilitate handling and import such diverse types of data, the REACT suite contains an easy-to-use microarray import interface (Fig. 3).

During import, microarray data in any tabulated format is initially pre-loaded into the REACT import panel. REACT automatically detects the number of columns in the file and generates an adequate number of numerated preview columns for easy identification. After semiautomated discrimination of commentary lines, the appropriate type of information has to be assigned to each column. REACT needs at least one column containing the gene identifiers and one column for the signal or ratio values. Other types of information can also be assigned, such as the signal background, the control value, and the control background. Based on the assignment, REACT 'knows' what to do with the individual data, e.g. if background columns are specified, their values will be subtracted from the corresponding signal or control values. Ratios between signal and control values can either be directly imported or will be calculated, depending on the data provided. It is even possible to import data with only a single column containing the signal values (e.g. during time course experiments). In a later step, one of the imported arrays (e.g. time 0) can then be used as a standard control for all datasets to calculate the ratio values needed for most analyses. Large datasets containing many replicates of one experiment can be imported in a single table. In this case, REACT offers the possibility to average the sets of columns assigned for signal, control or ratio values.

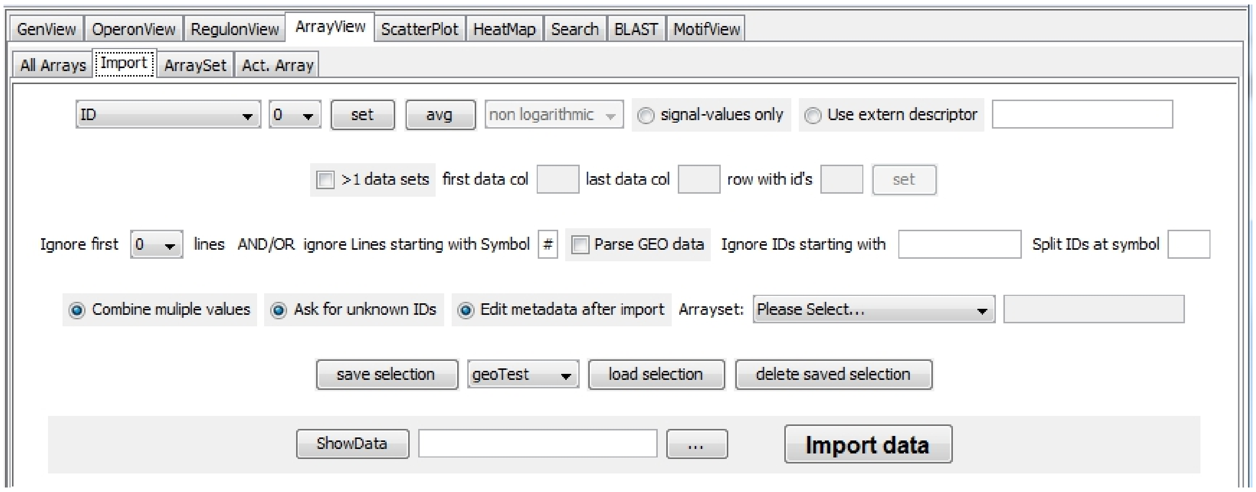


Figure 2. The microarray import interface of the *ArrayView*. See text for details.

If large numbers of different experiments are stored in a single table, they can be parsed at once using the "batch"-import. The user defines the different ratio-columns, and each column will be treated as a separate array, within a common array set. Moreover, it is possible to define, if ratio data are in logarithmic format (they will be converted to internal non-logarithmic values) or not.

One major challenge when comparing data from different sources and hence formats is dealing with variations and differences in the gene identifiers used in different microarray templates. REACT knows a large amount of different gene descriptions, as mentioned in section 3.1. During data import, REACT will accept any of these names and synonyms. But if unknown identifiers occur or synonyms have been assigned more than once in a microarray dataset (e.g. in case of different probes representing a single gene), REACT will ask the user for a specific decision. The user can then skip/delete the line, manually assign a gene name, or add the new synonym to the database for future use.

Taken together, REACT should be able to import virtually all formats of array data, as long as they are tabulated. For the more complex datasets, such as those generated by the GEO, special parsing options for the corresponding meta-information are available in REACT.

3.5. The *MotifView*

The *MotifView* is divided into five sub-sections, three of which (the "Upstream"-panel, the "Act. Motif" and the "MotifTable") are used for displaying the data and will be described here. The remaining two – MEME- and MAST-panel – are interfaces for the eponymous external analysis tools and will be discussed in more detail later (section 4.4).

The "Upstream"-panel is used to collect and display DNA regions upstream of coding sequences. Mostly, this will be intergenic regions, which are of particular interest, since they contain both (alternative) promoters and putative DNA-binding sequences of transcriptional regulators. The possibility to retrieve and manage such upstream regions is therefore of crucial importance in the context of regulon analyses. Upstream regions can be added to the "Upstream"-table by one of three means: (i) collectively from the active list of marked genes, (ii) individually by gene name, or (iii) directly from within the *GeneView* for the corresponding gene. In all cases, the user can define parameters for the retrieval, such as the sequence length, inclusion of sequence up to the upstream stop codon or exclusion of sequences of upstream genes, in the case that they overlap with the selected upstream sequence length.

The "Upstream"-table displays all upstream regions collected by the user in the course of an analysis by any of the three methods described above. For each upstream region, the ID and name of the corresponding gene, and the sequence and position of the respective region in the genome are displayed. These regions (or subsets thereof) can easily be removed or added, exported as FASTA-formatted sequence files or selected for further analyses, such as the MEME/MAST analyses (see section 4.4).

In the context of regulon analyses performed within the REACT suite, motifs are defined as short stretches of nucleotide sequence that are conserved in a collection of upstream regions, derived e.g. from co-expressed genes. They are expressed as so-called position-specific scoring matrices (PSSMs, also known as Position Weight Matrices, PWM) or regular expressions (RE), which both describe the probability for specific bases to occur at a specific position of the motif. Such matrices are graphically displayed as so-called "SequenceLogos", in which the height of the letters representing the four bases is a measure for the degree of conservation at any given position within the motif.

In REACT, defined motifs of known regulator-binding sites are stored in the "MotifTable". In this table, each motif is represented by the REACT-internal ID, the name of the motif (normally equivalent with the name of its cognate regulator), the motif length, the associated regular expression or PSSM, as well as the corresponding SequenceLogo. Selection of a motif opens the "Act. Motif"-panel, which provides all available information of one motif, including the name of the regulon it is associated with. This regulon name serves as an internal link to the corresponding page within the *RegulonView*. Moreover, a multiple sequence alignment of all (upstream) sequences underlying this motif is shown (if available), which can be exported as FASTA format.

4. Search options and analysis tools within the REACT suite

So far, this book chapter has described the major views that represent and display the data stored within the organism-specific REACT databases. In the following sections, we will describe the tools that allow the user to search the database and analyse genes, motifs, and microarray datasets in order to extract and define regulons. These tools include a search engine, an internal BLAST tool, cluster analysis and scatter blot tools for microarray datasets, as well as the MEME/MAST algorithms to identify and search for regulator binding sites in upstream regions of co-expressed genes.

