G

G Domain

► Groove (G) Domain

G Type Domain

► Groove (G) Domain

G1 Checkpoint

► Cell Cycle Transition, Detailed Regulation of Restriction Point

G1 Phase Checkpoint

► Cell Cycle Transition, Principles of Restriction Point

G2/M Transition

► Cell Cycle Transitions, G2/M

GCN Control

► Translational Control of GCN4

Gcn2-dependent General Translational Control

► Translational Control of GCN4

Gene and Allele Names

► Gene and Allele Nomenclature

Gene and Allele Nomenclature

Marie-Paule Lefranc
Laboratoire d'ImmunoGénétique Moléculaire, Institut de Génétique Humaine, UPR 1142, Université Montpellier 2, Montpellier, France

Synonyms

Gene and allele names; Gene and allele symbols

Definition

Gene and allele nomenclature for immunoglobulins (IG) or antibodies and T cell receptors (TR) has been set up by IMGT®, the international ImMunoGeneTics information system® (http://www.imgt.org) (► IMGT® Information System) (Lefranc and Lefranc 2001a, b). The gene and allele nomenclature is based on

W. Dubitzky et al. (eds.), *Encyclopedia of Systems Biology*, DOI 10.1007/978-1-4419-9863-7,

the concepts of classification (generated from the ▶ CLASSIFICATION Axiom) of ▶ IMGT-ONTOLOGY, the global reference in ▶ immunogenetics and ▶ immunoinformatics.

The four major concepts of classification, *Group*, *Subgroup*, *Gene*, and *Allele*, have allowed the gene and allele nomenclature for the V, D, J, and C gene type (▶ Gene Type) of the IG and TR whatever the receptor type, the chain type (▶ Chain Type), and the species from fish to human (▶ TaxonRank).

The *Group* concept allows to classify a set of genes which belong to the same multigene family, within the same species or between different species. For the IG and TR, "Group" allows to classify a set of genes which belong to the same ▶ GeneType (V, D, J or C).

The *Subgroup* concept allows to classify a subset of genes which belong to the same group, and which, in a given species, share at least 75% of identity at the nucleotide sequence level (and in the germline configuration (▶ Configuration Type) for the IG and TR V, D, and J genes).

The *Gene* concept allows to classify, in the ▶ IMGT® Information System, a unit of DNA sequence that can be potentially transcribed and/or translated (this definition includes the regulatory elements in 5′ and 3′, and the introns, if present). The leafconcepts (▶ IMGT-ONTOLOGY, Leafconcept) of "Gene" are gene names (Lefranc and Lefranc 2001a, b). In IMGT-ONTOLOGY, a gene name is composed of the name of the species (leafconcept of the ▶ TaxonRank "Species") and of the international Human Genome Organisation (HUGO) Nomenclature Committee (HGNC)/IMGT gene symbol, for example, *Homo sapiens* IGHV1-2. By extension, orphon (▶ Location Type) and ▶ pseudogene (▶ FunctionalityType) gene names are also leafconcepts of "Gene."

The *Allele* concept allows to classify a polymorphic variant of a gene. The leafconcepts of "Allele" are allele names. Alleles identified by the mutations of the nucleotide sequence are classified by reference to allele *01. For immunoglobulin (IG) and T cell receptor (TR) genes, full description of mutations and allele name designations are recorded for the core sequences (V-REGION, D-REGION, J-REGION, C-REGION). They are reported in Alignment tables, in IMGT Repertoire http://www.imgt.org and in IMGT/GENE-DB (Giudicelli et al. 2005) of IMGT®, the international ImMunoGeneTics information system® (▶ IMGT® Information System).

Cross-References

- ▶ Chain Type
- ▶ Configuration Type
- ▶ FunctionalityType
- ▶ Gene Type
- ▶ IMGT-ONTOLOGY
- ▶ IMGT-ONTOLOGY, CLASSIFICATION Axiom
- ▶ IMGT-ONTOLOGY, Leafconcept
- ▶ IMGT® Information System
- ▶ Immunogenetics
- ▶ Immunoinformatics
- ▶ Location Type
- ▶ Pseudogene
- ▶ Structure Type
- ▶ TaxonRank

References

Giudicelli V, Chaume D, Lefranc M-P (2005) IMGT/GENE-DB: a comprehensive database for human and mouse immunoglobulin and T cell receptor genes. Nucleic Acids Res 33:D256–D261

Lefranc M-P, Lefranc G (2001a) The immunoglobulin FactsBook. Academic Press, London, pp 1–458

Lefranc M-P, Lefranc G (2001b) The T cell receptor FactsBook. Academic, London, pp 1–398

Gene and Allele Symbols

- ▶ Gene and Allele Nomenclature

Gene Association Analysis, Frequent-Pattern Mining

Jesús Aguilar-Ruiz[1], Domingo Rodríguez -Baena[1] and Ronnie Alves[2]
[1]School of Engineering, Pablo de Olavide University, Escuela Politécnica Superior, Seville, Spain
[2]Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil

Definition

In ▶ transcriptomics, Gene association analysis helps to infer ▶ gene expression profiles from ▶ DNA

microarray studies by enumerating and evaluating all possible gene association patterns.

Gene association patterns are essentially ▶ association rules induced by sample frequency of a gene in an observed ▶ DNA microarray study. For example, an association rule between genes in the form *R1: geneA → geneB, and geneC* could be an indicative that when geneA is 'over-expressed', it is likely to observe an 'over-expression' in geneB and geneC.

The starting point in the Gene Association Analysis is a N × M matrix of gene expression values, where the rows correspond to experimental conditions and columns represent genes. Most of the techniques for extraction of gene association patterns require that the matrix passes through a discretization process. So, the input matrix is preprocessed in order to transform their values in binary values.

Depending on the application and on the type of information to be extracted, values "0" and "1" will have a specific meaning. As an example, consider that the microarray will be mapped to a binary gene expression matrix in such way that a gene tagged with "1" in a particular condition stands for an "Over-expression" and "0" stands for an "Under-expression." Then, all significant genes are selected and must pass through an enumeration process. Again, genes considered as "significants" depend on the specific experiment. In the former matrix example, significant genes could be the "Over-expressed" ones. Finally, frequency is observed to detect remarkable frequent patterns.

▶ Frequent pattern mining is the most costly task in gene association analysis (Alves et al. 2010). Given the high dimensionality of the gene expression matrices, classical techniques, like the ones based on *Apriori* (*see the review paper of* Han et al. 2007 *for further information*), might not be suitable to high-dimensional data analysis. New frequent pattern mining techniques properly devised for gene association analysis have been developed to face this new challenge.

