9 Experimental Methodology

This chapter describes all the data sets used in the results chapter and the parameter settings for the various methods. In the final section, brief overviews of the Gene Ontology (GO) database and overrepresentation analysis (ORA) are provided. For general distribution analyses, the CRAN R package AdaptGauss [Thrun/Ultsch, 2015; Ultsch et al., 2015] was used. For the topographic map and island visualization the CRAN R package GeneralizedUmatrix was used [Thrun/Ultsch, 2017b]. For the ABC analysis the CRAN R package ABCanalysis was used [Thrun et al. 2015]. For DBS clustering and Pswarm projection the CRAN R package Databionic swarm was used [Thrun, 2017].

9.1 Data Sets

For the comparison of Pswarm as a projection method with the swarm-organized projection (SOP) algorithm, the original data sets of [Herrmann, 2011] were used. The artificial data sets of the Fundamental Clustering Problems Suite (FCPS) [Ultsch, 2005a] are summarized in Tab. 1 with regard to the cluster structures discussed in chapter 2.

> *"The FCPS comprises a collection of intentionally simple data sets with known classifications offering a variety of problems on which the performance of clustering algorithms can be tested. The data sets in the FCPS are specially designed such that the performance of clustering algorithms on particular challenges, for example, outliers or density- vs. distance-defined clusters, can be tested" [Ultsch/Lötsch, 2016, p. 4].*

All FCPS data sets have uniquely unambiguously defined class labels. For the error rate is defined as 1-Accuracy (Eq. 3.1 on p. 29) was is used as a sum over all true positive labeled data points by the clustering algorithm. The best of all permutation of labels of the clustering algorithm regarding the accuracy was chosen in every trial, because the labels are arbitrarily defined by the algorithms.
Additional data sets that are used in later chapters are also described below in alphabetical order. If these data sets are not discussed directly in chapter 10 and 11 than please see to Supplement C and D where the clusterings and the visualizations of DBS are shown. The hydrology data set and the pain genes data set are separately introduced in chapter 12.

9.1.1 Atom

> *"The Atom data set [Ultsch, 2005c] consists of two clusters in $\mathbb{R}^3$. The first cluster is completely enclosed by the second one and, therefore, cannot be separated by linear decision boundaries. Additionally, both clusters have different densities and variances. The Atom data set consists of a dense core of 400 points surrounded by a well separated, but sparse hull of 400 points. Both clusters are not linearly separable and many algorithms cannot construct a cohesive projection. The core is located in the center of the hull, which, for some methods based on averaging, makes it hard to solve it. The density of the core is much higher than the density in the hull. For data in the hull, some of the inner-cluster distances are bigger than the distance to the other clusters. The data set was not preprocessed" [Herrmann, 2011, pp. 99-100].*

9.1.2 Chainlink

The Chainlink data set [Ultsch, 1995; Ultsch et al., 1994] consists of two clusters in $\mathbb{R}^3$. Together, the two clusters form intricate links of a chain, and therefore, they cannot be separated by linear decision boundaries [Herrmann, 2011, pp. 99-100]. The rings are cohesive in $\mathbb{R}^3$; however, many projections are not. This data set serves as an excellent demonstration of several

M. C. Thrun, *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, https://doi.org/10.1007/978-3-658-20540-9_9

challenges facing projection methods: The data lie on two well-separated manifolds such that the global proximities contradict the local ones in the sense that the center of each ring is closer to some elements of the other cluster than to elements of its own cluster [Herrmann, 2011, pp. 99-100]. The two rings are intertwined in $\mathbb{R}^3$ and have the same average distances and densities. The data set was not preprocessed [Herrmann, 2011, pp. 99-100]. Every cluster contains 500 points.

9.1.3 EngyTime

The EngyTime data set [Baggenstoss, 2002] contains 4,096 points belonging to two clusters in $\mathbb{R}^2$; the data set is typical for sonar applications with the variables "Engy" and "Time" as a two-dimensional mixture of Gaussians. The clusters overlap, and cluster borders can be defined only by using density information. There is no empty space between the clusters. The data set was not preprocessed [Herrmann, 2011, pp. 99-100].