4.1. The *Search* tool

The wealth of information stored in the REACT databases requires search tools to find specific data sets. The REACT Search tool contains four panels, enabling the user to search for genes, regulons, arrays and array-sets, respectively. These panels share the same general structure and differ only in minor features. The common features will be described for the gene search panel (Fig. 3).

Genes of interest can be searched by all gene-specific data fields, e.g. by gene-ID, name, synonyms, function, comments, but also any other user-defined field. These fields can be searched by a number of search strings, such as <containing>, <being equal to> or <starting with> a certain term. After the search hits are displayed in tabulated form in a result window below the search panel, where they can be marked or used as internal links to the respective views. Consecutive searches can be combined by <add>, <remove>, <keep> results or <negate> operations, thus enabling even for more sophisticated searches.

The search functions introduced so far are available in all four search panels. For genes and arrays, an additional function allows searching marked genes or arrays, respectively. Moreover, genes can also be successively searched by COG categories and COG terms.

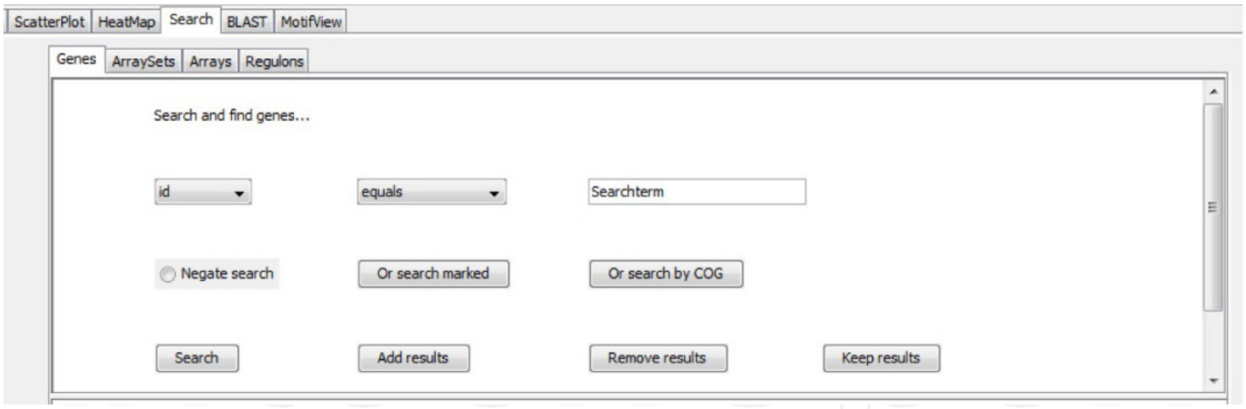


Figure 3. The *Search* tool of REACT, exemplified by the search panel for genes.

4.2. The internal *BLAST* tool

Within the REACT suite, BLAST analyses (Altschul *et al.*, 1990) can be performed in two different ways. First, it can be performed from within the *GeneView* via a direct external link to the NCBI BLAST server (see section 3.1). Second, REACT also provides an internal BLAST search, which allows comparing a gene of interest with the internal reference genomes of the corresponding REACT database. This internal search, which can be accessed by the corresponding BLAST panel, allows retrieving not only the homologous gene or protein sequences, but also the corresponding upstream regions for further analyses, such as MEME/MAST (see section 4.4).

Both external (pasted into the input window) and internal (derived from the gene/protein displayed in the current *GeneView*) sequences can be used as query, either as DNA or protein sequence. After choosing the appropriate BLAST algorithm and the sequence data to be analysed, the results are displayed in tabulated form in the corresponding panel. For each match, both gene-specific information (ID, name, function, organism) and BLAST-specific values (E-value, per cent identity, match length, number of mismatches/insertions/deletions) are displayed. Moreover, the genomic context is illustrated in a genome browser.

For each match, the DNA or amino acid sequence can be retrieved. Moreover, REACT also provides access to the corresponding upstream region via the “Retrieve upstream” function. The corresponding sequences will then be added to the “Upstream”-panel of the *MotifView* as described above (see section 3.5).

4.3. Microarray analysis tools

As mentioned before, the REACT suite is based on organism-specific databases that contain two types of data. The gene-centric data is derived from public genome sequence information and accessible through the *Gene-*, *Operon-*, *Regulon-*, and *MotifView*, while the array-centric data is displayed in the *ArrayView*. Two different types of tools have been implemented into the REACT suite in order to analyse this second type of data: (i) Scatter plot analyses (4.3.1) allow the comparison of up to two experimental conditions, while Cluster analyses (4.3.2) are used to extract expression values from multi-array comparisons.

4.3.1. The scatterplot tool

A scatter plot is a graphical way to project values for two variables of a data set into a two-dimensional grid, thereby placing similar samples in the same regions of the grid. The data is displayed as a collection of points, each having the value of one variable determining the position on the *x* axis and the value of the other variable determining the position on the *y* axis (Utts, 2005). A scatter plot is a very useful tool to identify similarities and differences in large, comparable datasets that agree in large parts with each other. The more the two data sets agree, the more the scatter tends to concentrate in the vicinity of a so-called identity line, where $y = x$.

Within REACT, scatter plot analyses are normally used to display genes according to their expression data of two selected arrays, using the expression value of the first array as *x* coordinate and the values of the other as *y* coordinate. This representation of the data results in an interactive panel where genes with similar expression patterns are grouped together.

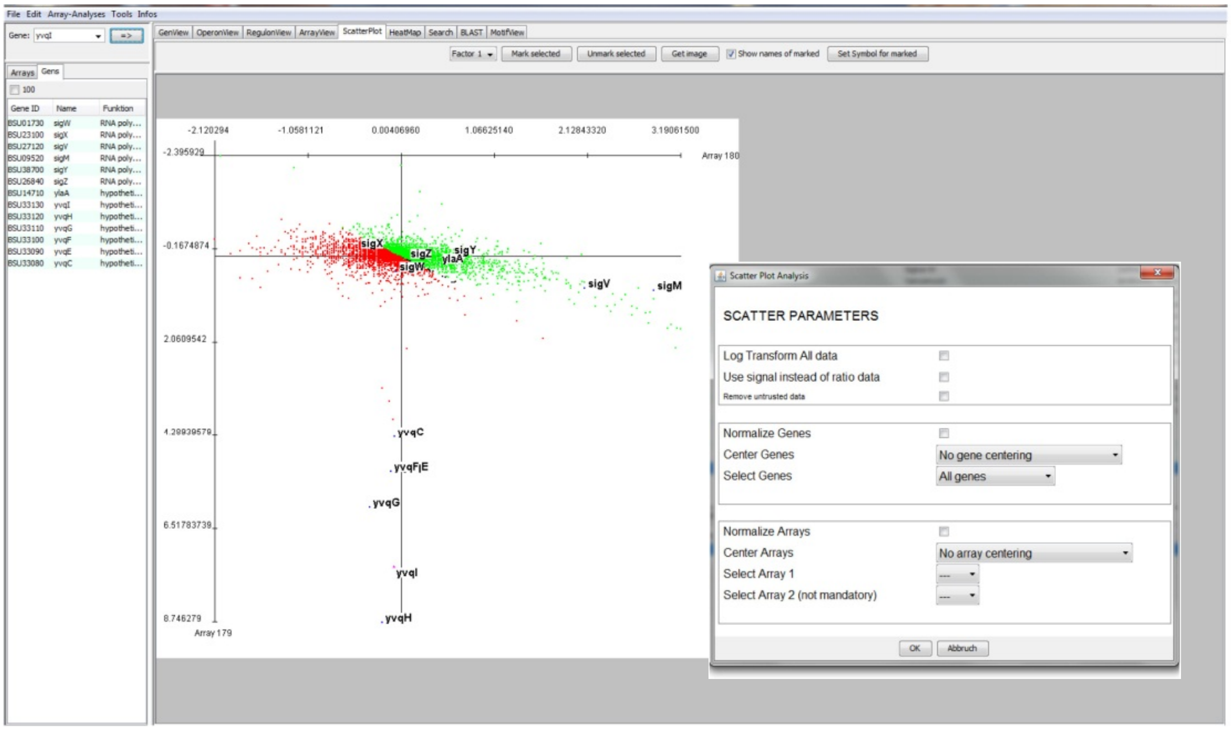
In most cases, the vast majority of analysed genes should show the same expression values/ratios under both conditions and will therefore be placed closely together on the x=y line. In contrast, genes that differ significantly in their behaviour between the two conditions will appear as outliers and can therefore be easily identified in the plot. Of course, comparisons of array datasets from different research groups tend to deviate more or less significantly from this ideal situation. Hence, the differences in experimental conditions need to be kept in mind when comparing array data sets.