Once all the frequent gene groups have been obtained, they are combined in order to generate association rules.

The significant gene association rules discovered (with remarkable frequency) are evaluated to check their biological relevance (Alves et al. 2010). To do so, biological databases could be used for checking, statistically, overrepresentations of gene association patterns with gene annotations. Text mining can be also applied for searching genes being already published in the medical literature. In fact, it is also possible to include biological background in early stages of the Gene Association Analysis, being especially important in integrative genomics. In this case, association patterns will present a global appreciation rather than a local perspective of a microarray study. For instance, a rule like *Ribosome → [−]T6, [−]T7* combines information about metabolic pathways, expression, and temporal data, meaning that genes involved in Ribosome pathway are under-expressed in that respective time points.

References

Alves R, Rodriguez-Baena D, Aguilar-Ruiz J (2010) Gene association analysis: a survey of frequent pattern mining from gene expression data. Brief Bioinform 11(2): 210–224

Han J, Cheng H, Xin D, Yan X (2007) Frequent pattern mining: current status and future directions. Data Min Knowl Discov 15:55–86

Gene Association and Linkage Analysis

Roger Higdon
Seattle Children's Research Institute, Seattle, WA, USA

Synonyms

Gene mapping; Genetic association; Genetic epidemiology; Genome-wide association; Linkage disequilibrium

Definition

Gene association is the association between a genetic variation (genotype, haplotype, or single nucleotide polymorphism (SNP)) and a physical trait (phenotype), typically the presence or absence of a disease. Linkage analysis is the study of gene association due to their proximity on the same chromosome.

Characteristics

Genetic linkage is the tendency of gene loci or alleles to be inherited together due to their physical proximity on the same chromosome. This proximity causes them to stay together during meiosis, and they are therefore genetically linked. Genetic association tests are used to find genetic linkage between a genetic trait or polymorphism and a physical trait or phenotype such as a disease (de Bakker et al. 2005). A similar term to genetic linkage with a different meaning is linkage disequilibrium, a term used in the study of population genetics for the nonrandom association of alleles at two or more loci, not necessarily on the same chromosome (Terwilliger and Ott 1994). Genetic association studies are truly testing for linkage disequilibrium which may or may not be due to actual linkage on a chromosome.

A number of different types of studies are used to test for genetic association or linkage. These tests may be used for testing the association of a disease versus a single genotype, but more commonly they are used to test against a large number of SNPs in order to determine the location of genes associated with a particular disease. The studies are known as genome-wide association studies (Manolio et al. 2010). The most commonly used tests are case-control studies which compare the distribution of genetic polymorphism between a sample of disease case and control subjects. A simple chi-squared or ▶ Fisher's exact test can be used to test for association. If cases and controls are not well matched for ethnicity or geographic origin, then genetic associations may be confounded with the effect of population stratification. Population stratification occurs where subpopulations have different frequencies of genetic traits due to common ancestry.

Family-based tests can have much more statistical power than case-control studies because they utilize the genetic relationships between family members. However, these studies are much more difficult to conduct since they can be difficult for genetic data among family members. The earliest forms of family-based tests are based on pairs of siblings affected with the same disease; this is known as the affected sib-pair (ASP) test. If a genetic marker is not linked to the disease, then transmission of alleles should be governed by random Mendelian inheritance, but if the disease gene and marker gene are linked, then the siblings should share alleles more often than by chance. The simplest ASP tests are based on a comparison of the number of observed shared alleles to the number expected based upon Mendelian inheritance using a chi-squared test. Another family-based test is the transmission disequilibrium test (TDT) (Spielman and Ewens 1996). The TDT utilizes the genotypes or haplotypes of parents and a disease-affected child. In the TDT the number of alleles transmitted from parents to the child is compared to the number not transmitted. Patterns of nonrandom transmission indicate association of genetic marker with a disease. The simplest TDT is based on McNemar's test for matched pairs.

For genome-wide association studies, a disease is compared to a large number of known genetic markers, typically SNPs using case-control studies or family-based tests. Genome-wide association studies identify the SNPs most closely associated with a disease enabling researchers to identify the most likely location(s) of disease-related genes. The log-odds (LOD) score is the most common metric for ranking association; it is simply the value of the log-likelihood test for genetic association. LOD scores are based upon parametric tests typically used in case-control studies or family-based tests; however, there are also nonparametric-based alternatives. Tests for association can be carried out using single point linkage, which considers the association disease and genetic markers on an individual basis, or by using multipoint linkage, which utilize multivariate measures to calculate the association between disease and a set of genetic markers. Multipoint linkage tests can increase statistical power but are much more complicated and numerically intensive to carry out.

Gene association and linkage studies can also be based on quantitative outcome rather than just dichotomous outcomes such as disease state. These are known as quantitative trait loci (QTL) models and studies (Sen and Churchill 2001).

Cross-References

▶ Fisher's Test

References

de Bakker PI, Yelensky R, Pe'er I, Gabriel SB, Daly MJ, Altshuler D (2005) Efficiency and power in genetic association studies. Nat Genet 37(11):1217–1223

Manolio TA, Guttmacher AE, Manolio TA (2010) Genomewide association studies and assessment of the risk of disease. N Engl J Med 363(2):166–176

Sen S, Churchill GA (2001) A statistical framework for quantitative trait mapping. Genetics 159:371–387
Spielman RS, Ewens WJ (1996) The TDT and other family-based tests for linkage disequilibrium and association. Am J Hum Genet 59(5):983–989, PMC 1914831
Terwilliger JD, Ott J (1994) Handbook of human genetic linkage. Johns Hopkins Press, Baltimore

Gene Expression

Yufei Huang
Picower Institute for Learning and Memory, Massachusetts Institute of Technology, Cambridge, MA, USA
Greehey Children's Cancer Research Institute, University of Texas at Texas Health Science Center at San Antonio, San Antonio, TX, USA
Department of Epidemiology and Biostatistics, University of Texas at Texas Health Science Center at San Antonio, San Antonio, TX, USA

Definition

Gene expression is the process by which a functional product, such as protein, is synthesized according to genetic information. It usually consists of the following stages: transcription, post-transcription, translation, and post-translation.

Cross-References

▶ Gene Regulation

Gene Expression Biomarkers

Beatriz Stransky[1] and Sandro J. de Souza[2]
[1]Universidade Federal do ABC, Santo Andre, Brazil
[2]Ludwig Institute for Cancer Research, Sao Paolo, Brazil

Definition

Gene expression biomarkers are biological markers identified through large-scale gene expression profiling. They may consist of a single gene product or represent a combination of several gene products, a gene expression signature.