9.1.4 Golf Ball

The Golf Ball data set "consists of an artificial data set with 4,002 points, resembling a 3-D view of a golf ball" [Ultsch/Lötsch, 2016, p. 3]. "The points are located on the surface of a sphere at equal distances from each of the six nearest neighbors" [Ultsch/Lötsch, 2016, p. 4]. This data set does not contain any natural clusters. The data set was not preprocessed.

9.1.5 Hepta

The Hepta data set [Ultsch, 2003a] is used to illustrate the general problems with quality measures (QMs) and projections from the perspective of structure preservation. The three-dimensional Hepta data set consists of seven clusters that are clearly separated by distance, one of which has a much higher density. The data set consists of 212 points, comprising seven clusters of thirty points each plus two additional points in the center cluster. The centroids of the clusters span the coordinate axes of $\mathbb{R}^3$. The density of the central cluster is almost twice as high as the density of the other six clusters. The structure of the data set is clearly defined by distances and is compact. The data set was not preprocessed.

9.1.6 Iris

> *"Anderson's [Anderson, 1935] Iris data set was made famous by Fisher [Fisher, 1936], who used it to exemplify his linear discriminant analysis. It has since served to demonstrate the performance of many clustering algorithms" [G. Ritter, 2014, p. 220].*

The Iris data set consists of data points in $\mathbb{R}^4$ with a prior classification and describes the geographic variation of *Iris* flowers. The data set consists of 50 samples from each of three species of *Iris* flowers, namely, Iris setosa, Iris virginica and Iris versicolor. Four features were measured for each sample: the lengths and widths of the sepals and petals (see [Herrmann, 2011, pp. 99-100]). The observations have "only two digits of precision preventing general position of the data" [G. Ritter, 2014, p. 220] and "observations 102 and 142 are even equal" [G. Ritter, 2014, p. 220]. The *I.* setosa cluster is well separated, whereas the *I.* virginica and *I.* versicolor clusters slightly overlap (see [Herrmann, 2011, pp. 99-100]). This presents "a challenge for any sensitive classifier" [G. Ritter, 2014, p. 220]. The data set was not preprocessed (see [Herrmann, 2011, pp. 99-100]).

9.1.7 Leukemia

The anonymized leukemia data set consists of 12,692 gene expressions[66] from 554 subjects and is available from a previous publication [Haferlach et al., 2010]. Each gene expression is a logarithmic luminance intensity (presence call), which was measured using Affymetrix technology. The presence calls are related to the number of specific RNAs in a cell, which signals how active a specific gene is. Of the subjects, 109 were **healthy**, 15 were diagnosed with acute promyelocytic leukemia (**APL**), 266 had chronic lymphocytic leukemia (**CLL**), and 164 had acute myeloid leukemia (**AML**). "The study design adhered to the tenets of the Declaration of Helsinki and was approved by the ethics committees of the participating institutions before its initiation" [Haferlach et al., 2010, p. 2530]. The leukemia data set was preprocessed, resulting in a high-dimensional data set with 7.747 variables and 554 data points separated into natural clusters, as determined by the illness status and defined by discontinuities (see chapter 2). Additionally, patient consent was obtained for the data set, in accordance with the Declaration of Helsinki, and the Marburg local ethics board approved the study (No. 138/16) [Brendel, 2016].

9.1.8 Lsun3D

The Lsun3D data set consists of three well-separated clusters and four outliers in $\mathbb{R}^3$; it is based on the two-dimensional Lsun data set of Moutarde and Ultsch [Moutarde/Ultsch, 2005]. Two of the clusters contain 100 points each, and the third contains 200 points. "The inter-cluster minimum distances, however, are in the same range as or even smaller than the inner-cluster mean distances" [Moutarde/Ultsch, 2005, p. 28]. The data set consists of 404 data points and was not preprocessed.

9.1.9 S-shape

"The plain s-curve data set is an artificial set sampled from an S-shaped two-dimensional surface embedded in three-dimensional space" [Venna et al., 2010, p. 462]. The authors claim that "an almost perfect two-dimensional representation should be possible for a non-linear dimensionality reduction method, so this data set works as a sanity check" [Venna et al., 2010, p. 462]. Here, it is more important that the data set does not possess any natural clusters. The data set consist of 2000 data points in $\mathbb{R}^3$ and was not preprocessed.