Scatter plot analyses can be performed using either signal or ratio array values, thereby allowing to compare the behaviour of genes in the presence of different stimuli (ratio data), but also to compare different time points from one time course experiment (using signal data). Such comparisons of expression data from two different microarrays are called two-dimensional scatter plots (see Fig. 4 for an example).

But the user can also compare the data of one array against itself, using the same signal or ratio values as coordinates for the *x* and *y* axes. As this results in the placement of all genes on one line (the identity line), it is called a one-dimensional scatter plot. Such an analysis can be helpful to verify that a group of related genes (e.g. from one operon) behaves in a similar fashion within one experiment.

The input (expression data) for both types of analyses can either be log-transformed or normalized for the arrays or for the genes (array- and gene-centering, respectively). Moreover, the data can be filtered to remove "untrusted" genes prior to the analysis. Here, REACT removes all genes previously flagged as untrusted and un-reliable (either automatically during the import or later by the user) in one or both array datasets.

The major advantage over using external standard scatter plot tools is the deep integration of the REACT scatter plots with the REACT database. Without pre-selection of genes, the analysis will be carried out with the complete microarray data sets. Genes that specifically respond to only one of the two conditions will appear as outliers and can then be easily selected directly from the plot and thereby added to the list of marked genes directly for further analyses within REACT. This deep integration and direct connection of array-centric results with gene-centric information is one of the major strengths of the REACT suite, which enables the user to efficiently analyse even complex datasets.



☐ 100

Gene ID	Name	Funktion
BSU01730	sigW	RNA poly...
BSU23100	sigX	RNA poly...
BSU27120	sigV	RNA poly...
BSU09520	sigM	RNA poly...
BSU38700	sigY	RNA poly...
BSU26840	sigZ	RNA poly...
BSU14710	ylaA	hypotheti...
BSU33130	yvqI	hypotheti...
BSU33120	yvqH	hypotheti...
BSU33110	yvqG	hypotheti...
BSU33100	yvqF	hypotheti...
BSU33090	yvqE	hypotheti...
BSU33080	yvqC	hypotheti...

SCATTER PARAMETERS

Log Transform All data	☐
Use signal instead of ratio data	☐
Remove untrusted data	☐
Normalize Genes	☐
Center Genes	No gene centering
Select Genes	All genes
Normalize Arrays	☐
Center Arrays	No array centering
Select Array 1	---
Select Array 2 (not mandatory)	---

Figure 4. Example of a two-dimensional scatter plot analysis with labelled genes. The inset on the right shows the parameter window for choosing the settings for a scatter plot analysis.

But scatter plots can also be performed on only a small group of genes collected in previous analyses, thereby enabling the user to focus on a relevant subset of the data. The second approach is for example useful if these genes are known or suspected to belong to one regulon, in which case they should show a similar behaviour under various conditions. Two-dimensional scatter plots provide an easy way to test this hypothesis, since currently marked genes can be labelled in the plot and thereby easily visualized (Fig. 4).

Images of the scatter-plots can be directly retrieved. For presentation or publication purposes, individual genes can be labelled with their names, or specific symbols can be assigned to groups of genes, in order to distinguish them.

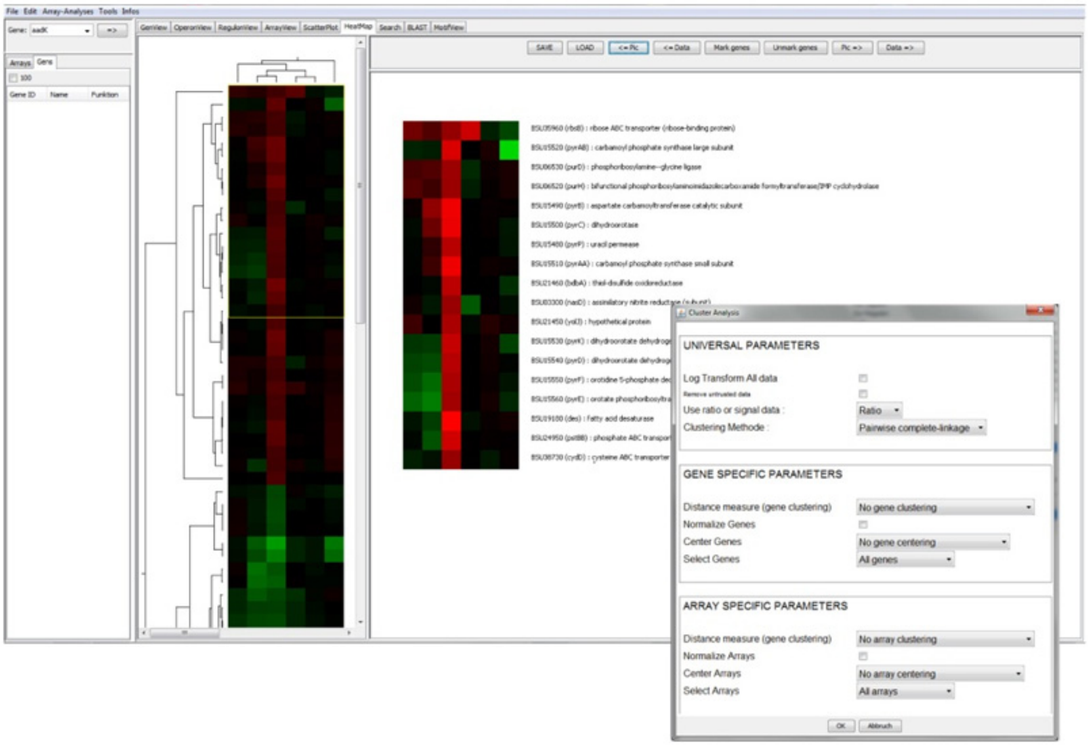
4.3.2. The cluster analysis (HeatMap) tool

To perform more sophisticated expression analyses of multiple microarray datasets, the hierarchical clustering functions of the Cluster 3.0 Software (de Hoon *et al.*, 2004), an enhanced version of the Cluster Software[20], were integrated into REACT. This analysis assigns sets of genes into groups (the so-called clusters), so that the behaviour of the genes

[20] URL for the source code of the Cluster software: http://rana.lbl.gov/EisenSoftwareSource.htm

from within the same cluster is more similar to each other than to those in other clusters. Clustering is based on calculating a distance measure, which determines the similarity of two elements. During this calculation, the often *n*-dimensional data sets are reduced to their respective distances (one distance for each pair of objects). This less complex data set is then used as input for the final clustering. The corresponding algorithms achieving this differ significantly in their notion of what constitutes a cluster and in their efficiency of finding them.