Characteristics

In the past, most of the studies exploring the transcriptome (collection of all RNA molecules in a cell or tissue) were done with methods targeting single or few molecules, like Northern blotting. The last 20 years has witnessed a great development in technologies that allow a large-scale screening of genes expressed in a given cell or tissue. In this respect, the technologies that have contributed most for this exploratory work are Expressed Sequence Tags (ESTs), ▶ DNA microarray, and Serial Analysis of Gene Expression. Although important, these technologies have all significant limitations specially related to sensitivity. It is believed that all these technologies cover just a fraction of the whole transcriptome of a given sample, missing for example most of the low abundant RNA messages.

In spite of these limitations, these technologies have made important contributions to the process of ▶ biomarkers discovery, especially for gene expression biomarkers. A gene expression biomarker can be either a product from a single gene or a signature comprised of many gene products. Examples of the first type of biomarker include the transmenbrane protein HER2, a breast cancer biomarker that presents a direct correlation to prognostic and if a given tumor will be sensitive to antibody therapy with trastuzumab. The most known example of the second type of gene expression biomarker is MammaPrint™, a 70-gene breast cancer signature that assess the recurrence potential of certain types of breast tumors.

Although a gene expression biomarker can be used for the classification of any cell or tissue state, most of the efforts for the identification of such biomarkers have been devoted to pathological states, especially cancer. Gene expression biomarkers have clinical relevance (▶ Biomarkers, Clinical Relevance) since they can be used to determine the presence of the disease, its stage, and to analyze and monitor the responses to treatments. They can also be classified as diagnostic biomarkers or predictive biomarkers (Blomme and Warder 2008).

Discovery Process

The standard pathway for gene expression biomarker discovery and development can be described by four

steps: (1) the establishment of sets of high-quality samples; (2) the use of a large-scale gene expression platform; (3) the identification of a gene expression biomarker through mathematical and computational strategies; and (4) validation of the discriminatory power of the gene expression biomarker in an independent set of samples.

Initially, the discovery process for gene expression biomarkers involves the purification of RNA for a given sample and the use of one of the above platforms for gene expression profiling. The process is only effective if a large number of samples are used to account for biological variability. ▶ Data mining strategies are then used to identify putative gene expression biomarkers. These biomarkers, either a single product or a signature, need to be further validated in a larger and independent panel of samples.

In the last few years, the field of gene expression biomarkers has been revolutionized by the development of next-generation sequencing. These new sequencing technologies have allowed an exhaustive analysis of the transcriptome, covering with high sensitivity all RNA molecules in a sample. Moreover, they have allowed the identification of a large collection of transcripts variants generated through processes like alternative splicing and alternative polyadenylation (Caballero et al. 2001).

The last decade has also witnessed the emergence of non-coding RNAs, especially ▶ microRNAs, as critical regulatory molecules in many normal and pathological conditions. Not surprisingly, micro-RNAs have been characterized as gene expression biomarkers in many biological conditions, especially cancer (Jeffrey 2008).

Gene Expression Biomarkers and Systems Biology

Although these large-scale gene expression profiling technologies have generated datasets that are per se very informative, their integrated use has proven to be the most effective way to extract more valuable information. In that aspect, platforms based on systems biology approaches have been very useful in serving as a scaffold for integration of gene expression data.

A critical issue when integrating gene expression data into a system biology approach is the ▶ ontology of genes and gene products. To be effective, this integration has to obey certain classification rules that will allow the identification of pathways and functional modules as gene expression biomarkers. The most used networks to serve as a scaffold for the integration of data are protein-protein interaction networks (interactome) and gene regulatory networks. The use of these networks allows the use of graph theory to explore other quantitative parameters of the data.

The dissection of a gene expression profiling into pathways and functional modules (through a system biology approach) will serve to improve the identification of new gene expression biomarkers as well as to increase the opportunities for therapeutic intervention in case of diseases. Even if a drug target is not directly differentially expressed in a disease state, the corresponding drug may still be useful if the pathway that the target belongs is seen differentially expressed in that disease.

Cross-References

- ▶ Biomarker Discovery, Knowledge Base
- ▶ Biomarkers
- ▶ Biomarkers, Clinical Relevance
- ▶ Biomarkers, Protein Expression
- ▶ Data Mining
- ▶ DNA Microarrays
- ▶ Gene Expression
- ▶ Gene Ontology

References

Blomme EAG, Warder SE (2008) Gene expression-based biomarkers of drug safety. In: Wang F (ed) Biomarker methods in drug discovery and development. Humana Press, Totowa

Caballero OL, de Souza SJ, Brentani RR, Simpson AJG (2001) Alternative spliced transcripts as cancer markers. Dis Markers 17:67–75

Jeffrey SS (2008) Cancer biomarker profilling with microRNAs. Nat Biotechnol 26:400–401

Gene Expression Biomarkers, Ranking

Ronnie Alves
Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil

Synonyms

Differential expression analysis; Gene ranking; Order statistics; Top-K lists

Definition

Gene expression biomarkers by ranking refer to the task of selection and ordering of the most significant genes from transcriptomic studies, where, usually, a control versus target experimental setting is properly designed for the evaluation of differentially expressed genes (DEG) associated to a particular tissue or cell in the related study. Selection is evaluated through a differentiation score (Statistics), and then, genes are ranked in ascending order having high-scored genes (Gene Markers) in the top of the gene list.

Characteristics

Ranking Gene Expression Differences

Differential expression analysis is the traditional strategy for ranking genes (Biomarkers, Gene Markers) from gene expression data. The task of finding DEG falls into the following steps (Steinhoff and Vingron 2006):

- Ranking: genes are ranked according to their evidence of differential expression.
- Assigning significance: a statistical significance is being assigned to each gene.
- Cut-off value: to arrive at a limited number of DEG a cut-off value for the statistical significance needs to be determined.

The simplest experimental setting is the comparison of two experimental groups C_A and C_B and asking for their differences in each gene. Basically, one can use the empirical intensity values of each series C_A and C_B and introduce an ordered list of ranked differences between them. Typically, in this "simple" approach a fixed cut-off is chosen, usually this is a fold change of two. Thus, all genes showing a fold change of more than two are considered to be differential. It has been shown in several studies that such approach is able to provide a decent DEG list with high reproducibility, but it has also been reported having low statistical power and high false discovery rate (FDR).