9.1.10 Swiss Banknotes

> *"The idea is to produce bills at a cost substantially lower than the imprinted number. This calls for a compromise and forgeries are not perfect" [G. Ritter, 2014, pp. 223-224]. "If a bank note is suspect but refined, then it is sent to a money-printing company, where it is carefully examined with regard to printing process, type of paper, water mark, colors, composition of inks, and more. Flury and Riedwyl [Flury/Riedwyl, 1988] had the idea to replace the features obtained from the sophisticated equipment needed for the analysis with simple linear dimensions" [G. Ritter, 2014, p. 224].*

The Swiss Banknotes data set consists of six variables measured on 100 genuine and 100 counterfeit old Swiss 1000-franc bank notes. The variables are the length of the bank note, the height of the bank note (measured on the left side), the height of the bank note (measured on the right side), the distance from the inner frame to the lower border, the distance from the inner frame to the upper border and the length on the diagonal. The robust normalization of Milligan and

[66] Process with which information from a gene is used in the synthesis of functional RNA or protein.

Cooper [Milligan/Cooper, 1988] is applied to prevent a few features from dominating the obtained distances [Herrmann, 2011, pp. 99-100].

9.1.11 Target

The Target data set [Ultsch, 2005c] consists of two main clusters and four groups of four outliers each. The first main cluster is a sphere of 363 points, and the second cluster is a ring around the sphere and consists of 395 points. The data set as a whole consists of 770 points in $\mathbb{R}^2$. The main challenge of this data set is the four groups of outliers in the four corners. The data set was not preprocessed.

9.1.12 Tetra

The Tetra data set, which is part of the FCPS, consists of 400 data points in four clusters in $\mathbb{R}^3$ that have large intracluster distances [Ultsch, 2005c]. The clusters are nearly touching each other, resulting in low intercluster distances.

9.1.13 Tetragonula

The Tetragonula data set was published in [Franck et al., 2004] and is available to the public in the R package prabclus:

> *"It contains the genetic data of 236 Tetragonula (Apidae) bees from Australia and Southeast Asia. The data give pairs of alleles (codominant markers) for 13 microsatellite loci. The 13 string variables consist of six digits each" [Hennig, 2014]. The format is derived from the data format used by the GENEPOP 4.0 software implemented by Rousset in 2010. "Alleles have a three digit code, so a value of "258260" on variable V10 means that on locus 10, the two alleles have codes 258 and 260. "000" refers to missing values" [Hennig, 2014].*

9.1.14 Cuboid

The uniform Cuboid data set "was constructed by filling a cuboid with uniformly distributed random numbers in the x, y and z directions" [Ultsch/Lötsch, 2016, p. 5]. It was introduced in this publication. "A group structure [is] clearly absent by construction" [Ultsch/Lötsch, 2016, p. 5]; thus, the data set does not possess any natural clusters. The data set consists of 1000 data points in $\mathbb{R}^3$ and was not preprocessed. Additionally, another data set was generated by filling the same cuboid with Gaussian-distributed random numbers in the x, y and z directions.

9.1.15 Two Diamonds

"The data consists of two clusters of two-dimensional points. Inside each "diamond" the values for each data point were drawn independently from uniform distributions" [Ultsch, 2003c, p. 8]. The clusters contain 300 points each. "[In] [e]ach cluster[, the] points are uniformly distributed within a square, and at one point the two squares almost touch. This data set is critical for clustering algorithms using only distances" [Moutarde/Ultsch, 2005, p. 28]. The data set was not preprocessed.

9.1.16 Wine

The Wine data set [Aeberhard et al., 1992] is a 13-dimensional, real-valued data set. It consists of chemical measurements of wines grown in the same region in Italy but derived from three different cultivars. The robust normalization of Milligan and Cooper [Milligan/Cooper, 1988] is applied to prevent a few features from dominating the obtained distances [Herrmann, 2011, pp. 99-100].