The hierarchical cluster analyses embedded in the REACT suite provide a way to compare the expression behaviour of genes over multiple microarray datasets but also, if needed, to group and cluster arrays. The result is a two-dimensional, colour-coded matrix (or grid) in which each row represents one gene, while each column corresponds to one array dataset. Rows and/or columns are sorted according to their overall distances, and this clustering is illustrated by flanking distance trees, in which the length of the branches serves as a measure for similarity: The shorter the branches, the higher their similarity (Fig. 5).


File Edit Array-Analyses Tools Infos
Gene: aadK
Arrays Gene
Gene ID Name Funktion
GeneView OperonView RegulonView ArrayView ScatterPlot HeatMap Search BLAST MotifView
SAVE
LOAD
<= Pic
<= Data
Mark genes
Unmark genes
Pic =>
Data =>
BSU35960 (rbsB) : ribose ABC transporter (ribose-binding protein)
BSU15520 (pyrAB) : carbamoyl phosphate synthase large subunit
BSU06530 (purD) : phosphoribosylamine--glycine ligase
BSU06520 (purH) : bifunctional phosphoribosylaminoimidazolecarboxamide formyltransferase/IMP cyclohydrolase
BSU15490 (pyrB) : aspartate carbamoyltransferase catalytic subunit
BSU15500 (pyrC) : dihydroorotase
BSU15480 (pyrP) : uracil permease
BSU15510 (pyrAA) : carbamoyl phosphate synthase small subunit
BSU21460 (bdbA) : thiol-disulfide oxidoreductase
BSU03300 (nasD) : assimilatory nitrite reductase (subunit)
BSU21450 (yolJ) : hypothetical protein
BSU15530 (pyrK) : dihydroorotate dehydrog
BSU15540 (pyrD) : dihydroorotate dehydrog
BSU15550 (pyrF) : orotidine 5-phosphate dec
BSU15560 (pyrE) : orotate phosphoribosyltra
BSU19180 (des) : fatty acid desaturase
BSU24950 (pstBB) : phosphate ABC transpor
BSU38730 (cydD) : cysteine ABC transporter
Cluster Analysis
OK
Abbruch

UNIVERSAL PARAMETERS

Log Transform All data □
Remove untreated data □
Use ratio or signal data : Ratio
Clustering Methode : Pairwise complete-linkage

GENE SPECIFIC PARAMETERS

Distance measure (gene clustering) No gene clustering
Normalize Genes □
Center Genes No gene centering
Select Genes All genes

ARRAY SPECIFIC PARAMETERS

Distance measure (gene clustering) No array clustering
Normalize Arrays □
Center Arrays No array centering
Select Arrays All arrays

Figure 5. Example of a cluster analysis performed with the *HeatMap* tool. The inset on the right shows the parameter window for choosing the settings for a cluster analysis.

The complexity of the data is not lost, as all ratio or signal values for each gene within all arrays are visualized by the colour of the individual cells within the heat map grid. When ratio values are displayed, green colour indicates an increase (positive ratio value) and red a decrease (negative ratio value) of the expression in comparison to the control condition of the array, while the intensity of the colour is an indicator for the magnitude of change (Fig. 5). Signal values are coloured according to their percentage from the lowest and highest measured value within the array.

To run a cluster analysis, the user has first to decide, which genes and microarray datasets are to be included. Again, the active list of marked genes/arrays can directly be applied. Since REACT only serves as an interface to the Cluster 3.0 software package, its panel mimics the original input fields, with some modifications (inset to Fig. 5). The choice of parameters includes: clustering of only genes, arrays, or both; (ii) use of ratio or signal values; (iii) log-transformation of the data; (iv) removal of "untrusted" data (see above). Distance measures such as Euklidian distance, Kendall's tau, Pearson correlation or Spearman's rank correlation are available for both gene- and array-clustering. Moreover, genes and arrays can again be normalized as well as centered, as described for the scatter plot analysis above. For the final linkage, the user can choose between Pairwise Single, Pairwise Complete, Pairwise Centroide and Pairwise Average Linkage clustering methods[21].

The results of the cluster analysis are displayed in the *HeatMap* window (Fig. 5). This view is vertically split into two subpanels. The left panel displays the complete heat map, including the distance trees, while the right subpanel displays selected areas in more detail, including gene-IDs, names and descriptions. The heat map is interactive and selecting one row will directly open the corresponding *GeneView*. Marked genes are highlighted in red in the heat map.

To further analyse a certain gene cluster, it can directly be selected from the flanking distance trees, which are also interactive: Selecting any branch will mark the corresponding rows or columns. Intersections of selected rows and columns can be obtained and selected parts of the heat map can be displayed in higher resolution in the right subpanel of the *HeatMap* window, as described above.

The content of each subpanel can be exported both as an image file (different file formats can be chosen), as well as in tabulated form. Cluster results can also be stored and reloaded again, e.g. to enable the user to compare the clustering of specific groups of genes between different analyses.

4.3.3. The "Show regulons of marked genes" function

Co-expression – and therefore co-clustering – of groups of genes is a strong indication that they presumably belong to one regulon, i.e. are under the direct control of a common transcriptional regulator. In case of the two model bacteria currently implemented in the REACT suite, *B. subtilis* and *E. coli*, many of these regulons are already known.

To simplify the identification of known regulons within a marked group of genes derived from one of the above analyses, the "Show regulons of marked genes" function was implemented in the REACT suite, which displays all regulons to which at least one currently marked gene is associated in an additional window. Moreover, the results window will also list all operons and genes of any identified regulon, thereby providing a direct overview of the coverage of a given regulon within the group of marked genes identified by the cluster

[21] For details on clustering, see: http://bonsai.hgc.jp/~mdehoon/software/cluster/manual/Hierarchical.html#Hierarchical

analysis. As usual, the identifiers of the regulons, operons and genes function as internal links to the corresponding views, enabling a seamless integration with subsequent analyses of the identified transcriptional units.

This function therefore offers a very straightforward and easy-to-use approach to identify the regulators responsible for an observed co-expression of a group of genes.

4.4. Motif analysis tools

If the above mentioned function did not yield a direct insight into regulatory principles underlying an observed co-expression, the next step of a typical analysis would be to search for putative regulator binding sites in the upstream genomic regions of co-expressed genes and operons. To facilitate these analyses, the MEME/MAST tools from the MEME (Multiple EM for Motif Elicitation) suite[22] (Bailey *et al.*, 2009) were incorporated into REACT. MEME allows the identification of short overrepresented sequence motifs in a group of unaligned sequences of different length. MAST is a sequence similarity search algorithm that utilizes motifs either provided by the user or from a previous MEME analysis, to search for similar motifs in genome sequences. Starting from the upstream regions of co-clustering genes, these two tools, if applied in combination, often allow to identify putative regulator binding site in novel regulons.