Availability of repetitions provides for a richer spectrum of applicable statistical procedures. Either one compares two groups or multiple groups. In the two-group comparison, one considers either a paired or unpaired situation. Comparing a healthy group with a diseased one is an example for an unpaired experiment because the samples are independent. An example for a paired situation is gene expression measurements of one cell line before and after chemical treatment. Furthermore, the availability of replicates enables to rank genes according to their associated t-statistic for each gene:

$$t = m/(st/\sqrt{n}) \quad (1)$$

where m is the difference of means across replicates; st, the within groups standard deviation; and n, the number of genes considered for testing. F-scores are the straightforward generalization of t-scores in the multiconditional case (Ewens and Grant 2004). Problems arise when genes with small intensity differences present almost no changes between groups. This might yield high t-scores and thus, these genes will pop up in the top of the list. To overcome this situation one could artificially enlarge these variances by employing different penalizing factors in the t-statistic test. In (Lonnstedt and Speed 2002) a parametric empirical Bayes approach being equivalent to a penalized t-statistic is introduced:

$$t = m/\sqrt{\left(\left(f + \mathrm{std}^2\right)/n\right)} \quad (2)$$

where f is the penalty value which is estimated from the mean and standard deviation of the variance across samples. The approach entitled "significance analysis of microarrays," or just SAM (Tusher et al. 2001), is one of the most used penalized t-statistic. Another similar strategy based on percentiles is proposed in (Efron et al. 2001), although in this case it suggested the application of an additive penalizing factor in the denominator of the t-statistic that it is the 90th percentile of the standard deviation across samples. If the penalization factor is zero the method is solely based on an ordinary t-statistic test.

A number of linear methods have also been proposed for ranking gene expression. In the ANOVA (Regression, Statistics) model it is assumed a linear model of specific effects (like dye, slide, treatment, gene effects and their associated interactions) for log intensities of all genes. A moderated t-statistic is suggested in (Smyth 2004) which is proportional to the t-statistic with sample variance offset. It is also possible to design a robust linear model for each single gene, estimating contrasts of all pairwise comparisons of tested groups. Rather than applying t-statistics approaches for ranking gene expression differences, one could take advantages of nonparametric tests,

usually based on a Wilcoxon rank sum test or permutation *t*-test. Rank-based strategies use rank scale information instead of the numerical ones for differentiation expression analysis and it can be applied without any assumptions regarding data distribution.

Significance of the Ranked Gene Lists

Once gene rankings are found, following statistical tests, the next step is checking its significance. Usually, researchers use a *P-value* cut-off of 0.05 and genes presenting a lower *P-value* are those showing significance. On the other hand, in order to obtain such p-value multiples tests are conducted, and, in the end, it can increase the false discovery rate. To overcome this situation one alternative is finding a criterion to limit the number of testing procedures. Thus, one can either remove from the test genes which are not expected to be relevant or ignoring those genes presenting quite low expression variation across all experimental conditions. Therefore, when using multiple tests one must evaluate the error rate associated when assessing the statistical significance. Given a type I error rate (false positive or false discovery rate) controlling for multiple testing means correcting *P-values* such that the given error rate can be guaranteed for all tests. Methods can be divided into those that control the family wise error rate (FWER) or the false discovery rate (FDR). The probability of at least one type I error within the significant genes is called FWER. The FDR is the expected proportion of type I errors within the rejected hypotheses. Table 1 describes the various outcomes when applying multiple tests to determine which of the m hypothesis tests are statistically significant. Specifically, V is the number of type I errors and R is the total number of significant null hypotheses (total discoveries). The FWER is defined to be

$$FWER = \Pr(V \geq 1) \tag{3}$$

and the FDR is usually defined to be (Benjamini and Hochberg 1995)

$$FDR = E\left[\frac{V}{R \vee 1}\right] = E\left[\frac{V}{R}, R > 0\right] \Pr(R > 0) \tag{4}$$

The effect of "R v 1" in the denominator (Eq. 4) of the first expectation is to set V/R = 0 when R = 0.

Gene Expression Biomarkers, Ranking, Table 1 Possible outcomes from m hypothesis tests based on a significance threshold t ∈ (0, 1] to their associated *P-values*

	Not significant (P-value > t)	Significant (p-value ≤ t)	Total
Null true	U	V	m_0
Alternative true	T	S	m_1
	W	R	m

As demonstrated in (Benjamini and Hochberg 1995), the FDR offers a less strict multiple test criterion than the FWER, being more appropriated for differential expression analysis.

Consensus Gene-Ranking

Several statistical tests have been proposed in the literature making the selection of one unique ranking method a hard task. Indeed, there is no consensus with respect to one universal (unique) rank test (r). One suggested alternative is to evaluate empirically the ranked list while combining several tests before electing the final gene list, rather than pushing optimizations into one particular test trying to find an expected gene list.

The common criterion is evaluating the level of consensus among the top-100 ranked genes by intersecting the top-k lists:

$$s(r, r', k) = \sum_{j=1}^{p} I(r_j \leq k \wedge r'_j \leq k) \tag{5}$$

where I denotes the indicator function [I(A) = 1 if A is true, I(A) = 0 otherwise], or the proportion s(r, r′, k)/k of genes in the top-k list from l that, are also in the top-k list from l′, also denoted as percentage of overlap or percentage of overlapping genes (POG). An overview regarding stability measures for consensus gene-ranking is provided in (Boulesteix and Slawski 2009).

Consensus can also be achieved by measuring the levels of convergence and divergence while evaluating several dissimilarity matrices (r_{dist}) based on finding gene rankings. The motivation is to use a "voting" strategy in such way that gene rankings that are identified as closer to each other could be an indication of agreement among different tests. Thus, like employing ensemble-learning (Baldi and Brunak 2001) strategies in supervised problems, one could use an ensemble clustering solution by combining several candidate-ranking solutions for

devising a unified gene ranking. Once all distance matrices are calculated for all gene rankings, a simple consensus function could be applied:

$$Consensus_{dist(i,j)} = \min(r_{dist(i,j)}, r'_{dist(i,j)}) \quad (6)$$

Gene rankings can be grouped according to a consensus function, and a hierarchical clustering approach could be used to define the meta-rankings (Gene Modules). Given that more than two gene rankings could be used in the unified model, *quantiles* could be used as they provide more robust statistics.

Gene Expression Biomarkers Methods and Applications

Most biomarkers have been discovered by molecular profiling studies, based on association or correlation. One of the first molecular profiling studies was reported in (Golub et al. 1999), who showed that gene expression patterns could classify tumors, thereby remarking new insights like the stage, grade, clinical course, and response to treatment of a tumor. Recent work in several groups has indentified unique gene expression patterns, being strong correlated with clinical outcomes. Candidate biomarkers solely based on gene expression data depend on both available samples and on the ranking algorithm, increasing the amount of data (meta-analysis) can improve the reproducibility of the resulting gene-ranking models.