9.1.17 Wing Nut

> *"The Wing Nut dataset […] consists [of] two symmetric data subsets of 500 points each. Each of these subsets is an overlay of equal[ly] spaced points with a lattice distance of 0.2 and random points with a growing density in one corner. The data sets are mirrored and shifted such that the gap between the subsets is larger than 0.3. Although there is a bigger distance in between the subsets than within the data of a subset, clustering algorithms like K-means parameterized with the right number of clusters (k=2) produce classification errors" [Moutarde/Ultsch, 2005, pp. 27-28].*

The data set was not preprocessed.

9.1.18 World Gross Domestic Product (World GDP)

The World GDP data set of [Leister, 2016] was constructed by selecting the purchasing power parity (PPP)-converted gross domestic product (GDP) per capita for the years from 1970 to 2010 from the data published in [Heston et al., 2012] of 190 countries. The data were logarithmized, and countries with missing values were not considered. In the resulting data set, 160 countries remain.

Table 9.1: Structures of the clusters in the artificial benchmark sets of the FCPS [Ultsch, 2005a] as defined in Chapter 2.

Data Set	Cluster Structure
Atom	Connected, direction-based, varying density, non-linear separable
Chainlink	Connected, direction-based, non-linear separable
EngyTime	Connected, unidirectional, varying density
Hepta	Compact, spherical, high intercluster distance
Lsun3D	Compact, ellipsoidal, outliers
Target	Connected, direction-based, outliers
Tetra	Compact, spherical, low intercluster distance
Two Diamonds	Compact, spherical, borders defined by discontinuity
Wing Nut	Connected, direction-based, linear separable
Golf Ball	No natural clustering tendency

9.2 Parameter Settings

The parameter settings for the clustering algorithms, the projection methods and the QMs used in this thesis are as follows.

9.2.1 Quality Measures (QMs)

Freely available implementations of the trustworthiness and discontinuity (T&D) measures and the precision and recall (P&R) measures (see chapter 6.1) in C++ code were used [Nybo/Venna, 2015]. For all other measures, self-developed implementations were used. Every QM is available in our R package, projections, which also includes R wrappers for the C++ code for the T&D and P&R measures. Our density-based version of the Shepard diagram is also available in the R package projections. This package can be downloaded from CRAN.

9.2.2 Projection Methods

For the projection methods considered here (see chapter 4), we used freely available code which is summarized in the ProjectionBasedClustering CRAN package [Thrun et al., 2017]: for principal component analysis (PCA) [Pearson, 1901], we used the PCA software available in the R package stats [R Development Core Team, 2008]; due to technical limitations ICA was omitted in the analysis; for curvilinear component analysis (CCA) [Demartines/Hérault, 1995], the CCA source code [Alhoniemi, et al., 2005] was ported from MATLAB to R and for t-distributed stochastic neighbor embedding (t-SNE) [Van der Maaten/Hinton, 2008], we used Donaldson's t-SNE implementation. Also included in the evaluation of various projection methods were the Neighbor Retrieval Visualizer (NeRV) algorithm ([Venna et al., 2010]) as implemented in the freely available C++ code [Nybo/ Venna, 2015] called in R (Thrun et al., 2017b]), the Sammon mapping technique for multidimensional scaling (MDS) [Sammon, 1969] available from [R Development Core Team, 2008], and the emergent self-organizing map (ESOM) algorithm as implemented in the R package Umatrix [Thrun et al., 2016a] which reproduced the results of [Ultsch/Mörchen, 2005].

For every projection method, only the default parameters were used, as given here (see also [Thrun et al., 2017]): The ESOM algorithm was set with 20 epochs; a planar lattice; 50 lines; 80 columns; a Euclidean neighborhood function; and a linear annealing scheme with a starting radius of 25, an end radius of 1, a starting learning rate of 0.5 and an end learning rate of 0.1.