4.4.1. The MEME-Analysis tool

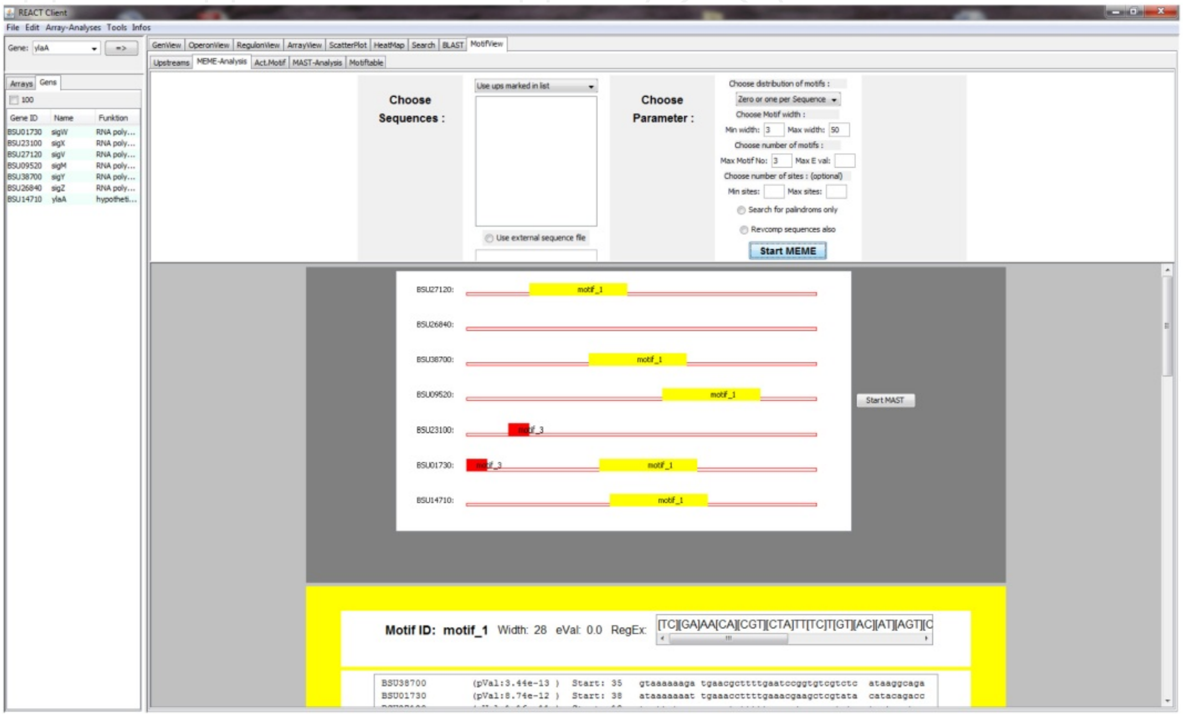
A prerequisite for any motif search is a collection of (upstream) sequences that are supposed to contain a common motif. In the REACT suite, this is facilitated by the “get upstream” function, which can be found in a number of views, including the *Gene-, Operon-* or *MotifView.* The latter also contains the panels for the MEME and MAST analyses. Again, REACT's motif discovery function is just an embedded interface to these freely available and well established tools, which are components of the MEME suite. MEME is a tool for discovering motifs in a group of related DNA or protein sequences. It represents motifs as position-specific scoring matrices (PSSM's), which describe the probability of each possible letter at each position within the gapless pattern. MEME uses statistical modelling techniques to automatically choose the best width, number of occurrences, and description for each motif to reduce the number of false-positive hits. Nevertheless, they can occur incidentally, especially if the motifs are very short, and therefore have to be validated experimentally, both *in vivo* and *in vitro* (Cao *et al.*, 2002).

Like other analysis panels of REACT, the MEME view is also divided into two areas: in the upper part, the sequences and analysis parameters can be specified, while the results will be displayed in the lower panel (Fig. 6).

To start a MEME analysis, the user has to provide the sequences (in this case: upstream regions of genes), which are believed to share a common motif. This can be done by one of three ways: (i) Selection of upstream sequences from the “Upstream sequence” panel, (ii)

[22] URL for online access to the complete MEME suite at: http://meme.nbcr.net

directly pasting sequences into the respective sequence window of the MEME interface, or (iii) uploading an external file. The latter options enable the inclusion of sequences, which are not derived from the REACT database. Next, the number of allowed (or expected) motifs per sequence needs to be defined. Additional parameters include (i) the minimum and maximum motif-width, (ii) the maximum number of motifs to be discovered, (iii) a statistical threshold value, and (iv) limitation to palindromic sequences.



Gene: ylaA

☐ 100

Gene ID	Name	Funktion
BSU01730	sigW	RNA poly...
BSU23100	sigX	RNA poly...
BSU27120	sigV	RNA poly...
BSU09520	sigM	RNA poly...
BSU38700	sigY	RNA poly...
BSU26840	sigZ	RNA poly...
BSU14710	ylaA	hypotheti...

Use ups marked in list

○ Use external sequence file

Choose distribution of motifs : Zero or one per Sequence

Choose Motif width :
Min width: 3 Max width: 50

Choose number of motifs :
Max Motif No: 3 Max E val:

Choose number of sites : (optional)
Min sites: Max sites:

○ Search for palindroms only

○ Revcomp sequences also

Motif ID: motif_1 Width: 28 eVal: 0.0 RegEx: [TC][GA]AA[CA][CGT][CTA]TT[TC]T[GT][AC][AT][AGT][C

```
BSU38700        (pVal:3.44e-13 )  Start: 35   gtaaaaaaga tgaacgcttttgaatccggtgtcgtctc  ataaggcaga
BSU01730        (pVal:8.74e-12 )  Start: 38   ataaaaaaat tgaaaccttttgaaacgaagctcgtata  catacagacc
```

Figure 6. The MEME analysis interface embedded in the *MotifView*.

REACT`s MEME results consist of a graphical overview of the analysed sequences (Fig. 6) illustrating the occurrence and position of the motifs. Each motif is described by the following information: a motif ID, the length of the motif, a statistical value as a measure for the reliability of the motif, and a corresponding SequenceLogo as a graphical representation of the motif. As computable definitions, the description also includes the Regular Expression, an alignment of the motif from the analysed sequences, and the PSSM, which can all be exported. Alternatively, these definitions can be used directly for a MAST analysis to screen genome sequences from the REACT database for additional upstream regions containing this pattern (described in the following section) or stored in the REACT database for later analyses.

4.4.2. *The MAST-Analysis tool*

An important strategy to identify regulon members in large datasets, such as (multiple) genome sequences, is to screen them for the presence of sequence motifs, especially in intergenic regions, that are known or postulated to function as regulator binding sites. Such patterns can be derived from known operator sites described in other, closely related organisms (Wecke *et al.*, 2006), or from motifs identified by MEME analyses from collections of co-expressed loci, as described above. One way of testing predicted motifs *in silico* is to apply them to larger data sets. This not only allows the identification of additional putative matches, but it might also help to improve the motif through iterations. In the REACT suite, this can be done with the MAST analysis interface.