A large variety of ranking algorithms are available as workflow applications such as GSEA, GeneTrailExpress, g:Profiler, and Taverna; or web-based bioinfomatics resources as omniBioMarker and Oncominer; or customized software packages as the widely used R Bioconductor.

Cross-References

- ▶ Biomarkers, Ranking
- ▶ Gene Expression Biomarkers
- ▶ Gene Expression Biomarkers, Ranking
- ▶ Gene Regulation
- ▶ Modular Organization of Gene Regulatory Networks
- ▶ Regression Analysis

References

Baldi P, Brunak S (2001) Bioinformatics: the machine learning approach (adaptive computation and machine learning), 2nd edn. MIT, Cambridge

Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc B Methodol 57(1):289–300

Boulesteix ALL, Slawski M (2009) Stability and aggregation of ranked gene lists. Brief Bioinform 10(5):556–568

Efron B, Tibshirani R, Storey JD (2001) Empirical Bayes analysis of a microarray experiment. J Am Stat Assoc 96(456):1151–1160

Ewens WJ, Grant GR (2004) Statistical methods in bioinformatics: an introduction (statistics for biology and health), 2nd edn. Springer, New York

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537

Lonnstedt I, Speed TP (2002) Replicated microarray data. Stat Sin 12:31–46

Smyth GK (2004) Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Stat Appl Genet Mol Biol 3(1):1–25

Steinhoff C, Vingron M (2006) Normalization and quantification of differential expression in gene expression microarrays. Brief Bioinform 7(2):166–177

Tusher V, Tibshirani R, Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. PNAS 98:5116–5124

Gene Expression Regulation

- ▶ Regulation and Autoregulation

Gene Mapping

- ▶ Gene Association and Linkage Analysis

Gene Modulation

- ▶ Gene Regulation

Gene Network

- ▶ Biological Disease Mechanism Networks
- ▶ Gene Regulatory Networks

Gene Normalization with GNAT

Conrad Plake
Biotechnology Center (BIOTEC), Technische Universität Dresden, Dresden, Germany

Definition

Gene normalization in literature (▶ Entity Mention Normalization) refers to the process of assigning a gene name occurring in text its corresponding entry in a gene database. The ▶ text mining system GNAT has been developed to accomplish this task (Hakenberg et al. 2008).

Characteristics

Gene normalization is required for data integration purposes, such as the extraction of gene annotation from scientific publications to complement data stored in gene databases. Due to ambiguity of many gene names, which often refer to orthologous or entirely different genes, named after phenotypes and other biomedical terms, or resemble common English words, developing automated methods to gene normalization is challenging (Fundel and Zimmer 2006). The process of gene normalization carried out by text mining systems such as GNAT typically involves multiple steps starting from building a gene name dictionary (data acquisition), finding gene mentions in text, and assigning potential gene identifiers (gene mention recognition), to finally deciding on the correct identifier for each gene mention as reference to a gene database (gene mention normalization).

Data Acquisition

GNAT utilizes data available from the gene database Entrez Gene and the protein database UniProt. These databases contain known gene symbols, synonyms, and alternative designations, as well as additional annotations, which further describe genes and their products. Prior to building a gene name dictionary, each name is transformed into a regular expression to allow for its recognition in text even if slight spelling variations occur (e.g., hyphens vs. white spaces and Roman vs. Arabic numbers). All regular expressions are then compiled into a deterministic finite state automaton, where each accepting state stores the database identifiers for all names that end at this state. Besides the dictionary, GNAT generates a profile for each gene comprising species information, known annotations provided by the ▶ gene ontology, associated diseases, sequence annotations, and other textual descriptions separated by type.

Gene Mention Recognition

GNAT parses a text at the character level using the dictionary automaton, starting from the beginning of every word. If an accepting state is reached with the last character of a word, the text passage consumed so far is stored as a gene mention together with all potential gene identifiers. Shorter mentions contained within longer ones are ignored. Next, each mention and its surrounding words are compared against predefined word lists and regular expressions to identify and discard mentions that resemble a gene name but in the current context refer to something else (e.g., cell line, disease, protein domain).

Gene Mention Normalization

After recognition of gene mentions, the task of normalization is to identify each mention by assigning the database identifier of the referenced gene. If a text also mentions one or more species, GNAT only allows for genes from these species, ignoring all others. Species recognition is carried out by ▶ AliBaba, but can, for example, also be performed using LINNAEUS (▶ Named Entity Recognition and Normalization of Species, LINNAEUS). If a gene mention is left with a single gene identifier assigned, this identifier is taken as reference to the Entrez Gene database. In cases where GNAT encounters an ambiguous gene mention, that is, a mention with more than one gene identifier assigned, it compares the text at hand against each candidate gene's profile. This comparison, which takes the different types of annotation available into account, yields a normalized score that reflects the similarity between text and profile. The gene whose profile is most similar to the text is taken as reference for the ambiguous mention.

Related Tools

Other text mining tools (▶ Text Mining, Tools) performing gene normalization in text are available via the biocreative metaserver (▶ BioCreative MetaServer and Text-Mining Interoperability Standard).

Cross-References

- ▶ AliBaba
- ▶ Applied Text Mining
- ▶ BioCreative Meta-Server and Text-Mining Interoperability Standard
- ▶ Entity Mention Normalization
- ▶ Gene Ontology
- ▶ Named Entity Recognition
- ▶ Named Entity Recognition and Normalization of Species, LINNAEUS
- ▶ Text Mining, Tools

References

Fundel K, Zimmer R (2006) Gene and protein nomenclature in public databases. BMC Bioinformatics 7:372. doi:10.1186/1471-2105-7-372. http://dx.doi.org/10.1186/1471-2105-7-372

Hakenberg J, Plake C, Leaman R, Schroeder M, Gonzalez G (2008) Inter-species normalization of gene mentions with GNAT. Bioinformatics 24(16):i126–i132. doi:10.1093/bioinformatics/btn299. http://dx.doi.org/10.1093/bioinformatics/btn299

Gene Ontology

Jiguang Wang
Beijing Institute of Genomics, Chinese Academy of Sciences, Beijing, China

Definition

Gene Ontology (GO) (Ashburner et al. 2000) provides a hierarchical vocabulary of GO terms for describing functions and characteristics of gene or gene products in different databases and different species. There are three structured, species-independent ontologies, i.e., biological processes, cellular components, and molecular functions. GO project is a collaborative effect to maintain, make cross-links for, and develop tools to use the three ontologies.

References

Ashburner M, et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet. 2000;25:25–29.