For the NeRV method, lambda was set to 0.5 (for DCE baseline with PCA initialization) and 0.1 (default); the optimization scheme was set with 20 neighbors, 10 iterations, 2 conjugate gradient steps per iteration, and 20 conjugate gradient steps in the final iteration; and the points were randomly initialized (default). PCA and Sammon mapping did not require any input parameters. For CCA, 20 epochs, an initial step size of 0.5, and a radius of influence of 3*max($std(data)$) were specified. The t-SNE method was set with a perplexity of 30,100 epochs and a maximum number of iterations of 1.000. Aside from ESOM, every projection method is available through standardized wrappers in our R package projections on CRAN. The NeRV source code was modified only as required for compatibility with the CRAN package Rcpp. The Delaunay classification error (DCE) measure is also available in our R package projections on CRAN.

9.2.2.1 Swarm-Organized Projection (SOP)

The SOP parameterization was chosen following Herrmann [Herrmann, 2011, p. 98], using a 64 x 64 toroidal lattice with Gaussian neighborhoods, as described above. Further parameter specifications included a maximum of 500 iterations per epoch (for a single radius) and a jumping DataBot threshold of 5%. In a given iteration, the DataBots were allowed to jump only if the number of DataBots that wished to jump was above this threshold. If only 5% or fewer of the DataBots could find a better position or if the maximum number of iterations was exceeded, the radius was reduced. The starting radius was set to the maximum possible distance in the output space as defined by [Herrmann, 2011, p. 65]. The source code was implemented in R by Kohlhof [Kohlhof, 2010] under the supervision of Lutz Hermann and the SOP algorithm was executed using version 3.2.3 of R on a 64-bit Windows 7 system. Only Euclidean distances were used for SOP, consistent with the settings defined by [Herrmann, 2011, p. 98] and the restrictions of the source code. For this reason, the GDP194 data set was excluded because this

data set requires the use of special dissimilarities [Herrmann, 2011, p. 100]. Moreover, it should be mentioned that Rmin was set to a value much larger than 1 for this data set, although the precise number was not recorded [Herrmann, 2011, p. 167].
Other functional code for SOP or its extension for very large data sets, swarm-organized quantization, was not available to the author[67]. A self-developed implementation based on the algorithm exactly as described in chapter 7 yielded worse results on the data sets compared with that of Kohlhof [Kohlhof, 2010] because of the problems discussed in chapter 8.

9.2.2.2 Pswarm

For Pswarm, there are no parameters to set. In the case of the Wine data set, the distances were changed to squared Euclidean distances because the resulting distance distribution yielded a better distinction between the intra- and intercluster distances (see supplement B). The data sets were compared using the generalized U-matrix technique for three-dimensional visualization, as described in chapter 5. The CRAN R package Databionic swarm was used [Thrun, 2017].
Notably, the three-dimensional topographic map with hypsometric tints that is referred to as the generalized U-matrix in this thesis is completely different from the gray-scale two-dimensional visualization of Hermann [Herrmann, 2011, p. 72], which was also called the generalized U-matrix.All source code was executed in R 3.3.1 [R project, , 2008] on a 64-bit Windows 7 system.

9.2.3 Common clustering algorithms

For the k-means algorithm, the CRAN R package cclust was used [Dimitriadou/Hornik 2017]. For the single linkage (SL) and Ward algorithms, the CRAN R package stats was used [R Development Core Team, 2008]. For the Ward algorithm, the option “ward.D2” was used, which is an implementation of the algorithm as described in [Ward Jr, 1963]. For the spectral clustering algorithm, the CRAN R package kernlab was used [Karatzoglou et al., 2016] with the default parameter settings: “The default character string “automatic” uses a heuristic to determine a suitable value for the width parameter of the RBF kernel”, which is a “radial basis kernel function of the “Gaussian” type”. The “Nyström method of calculating eigenvectors” was not used (=FALSE). The “proportion of data to use when estimating sigma” was set to the default value of 0.75, and the maximum number of iterations was restricted to 200 because of the algorithm’s long computation time (on the order of days) for 100 trials using the FCPS data sets. For the mixture of Gaussians (MoG) algorithm, the CRAN R package mclust was used [Fraley et al., 2017]. In this instance, the default settings for the function “Mclust()” were used, which are not specified in the documentation. For the partitioning around medoids (PAM) algorithm, the CRAN R package cluster was used [Maechler et al., 2017].