MAST is a tool for searching biological sequence databases for sequences that contain one or more copies of a known motif. The quality of a resulting hit is calculated as the strength of the similarity of the particular sequence to all motifs, based on statistical probabilities. MAST works by calculating match scores for each sequence in the database compared with each of the provided motifs. These initial scores are then converted into statistical probability values, which are used to determine the overall match of the sequence to the group of motifs. By this approach, the best fitting sequences in the analysed data set can ultimately be identified.

The MAST interface of the REACT suite is located within the *MotifView* and resembles the one for the MEME analysis both in appearance and overall logic. Two important parameters need to be defined by the user. The first one is the motif. It can be directly imported from a MEME result table, from the motifs stored in the REACT database (accessed via the motif table), but also manually imported from an external motif definition, expressed as a PSSM.

The second important parameter is the sequence database to be searched. REACT contains pre-compiled data files containing all upstream regions of the currently implemented two model organisms but also of all of the respective reference organism. These regions are defined as the 200 bases upstream of the start codon of each gene. Other parameters to be defined are the maximum number of sequences to be displayed, a probability threshold and if genes overlapping with the upstream regions should be displayed in the results.

After the analysis has been performed, a graphical overview of the results in the form of a block diagram is displayed. It shows the matching regions for each motif within each sequence, the direction of the match (forward or reverse), the gene ID to which the upstream region belongs, and a probability value indicating the match strength. The information can also be displayed as in tabulated form. As usual, the diagram is interactive and provides a direct link to the corresponding gene-specific information.

If additional promising matches could be identified, they can then be integrated into a new iteration of creating motifs with the MEME-tool and re-checking them with MAST. Again, the integrative nature of REACT will enable and simplify such follow-up analyses.

5. Operating the REACT suite

We will conclude this chapter with a brief summary of how the REACT suite can be navigated and modified. For this purpose, we will first describe a typical work flow through the features of REACT from the perspective of a user (5.1). In the second section, we will specifically address the rights and options of REACT-administrators (5.2). Finally, we will provide a brief summary of the REACT concept and infrastructure (5.3).

5.1. Navigating REACT: The user approach

The functionality of the REACT suite relies on curated and comprehensive data that is provided by the organism-specific REACT database. It provides three different types of data: (i) gene-centric data (derived from genome sequences and their annotation), (ii) array-centric data (extracted from microarray databases and individual sources of transcriptome experiments), and (iii) motif data (based on experimental and computational evidence).

While there are many ways to use the REACT suite, it was developed with the goal in mind to enable the user to identify and characterize regulons starting from in-depth analyses of microarray datasets. Here, we will illustrate a typical workflow through the REACT suite (Fig. 7), in order to highlight the concept of REACT by connecting the central features that have so far been primarily described in isolation in the previous sections.

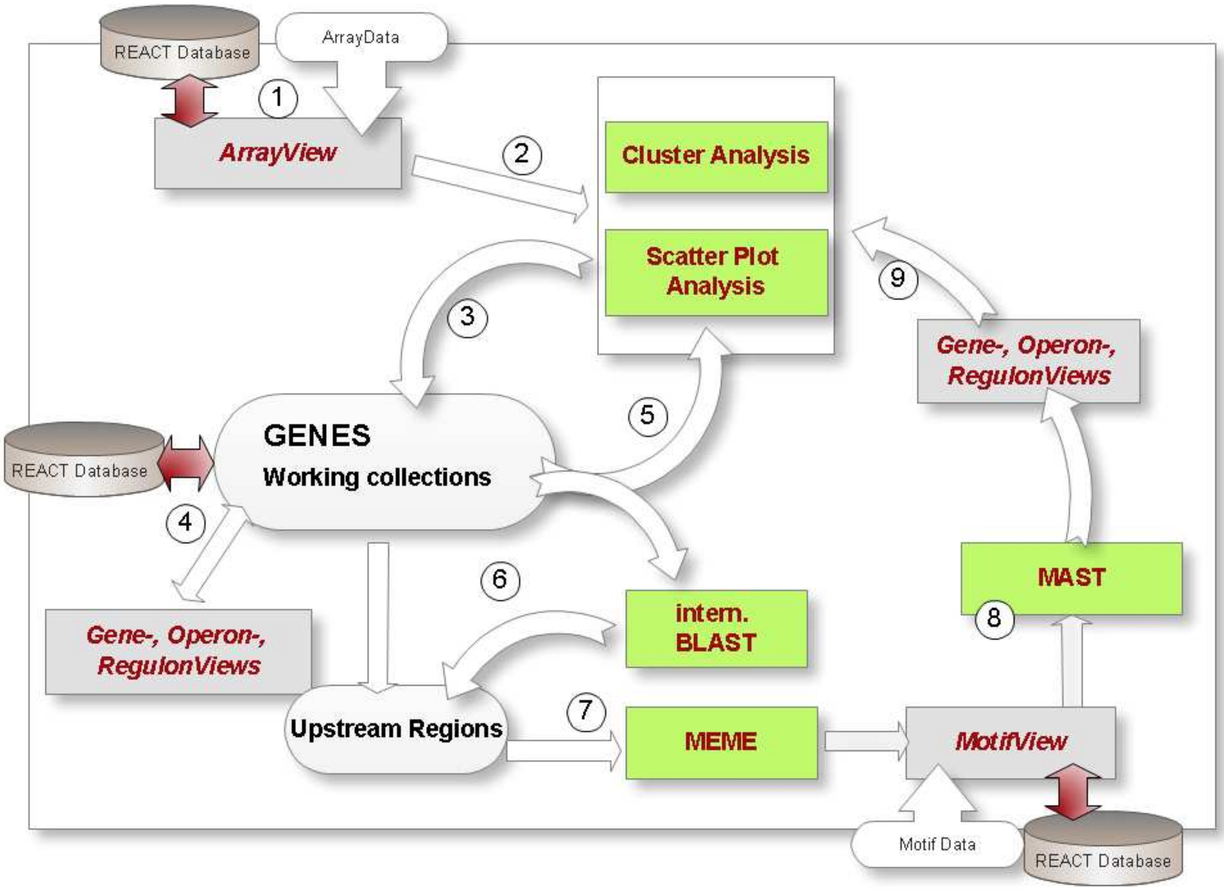


Figure 7. Work flow through the REACT suite. Major views are indicated by light grey, analysis tools by the green colour. The numbers refer to the description in the text.

A typical experiment could start with importing new microarray datasets ① to be subsequently analysed in detail by scatter plot or cluster analysis ②. These initial studies will presumably be performed genome-wide, but with a limited number of relevant microarray datasets. As a result, groups of interesting genes will be identified ③ that respond in a condition-specific manner and could potentially be co-expressed and therefore co-regulated. All unknown genes can be subjected to in-depth analyses, primarily using the information stored in the *GeneView* (including the external links), but to some extend also data from the *OperonView*, and *RegulonView* ④. Moreover, the group of selected genes could also be subjected to a second round of Cluster analysis, now incorporating a more diverse set of array conditions on this limited number of genes, to refine the clustering ⑤. Genes of interest derived from any of these studies can be selected and thereby added to the list of marked genes. For genes of interest that cannot be associated with known regulons, the upstream regions will be retrieved for the subsequent steps of the regulon annotation. To increase the chance of identifying regulator binding sites, internal BLAST analyses can be performed to extract upstream regions from orthologous genes from closely related species ⑥, assuming that they are subject to the same regulation. The collection of upstream regions will then be subjected to a MEME analysis to identify common sequence motifs as candidates for putative regulator binding sites ⑦. The motif definitions will be incorporated into the *MotifView* and subsequently used to screen genome sequences for additional candidates, using the MAST tool ⑧. If new candidate target genes preceded by the conserved motif could be identified, they will then be selected and subjected to the comprehensive studies described above, including Cluster analysis to compare their behaviour to the group of genes initially selected ⑨.