Gene Ranking

- ▶ Biomarkers, Ranking
- ▶ Gene Expression Biomarkers, Ranking

Gene Redundancy

- ▶ Genetic Redundancy

Gene Regulation

Jia Meng[1] and Yufei Huang[1,2,3]
[1]Picower Institute for Learning and Memory, Massachusetts Institute of Technology, Cambridge, MA, USA
[2]Greehey Children's Cancer Research Institute, University of Texas at Texas Health Science Center at San Antonio, San Antonio, TX, USA
[3]Department of Epidemiology and Biostatistics, University of Texas at Texas Health Science Center at San Antonio, San Antonio, TX, USA

Synonyms

Gene modulation; Regulation of gene expression

Definition

Gene regulation (Watson and Roberts 1965) concerns the control of the synthesis of functional gene products or expression (mainly mRNAs or proteins) in cells. Proper gene regulation defines the distinct phenotype of a biological system and ensures its stability, and misregulation is usually associated with disease. It is also essential in promoting the versatility and adaptability of a living organism under various environmental conditions.

Characteristics

Stages of Gene Expression Regulation

To control the synthesis of gene product (mRNA or proteins), gene regulation may assume different modes

Gene Regulation, Table 1 Different gene regulations and stages

Stage	Targets	Regulatory process	High-throughput technology
Chromatin domains	DNA	DNA methylation	Methylation array/seq, ChIP-seq, LC-MS
		DNA phosphorylation	
		Histone deacetylation	
Transcription	DNA-RNA	Transcription factor	ChIP-chip, ChIP-seq
		Repressors	
		Activators	
		Enhancers	
Post-transcription	RNA	Capping	Microarray, RNA-seq, Exon array, HITS-CLIP, RIP-seq
		Splicing	
		Polyadenylation	
		RNA editing	
		microRNA silencing	
Translation	RNA-protein	Translational initiation	
		Peptide elongation	
		Termination	
Post-translation	Protein	Acylation	Protein array, LC-MS
		Phosphorylation	
		Protein degradation	

at including chromatin domain, transcription, post-transcription, and translation. Summarized in Table 1 is a list of different modes of gene regulations.

High-Throughput Technologies for Gene Regulation

As gene regulation occurs at multiple stages of gene expression, different high-throughput technologies based on microarray, deep sequencing, or Liquid chromatography-mass spectrometry (LC-MS) have been developed for investigating different gene regulations, which monitor the expression profiles of thousands of genes simultaneously. A summary of these technologies is shown in Table 1. Note that, although the final product of gene regulation is protein, most of the technologies are array/sequencing based and they mainly measure mRNA activities. This reality is due to the low sensitivity of existing proteomics technologies including LC-MS and protein array in measuring protein expression.

Systems Biology and Gene Regulatory Network

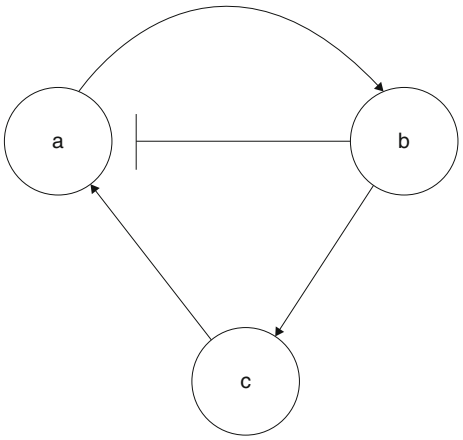
Systems biology seeks to model all the individual units of a biological system under a proper mathematical framework, such that both the overall structure and construction units of the system can be captured simultaneously. Since response of cells to changing endogenous or exogenous conditions is governed by intricate gene regulations, gene regulatory network is a systems biology approach toward understanding overall molecular mechanisms that control the cell response. Computational reconstruction of gene regulatory network is one of the current research focuses of computational system biology.

Modeling of Gene Regulatory Network

Modeling is the key step in uncovering gene regulatory network. Various models have been proposed and we discuss some most popular models in the following.

Boolean Network

A Boolean network (Shmulevich and Dougherty 2009) models the association instead of direct regulation between sets of genes, where transition of the association states are modeled by Boolean functions. In a Boolean network–based gene regulatory network, both the regulatory states and regulatory effects are digitalized, which facilitate the related computation and analysis. Particularly, a Boolean node represents the binary state (on/activation or off/repression) of a gene, and an edge indicates the regulatory effect/association between the two nodes, which is modeled by Boolean functions. When modeling the regulatory dynamics, states of all Boolean variables can update with time. A simple example of a Boolean network is



Gene Regulation, Fig. 1 Boolean network

Gene Regulation, Table 2 State transition table

Previous			Next		
A	B	C	A	B	C
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Gene Regulation, Table 3 Boolean network state update

Time	0	1	2	3	4	5	6	7	8	...
a	1	0	0	1	0	1	0	1	0	...
b	0	1	0	1	0	1	0	1	0	...
c	0	0	1	0	1	0	1	0	1	...

illustrated in Fig. 1, where three genes are regulating each other. Its state transition table is shown in Table 2, and given the initial states (1, 0, 0), their states will be updated as shown in Table 3.

Coupled Ordinary Differential Equations

Instead of describing the final states of regulated targets, Coupled Ordinary Differential Equations (ODEs) (Chicone 2006) or stochastic ODE seeks to describe the reaction kinetics, or the dynamics of regulatory effects. Suppose that an ODEs GRN consists of G nodes representing the expression of G genes, and let $y_g(t), g \in [1, 2, ..., G]$ represent the expression of the g-th gene at time t. Then, the temporal regulatory effects on a target gene by the regulators can be modeled by the ODE:

$$\frac{dy_g}{dt} = f_g(y_1, y_2, ..., y_G) \tag{1}$$

This equation describes quantitatively the temporal evolution of the regulatory network. The regulatory function f_g can be derived from known biochemical principles. The ODE model (1) provides a more accurate account of gene regulation than the Boolean network model, but it requires greater knowledge regarding gene regulation and is also computationally more complicated.