9.3 Gene Ontology (GO)

An ontology is a representation of knowledge in which the relationships *part of* and *is a* are visualized in a directed acyclic graph (DAG). For the analysis of pain genes, the GO database was accessed via R 3.3.1 [R Development Core Team, 2008]. In the GO database, knowledge

[67] Lutz Herrmann’s 2011 Java implementation is largely identical to that of [Kohlhof, 2010], but the source code could not be compiled.

about molecular functions, biological processes and the cellular components of genes is defined using a controlled vocabulary consisting of labels called GO terms, which are used to represent biological concepts [Ashburner et al., 2000]. These terms describe and unify the attributes of genes and gene products[68] in a species-independent manner. "The GO terms are ordered in a directed acyclic graph (DAG), in which the set of genes annotated[69] to a certain term (node) is a subset of those annotated to its parent nodes" [Goeman/Mansmann, 2008]. Here, the important relationships between the nodes are of the "part of" type, resulting in a "top-down poly-hierarchy of GO terms" starting "at the root with terms with the broadest definition" and specializing "toward the leaves representing GO terms of the narrowest definition (details)" [Ultsch et al., 2016b]. Given a set of genes, ORA reveals the significance of a GO term that represents these genes or a subset of these genes [Backes et al., 2007].

9.3.1 Overrepresentation Analysis (ORA)

"In ORA, the most commonly used statistical test is based on the hypergeometric distribution or its binomial approximation ([…] among others). Let A denote a GO term or the set of genes annotated to A (with cardinality I_A), and let S denote the set of genes (with cardinality I_S) based on a certain criterion (i.e. differential expression) from a full gene list G (with cardinality I) in an experiment. The number of genes belonging to both S and A ($S \cap A$), denoted by n_A, indicates the representation of A in S. Under the null hypothesis that S and A are independent (i.e. the GO term is irrelevant to the gene cluster), n_A follows a hypergeometric distribution. The [p-value p] measuring the significance of association is the tail probability of observing n_A, or more genes annotated by A in S,

$$p = \sum_{k=n_A}^{min(I_A, I_S)} \frac{\binom{I_A}{k}\binom{I - I_A}{I_S - k}}{\binom{I}{I_S}} \qquad (9.1)$$

where $\binom{m}{n} = \frac{m!}{n!(m-n)!}$ is the binomial coefficient. Many software packages and webtools (Onto-Express, CLASSIFI, GoMiner, EASEonline, GeneMerge, FuncAssociate, GOTree Machine, etc.) have been developed based on the hypergeometric [p-value]. A detailed review can be found in Khatri and Drăghici [Khatri/Drăghici, 2005].

The hypergeometric [p-value] provides a straightforward measure of overrepresentation for each individual GO term. However, the major drawback of this approach is that it ignores the hierarchical structure in the GO DAG, which contains a substantial amount of information regarding the interactions among the GO terms" [Zhang et al., 2010, pp. 905-906].

For the ORA algorithm, the R package ORA was used [Lippmann et al., 2016].

9.3.2 Filtering via ABC Analysis

The resulting p-values p were filtered via ABC analysis (see chapter 5.3.2 on p. 49 for further explanation) [Ultsch/Lötsch, 2015]; thereafter, only the most important group A was considered for interpretation. For the ABC analysis algorithm, the CRAN R package ABC analysis was used [Thrun et al., 2015].

Here, it is argued that changing the threshold with respect to the significance of the p-value does not lead to better results. Aside from the problems discussed by Button and Nuzzo [Button et al., 2013; Nuzzo, 2014], the paramount goal of a gene analysis is to find GO terms with a

[68] Usually either Ribonucleic acid (RNA) or a protein

[69] For further details, see [Camon et al., 2003] and [Camon et al., 2004].

high effect strength. For this purpose, it is sufficient for the effect to be significant with regard to a commonly used (arbitrary) p-value threshold.

Let *E* be the strength of an effect as defined with respect to its p-value significance *p* (expressed as a percent), as follows:

$$E = -10\log(p) \qquad (9.2)$$

At first glance, the definition given in Eq. 9.2 is contradictory to the equation above (1).