Enabling such iterative and interactive processes that rely on both sequence-based and array-based data and analysis tools is a major advantage of the REACT suite. Because of its concept and architecture, the necessary information and data flow can be controlled easily and the analyses can be performed efficiently.

5.2. Modifying REACT: The administrator mode

In the age of omics, new genome sequences and microarray studies are published with ever-increasing speed. It is therefore important that a REACT database, once established, can be updated regularly to grow with the increase of available data and information. But as a precautious measure to avoid data corruption and thereby ensuring the integrity of the database, it is advisable that not all users have the right to modify the core data at all times during analyses. REACT has therefore implemented two different user roles: the REACT-user normally works in the "read-only" mode. This will allow him to browse the data, perform analyses, and export data to external files. In contrast, login as a REACT-administrator enables the user to permanently import additional data (such as microarray datasets or new reference genomes), to edit data already implemented in the REACT database, and even to change the main views of REACT by incorporating additional links and features.

When logged in as REACT-administrator, most data displayed in the different views can be edited manually. To prevent unintentional data corruption, data can only be changed after deliberately switching into the edit-mode via the appropriate buttons, provided in each view. In the edit-mode, all editable data is displayed in green and all links are disabled. Any changes applied to the data remains transient until they are confirmed by the REACT-administrator and thereby sent to the REACT-server and stored permanently.

However, some data fields are not editable, as REACT uses them as immutable internal references (e.g. as primary and secondary database keys) to identify the complete dataset. This includes the names and IDs of genes, operons, or microarray datasets, as well as DNA and amino acid sequences from the *GeneView*, which are derived from and defined by the respective genomic sequence.

A REACT-administrator can also define new data fields for the above listed views according to the individual requirements, including plain text fields and numeric fields. Moreover, new external links can also be added to the views. While it is quite easy to generate plain text or numeric fields (as just the field name and type have to be defined within the REACT-administrator dialogue), creation of additional link-fields is technically a bit more demanding.

In addition to the aforementioned options, REACT-administrators can import additional array data, create new array sets or change the assignments of arrays to a set. They can also store motifs computed during a MEME analysis permanently within the REACT database or define new regulons. In the *RegulonView*, new operons can be connected to or removed from the regulons.

5.3. Expanding REACT: Embedding new organism-specific databases

REACT was initially developed for the analysis of two model organisms, *E. coli* and *B. subtilis*. But given the wealth of knowledge already available for these organisms, the potential of REACT may be even higher when applied to genomic and expression data of other, less well-characterized organisms.

Therefore, REACT is equipped with a small set of additional tools that enables researchers with little knowledge of programming languages or database administration to create new organism-specific REACT databases from scratch. Following the instructions provided by the software, the user has to download freely available files from sources like the NCBI, Uniprot or MicrobesOnline databases that contain the data used by REACT. Additional information (e.g. links to PFAM or PDB) will be obtained from the KEGG web service via SOAP/WSDL, again without the need for more than very basic user interaction.

After the creation of an initial, empty REACT database (done by importing a provided sql-file into the SQL database), the information contained in the downloaded files and provided by KEGG are parsed by helper tools provided by the REACT package, again minimizing user interaction.

Users with basic programming knowledge will then be able to extend the new REACT database by parsing data from additional data sources, depending on the organism chosen and the focus of the respective database. Subsequently, additional data needed by REACT (e.g. interbl BLAST databases) will be computed automatically. The user will now be able to connect with this newly created REACT database, in order to upload the first array sets.

5.4. Developing REACT: Concept, sources and infrastructure

TAs mentioned previously, the major aim of the REACT bioinformatics toolkit was the creation of an intuitive and interactive graphical user interface that allows an integrative view on genomic and microarray data and provides combined access to various bioinformatics tools commonly used in comparative genomic and transcriptomic studies. The overall structure of the REACT suite is illustrated in Fig. 8.

In the current release, the tools listed in Table 1 are integrated into the REACT suite. The software was implemented using a client/server architecture, enabling the parallel and locally distributed work of one to multiple users (Fig. 8). The REACT-server is the central computer running the database-managing software (MySQL), as well as all internal and integrated third-party analysis tools. The users will solely work with the corresponding client program, which can be installed on the personal computers or laptops of all users. Client and server are communicating via intra- or internet using remote method invocation (RMI) techniques. REACT is implemented as a java swing application, therefore client and server should run under a variety of operating systems depending only on the Java Runtime Environment (Version 5 or higher). However, in case of the server, this is limited by the external tools, as some of them (e.g. the MEME suite) depend on a Linux / Unix environment. To circumvent this limitation, REACT was developed and tested for being executable on Windows OS using Cygwin (1.5.x or higher), which is a Linux emulator for Windows and provides substantial Linux API functionality.

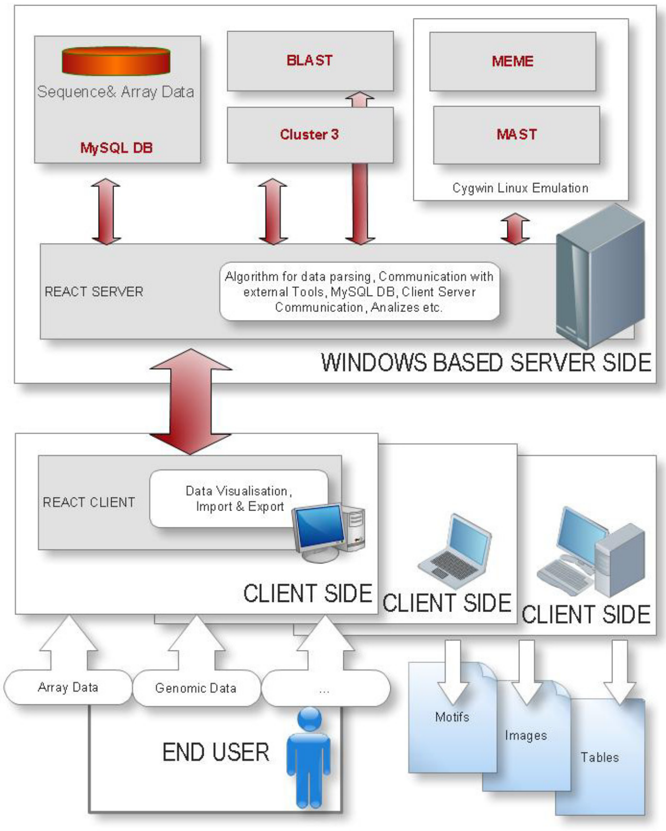


Figure 8. Structure, components and data flow of the REACT suite. See text for details.