Factor Analysis Model

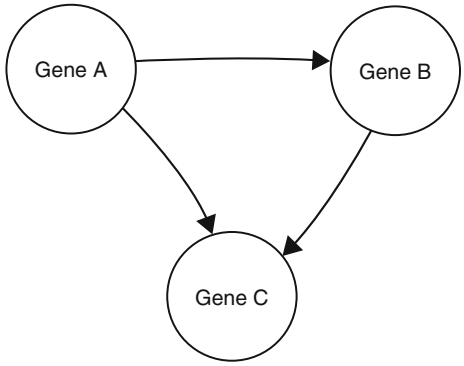
Factor analysis model (Child 2006) usually aims at modeling transcriptional regulatory network, that is, direct regulation of mRNA expression by transcription factors (TFs). Consider a set of genes regulated by a set of transcription factors. Let $y_g(t), g \in [1, 2, ..., G]$ represent the mRNA expression of the g-th gene at time t and $x_l(t), r \in [1, 2, ..., L]$ represent the protein-level expression of the l-th transcription factor also at time t. The TF regulation can be modeled by a linear relationship as in (2):

$$\begin{bmatrix} y_1(t) \\ \vdots \\ y_G(t) \end{bmatrix} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,l} \\ \vdots & \ddots & \vdots \\ a_{G,1} & \cdots & a_{G,L} \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_L(t) \end{bmatrix} \tag{2}$$

where $a_{g,l}$ is the regulatory coefficient of the g-th gene by the l-th transcription factor. Note that (2) can also be written in a matrix form:

$$\mathbf{Y} = \mathbf{AX} \tag{3}$$

where, $\mathbf{Y}$ is the mRNA expression matrix of genes, $\mathbf{X}$ is the protein expression level of transcription factors, which is usually difficult to measure, and $\mathbf{A}$ is the regulatory coefficient matrix or the loading matrix. In a factor model, both $\mathbf{A}$ and $\mathbf{X}$ are unknowns, and factor analysis seeks to estimate both $\mathbf{A}$ and $\mathbf{X}$ simultaneously from the observation $\mathbf{Y}$. Since the model is underdetermined,



Gene Regulation, Fig. 2 Bayesian network

Gene Regulation, Table 4 Probability of A

Gene A	
UP	DOWN
0.2	0.8

Gene Regulation, Table 5 Conditional probability of B given A

	Gene B	
Gene A	UP	DOWN
DOWN	0.4	0.6
UP	0.2	0.8

Gene Regulation, Table 6 Conditional probability of C given A and B

		Gene C	
Gene A	Gene B	UP	DOWN
DOWN	DOWN	0.01	0.99
DOWN	UP	0.8	0.2
UP	DOWN	0.1	1.9
UP	UP	0.95	0.05

additional knowledge needs to be incorporated to constrain the solution space. Examples of such constraints are orthogonal factors, the sparsity of A, known regulations, etc.

Bayesian Network

A Bayesian network (Heckerman 2008) or belief network is a probabilistic graphical model that models a set of genes and their conditional dependencies via a directed acyclic graph (DAG). Bayesian network includes a wide variety of models as special cases, ranging from the basic deterministic model to more sophisticated hierarchical probabilistic models (Huang et al. 2009). An example of the Bayesian network is shown in Fig. 2, and the conditional probability of each node is summarized in Tables 4–6.

Depending on the modeling and the availability of the data, the inference goal can be to estimate the unobserved variables (such as the regulatory coefficients between genes) and/or to identify the network structure (such as whether or not a gene is regulated by another gene).

Discussion

Various models have been proposed for modeling systems-level gene regulations. However, because of the complexity of gene regulation and the availability of data, no existing models can cover all aspects of gene regulations. Most existing models only apply to a limited subset of the mRNAs or proteins of interest, and they model mostly indirect regulations/association between genes products.

Limited by the current techniques, most current models utilize only mRNA expression levels for modeling gene regulation. The protein-level expression is often inappropriately approximated by the corresponding mRNA level; such approximation can lead to high modeling error. As the increase of computational processing power, integrating disparate data sets to account for multiple aspects of gene regulation is the current trend of systems biology, where proper modeling of different types of data is the key.

Cross-References

▶ Epigenetics
▶ Epigenetics, Drug Discovery

References

Chicone C (2006) Ordinary differential equations with applications. Springer, New York
Child D (2006) The essentials of factor analysis. Continuum, London
Heckerman D (2008) A tutorial on learning with Bayesian networks. Innov Bayesian Netw 156:33–82

Huang Y, Tienda-Luna I et al (2009) Reverse engineering gene regulatory networks. Signal Proc Mag IEEE 26(1):76–97
Shmulevich I, Dougherty E (2009) Probabilistic boolean networks: the modeling and control of gene regulatory networks. Siam, New York
Watson J, Roberts K (1965) Molecular biology of the gene. Wa Benjamin, New York

Gene Regulation Network

▶ MicroRNA-mRNA Regulation Networks

Gene Regulatory Networks

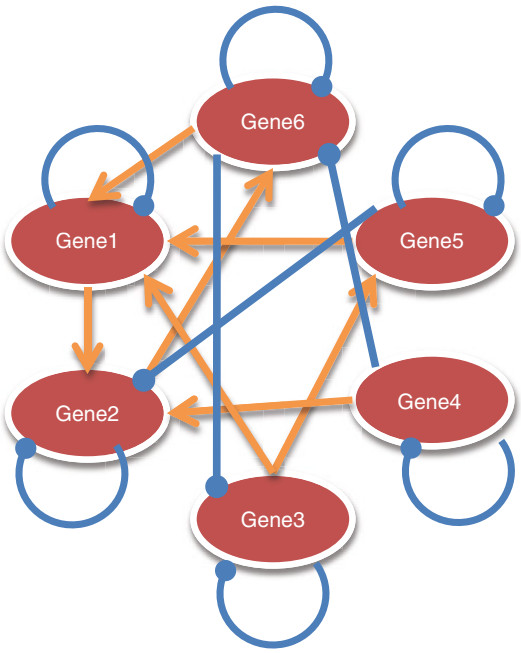
Yong Wang
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China

Synonyms

Gene network; Genetic network

Definition

Cells efficiently carry out molecular synthesis, energy transduction, and signal processing across a range of environmental conditions by networks of genes, which we define broadly as networks of interacting genes, proteins, and metabolites (Chen et al. 2009). Formally speaking, a gene regulatory network or genetic regulatory network (GRN) is a collection of DNA segments in a cell which interact with each other (indirectly through their RNA and protein expression products) and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA. In general, each mRNA molecule goes on to make a specific protein (or set of proteins). In some cases this protein will be structural, and will accumulate at the cell-wall or within the cell to give it particular structural properties. In other cases the protein will be an enzyme; a micro-machine that catalyses a certain reaction, such as the breakdown of a food source or toxin. Some proteins, though, serve only to activate other genes, and these are the transcription factors that are the main players in regulatory networks or cascades. By binding to the promoter region at the start of other genes they turn them on, initiating the production of another protein, and so on. Some transcription factors are inhibitory.