On the one hand, the calculation of p-values based on the Fisher test with $p(I_A, I_S, k, I)$ requires four parameters; on the other hand, one would calculate the strength of an effect based on the relative difference between the expected value *e* and the observed value *o*, known as the fold change FC:

$$FC(k, e) = 2\frac{o - e}{o + e} \qquad (9.3)$$

Here, the p-values are calculated analogously to Backes [Backes et al., 2007], where the formula is called the hypergeometric test. However, the hypergeometric test is simply the Fisher test based on the hypergeometric distribution [Ultsch, 2014a]. The hypergeometric distribution is defined as

$$f(I_A, I_S, k, I) = \frac{\binom{I_A}{k}\binom{I - I_A}{I_S - k}}{\binom{I}{I_S}} \qquad (9.4)$$

Given this distribution, the expected value $e(f)$ is defined as

$$e(f) = \sum_{k=0}^{I_S} k\frac{\binom{I_A}{k}\binom{I - I_A}{I_S - k}}{\binom{I}{I_S}} = I_S\frac{I_A}{I} \qquad (9.5)$$

It can be shown that Eq. 9.2 is directly proportional to the definition of the expected number of genes in Eq. 9.5 [Ultsch, 2014a]. Therefore, the observed number of genes *o* are compared against a hypergeometric distribution (Eq. 9.4) around the value for the expected genes number of *e* in Eq. 9.5, and in the special case of ORA, the p-values imply more than merely significance.

One may ask why the calculation must be complicated if the fold change, as defined in Eq. 9.3, could be used. The disadvantage of the fold change is illustrated in the following equation:

$$FC(o, e) = 2\frac{o - e}{o + e} = 2\frac{c * o - c * e}{c * o + c * e} \qquad (9.6)$$

According to this equation, one expected gene compared with four observed genes yields the same value as 100 expected genes compared with 400 observed genes. Clearly, the effect strength here is not the same.

It could be argued that this problem could be solved by reducing the p-value threshold to a low level, such that only a few GO terms are represented in the DAG. However, one would be obliged to do this manually for every ORA calculation. Moreover, to the author's knowledge, every tool or package that uses GO terms or performs ORA calculations has a different version of the GO database. Hence, the p-value calculation has a measurement error that is difficult to specify. Furthermore, even if a tool used the database obtained directly from the Gene Ontology Consortium, there is an even stronger source of measurement error: every list of genes I_S to be

analyzed was obtained based on microarray experiments with arbitrary thresholds or probe intensities (for a detailed discussion, see [Khatri et al., 2012, p. 3]).
Here, with regard to the definition of the effect strength given in (Eq. 9.2), it is assumed that the magnitudes of the p-values do not change regardless of measurement errors. This is the reason for taking the logarithm of the p-value in (Eq. 9.2). Moreover, Figure 9.1 shows the correlation between the fold change FC (Eq. 9.3) and the effect strength E (Eq. 9.2) for a given interval of the number of annotated genes per GO term. Consistent with Ultsch [Ultsch, 2014a], it is argued here that in ORA, the p-values are directly proportional to the effect sizes.
After setting the p-value threshold to *0.05*, which is a generally accepted level of significance, and calculating the corresponding GO terms, the results of an ABC analysis of the effect strengths as given by (2) can be obtained. The relevant GO terms are defined as those assigned to group A in the ABC analysis.

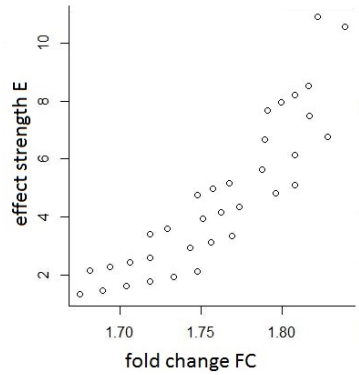

Figure 9.1: Scatter plot of the fold changes *FC* of Eq. 9.6 and the corresponding *E* value of Eq. 9.3 for numbers of annotated genes per GO term in the range [10,25] is proportional.