Name	Version	Link	Reference
Blast	2.2.x	ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/LATEST/	(Altschul *et al.*, 1990)
Cluster 3	3.0	http://bonsai.hgc.jp/~mdehoon/software/cluster/	(de Hoon *et al.*, 2004)
MEME suite	4.0.0	http://meme.sdsc.edu/meme/meme-download.html	(Bailey *et al.*, 2009)
Cygwin	1.7.5	http://www.cygwin.com/install.html	n.a.
MySQL	5.5.x	http://www.mysql.de/downloads/mysql/	n.a.

Table 1. Third party software tools implemented in the REACT suite. “n.a.”, not applicable.

6. Conclusion

This chapter aimed at providing a thorough overview of the concept and functions of the REACT suite, a bioinformatics toolkit that was developed to simplify regulon predictions and comparative transcriptomic analyses for biologists with little to no background in bioinformatics. REACT was written in the believe that it will provide a powerful, yet simple-to-use platform that will hopefully also support the work of other research groups in extracting meaningful data from transcriptome studies with the help of comparative genomics. The complete REACT suite, including the databases for *B. subtilis* and *E. coli*, are available from the corresponding author upon request.

Author details

Peter Ricke and Thorsten Mascher*
Ludwig-Maximilians-University Munich, Germany

Acknowledgement

The authors would like to thank Tina Wecke for beta-testing of the REACT suite, providing the figures and critical reading of the manuscript. Work in the Mascher lab is financially supported by grants from the Deutsche Forschungsgemeinschaft (DFG). Development of the REACT suite was enabled by funding from the ‘Concept for the future’ of the Karlsruhe Institute of Technology (KIT) within the framework of the German Excellence Initiative.

7. References

Altschul, S. F., W. Gish, W. Miller, E. W. Myers & D. J. Lipman, (1990) Basic local alignment search tool. *J Mol Biol* 215: 403-410.

* Corresponding Author

Bailey, T. L., M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li & W. S. Noble, (2009) MEME SUITE: tools for motif discovery and searching. *Nucleic acids research* 37: W202-208.

Bairoch, A., (2000) The ENZYME database in 2000. *Nucleic acids research* 28: 304-305.

Barrett, T., D. B. Troup, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, R. N. Muertter, M. Holko, O. Ayanbule, A. Yefanov & A. Soboleva, (2011) NCBI GEO: archive for functional genomics data sets - 10 years on. *Nucleic Acids Res* 39: D1005-1010.

Brazma, A., (2009) Minimum Information About a Microarray Experiment (MIAME)--successes, failures, challenges. *TheScientificWorldJournal* 9: 420-423.

Brazma, A., P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, T. Gaasterland, P. Glenisson, F. C. Holstege, I. F. Kim, V. Markowitz, J. C. Matese, H. Parkinson, A. Robinson, U. Sarkans, S. Schulze-Kremer, J. Stewart, R. Taylor, J. Vilo & M. Vingron, (2001) Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nature Genetics* 29: 365-371.

Cao, M., P. A. Kobel, M. M. Morshedi, M. F. Wu, C. Paddon & J. D. Helmann, (2002) Defining the *Bacillus subtilis* s^W regulon: a comparative analysis of promoter consensus search, run-off transcription/macroarray analysis (ROMA), and transcriptional profiling approaches. *J Mol Biol* 316: 443-457.

de Hoon, M. J. L., S. Imoto, J. Nolan & S. Miyano, (2004) Open source clustering software. *Bioinformatics* 20: 1453-1454.

Dehal, P. S., M. P. Joachimiak, M. N. Price, J. T. Bates, J. K. Baumohl, D. Chivian, G. D. Friedland, K. H. Huang, K. Keller, P. S. Novichkov, I. L. Dubchak, E. J. Alm & A. P. Arkin, (2010) MicrobesOnline: an integrated portal for comparative and functional genomics. *Nucleic acids research* 38: D396-400.

Demeter, J., C. Beauheim, J. Gollub, T. Hernandez-Boussard, H. Jin, D. Maier, J. C. Matese, M. Nitzberg, F. Wymore, Z. K. Zachariah, P. O. Brown, G. Sherlock & C. A. Ball, (2007) The Stanford Microarray Database: implementation of new analysis tools and open source release of software. *Nucleic Acids Res* 35: D766-770.

Finn, R. D., J. Mistry, J. Tate, P. Coggill, A. Heger, J. E. Pollington, O. L. Gavin, P. Gunasekaran, G. Ceric, K. Forslund, L. Holm, E. L. Sonnhammer, S. R. Eddy & A. Bateman, (2010) The Pfam protein families database. *Nucleic Acids Res* 38: D211-222.

Gama-Castro, S., H. Salgado, M. Peralta-Gil, A. Santos-Zavaleta, L. Muniz-Rascado, H. Solano-Lira, V. Jimenez-Jacinto, V. Weiss, J. S. Garcia-Sotelo, A. Lopez-Fuentes, L. Porron-Sotelo, S. Alquicira-Hernandez, A. Medina-Rivera, I. Martinez-Flores, K. Alquicira-Hernandez, R. Martinez-Adame, C. Bonavides-Martinez, J. Miranda-Rios, A. M. Huerta, A. Mendoza-Vargas, L. Collado-Torres, B. Taboada, L. Vega-Alvarado, M. Olvera, L. Olvera, R. Grande, E. Morett & J. Collado-Vides, (2011) RegulonDB version 7.0: transcriptional regulation of *Escherichia coli* K-12 integrated within genetic sensory response units (Gensor Units). *Nucleic Acids Res* 39: D98-105.

Lechat, P., L. Hummel, S. Rousseau & I. Moszer, (2008) GenoList: an integrated environment for comparative analysis of microbial genomes. *Nucleic Acids Res* 36: D469-474.

Letunic, I., T. Doerks & P. Bork, (2009) SMART 6: recent updates and new developments. *Nucleic Acids Res* 37: D229-232.

Liolios, K., I. M. Chen, K. Mavromatis, N. Tavernarakis, P. Hugenholtz, V. M. Markowitz & N. C. Kyrpides, (2010) The Genomes On Line Database (GOLD) in 2009: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Res* 38: D346-354.

Rose, P. W., B. Beran, C. Bi, W. F. Bluhm, D. Dimitropoulos, D. S. Goodsell, A. Prlic, M. Quesada, G. B. Quinn, J. D. Westbrook, J. Young, B. Yukich, C. Zardecki, H. M. Berman & P. E. Bourne, (2011) The RCSB Protein Data Bank: redesigned web site and web services. *Nucleic Acids Res* 39: D392-401.

Sierro, N., Y. Makita, M. de Hoon & K. Nakai, (2008) DBTBS: a database of transcriptional regulation in *Bacillus subtilis* containing upstream intergenic conservation information. *Nucleic Acids Res* 36: D93-96.

Sigrist, C. J., L. Cerutti, E. de Castro, P. S. Langendijk-Genevaux, V. Bulliard, A. Bairoch & N. Hulo, (2010) PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Res* 38: D161-166.

Tatusov, R. L., E. V. Koonin & D. J. Lipman, (1997) A genomic perspective on protein families. *Science* 278: 631-637.

Utts, J. M., (2005) *Seeing through statistics*. Thomson Brooks.

Wecke, T., B. Veith, A. Ehrenreich & T. Mascher, (2006) Cell envelope stress response in *Bacillus licheniformis*: Integrating comparative genomics, transcriptional profiling, and regulon mining to decipher a complex regulatory network. *J. Bacteriol.* 188: 7500-7511.