Gene Regulatory Networks, Fig. 1 A toy example for the gene regulatory network with six genes. Here, nodes denote genes, and edges denote their regulatory relationships. Specifically, *red arrows* represent activation, and *blue arcs* represent repression

Mathematically, gene regulatory network is defines as a directed graph (refer to Fig. 1 for a 6-gene network) in which nodes denote genes and edges denote their regulatory relationships. And usually we use a matrix J to represent the gene regulatory relationships. The regulatory relationships can be directed, signed, and weighted. For example, element J_{ij} represents an effect of gene j on gene i, while J_{ji} represents an effect of gene i on gene j. Thus the influence between gene i and gene j is directed. Furthermore, a sign associated with J_{ij} represents a specific role of regulation. For example, if the sign of J_{ij} is positive, gene j is the activator of gene i. On the other hand, if the sign of J_{ij} is negative, gene j is the repressor of gene i. Furthermore the associated weight (the absolute value) of element J_{ij} indicates how strong the regulatory interaction is. Obviously, a zero weight of J_{ij} indicates no interaction between two genes.

Characteristics

Significance of Gene Regulatory Networks

Many cellular processes such as cell cycle, cellular differentiation, and apoptosis are well controlled via gene regulation. On the one hand, gene regulatory network is essential for all viruses, prokaryotes, and eukaryotes. It includes the processes to turn the information in genes into gene products (proteins) to increase the versatility and adaptability of an organism by allowing the cell to express protein when needed. Even more complex, gene regulatory network drives the processes of cellular differentiation and morphogenesis, leading to the creation of different cell types in multicellular organisms where the different types of cells may possess different gene expression profiles though they all possess the same genome sequence. On the other hand, gene regulation system is extremely complex to allow intuitively understanding regarding to its nonlinear, dynamics, and robustness properties.

To understand the complex mechanisms inside the gene regulation system, biologists usually apply various treatments to perturb cells including heat shock, stress, and other techniques, and then observe the phenotype change of the cell such as the concentration changes of interested molecules in the cell. This type of research can produce small-scale regulatory relationships and it is hard to construct a mathematical model for whole-genome scale. Recently, the step to understand the gene regulation system inside a cell is greatly accelerated by the invention of new biological techniques. Specifically, DNA microarray and other high throughput technologies were developed which enabled an experimenter to simultaneously measure the concentration of thousands of molecules from a single sample of cells or tissues. Such data offer a possibility to systematically identify a model of a cell's underlying control systems.

Modeling Gene Regulatory Network

Biologists developed several experimental techniques to reveal the gene regulatory network. For example, gene perturbation experiments (e.g., knockouts or RNA interference) may indicate relationships between genes due to direct or indirect genetic interactions. In contrast, chromatin immunoprecipitation chip data may reveal direct protein–DNA interactions or cofactor associations with bound transcription factors.

Protein–DNA interaction data concerns the interactions between proteins and DNA, particularly between transcription factors and their target promoters. They fundamentally define the transcriptional regulatory network of the cell. The recently developed ChIP-chip methodology involves the chromatin immunoprecipitation of an epitope-tagged transcription factor (TF) bound to DNA fragments containing target promoters, followed by the hybridization of those amplified DNA fragments to an intergenic microarray. Currently large amounts of ChIP-chip data in yeast and other organisms are publicly available. For example, genome-wide location data performed in yeast by Harbison et al. (2004) and Lee et al. (2002) contain information regarding the binding of 204 regulators to their respective target genes in rich medium, and can be downloaded from their websites (http://web.wi.mit.edu/young/regulatory_network/). ChIP-chip data have the advantage that they provide a direct biochemical link between TFs and promoters and have the potential to identify targets without knowing the activating conditions. From this viewpoint, ChIP-chip data are a very important source of information for analyzing direct transcriptional regulatory interactions.

Sequencing techniques can also be used to reveal gene regulatory relationships by systematically analyzing gene upstream regions in the genome to identify potential regulatory elements (also known as regulatory binding motifs). These motifs, often represented as regular expressions, were transformed into the corresponding weight matrices. We can then simply count the occurrences of regular expression-type patterns with the goal of identifying possible gene regulatory relationships. The weight matrices corresponding to these motifs are subsequently used to screen all intergenic sequences. The higher the score of a motif hit in a gene, the more likely it will be a regulatory relationship (Brazma et al. 1998).

More reliable sources for gene regulatory relationships are from the literature and curated databases. For example, YEASTRACT (Yeast Search for Transcriptional Regulators And Consensus Tracking) is a curated repository of more than 12,500 regulatory associations between transcription factors and target genes in *Saccharomyces cerevisiae* (Teixeira et al. 2006), based on more than 900 bibliographic references. The information in YEASTRACT is updated regularly to match the recent literature on yeast regulatory networks. Since the regulatory relationships from literature and databases are usually generated by small-scale experiments, they are believed to be of high quality compared to large-scale experiments.

The most abundant data to model the gene regulatory network is the microarray data. Microarray technologies enable the simultaneous measurement of all RNA transcripts in a cell, producing tremendous amounts of gene expression data from different research groups. For instance, the Stanford Microarray Database (SMD) has deposited data for 70,113 experiments, from 341 labs and 56 organisms, as of 2007 (Demeter et al. 2007).

DNA microarray experiments are usually classified based on the type of array used in the experiment (cDNA and oligonucleotide arrays) or according to the organism that is profiled. From the viewpoint of gene regulatory network modeling, we distinguish between static and time series experiments. In static expression experiments, a snapshot of the expression of genes in different samples is measured. In time series expression experiments, a temporal process is measured at various time intervals. Another important difference between these two types of data is that while static data from a sample population (e.g., ovarian cancer patients) are assumed to be independently and identically distributed, time series data exhibit a strong autocorrelation between successive points.

Since many biological systems are dynamic systems, temporal profiles of gene expression levels during a given biological process can often provide more insights into how gene expression levels evolve in time and how genes are dependent among each other during a given biological process. One important feature of such time-course gene expression data is the possible dependency of gene expression levels across time points for a given gene. In addition, as gene expression levels evolve over time, time intervals can be an important factor that affects the gene expression levels. Methods which can preserve the time sequence and the time dependence of the observed data are needed for analyzing the time-course gene expression data (Wang et al. 2006).

Collectively, these microarray data enable the analysis on gene expression profiles to detect dependencies among genes over different conditions, i.e., reverse engineering the gene regulatory networks. So far, a wide variety of approaches have been proposed to infer gene regulatory networks from time-course data or perturbation experiments (De Hoon et al. 2003; Dewey and Galas 2001; Friedman 2004; Gardner et al. 2003; Holter et al. 2001; Husmeier 2003; Nachman et al. 2004; Tegner et al. 2003). These approaches include discrete models of Boolean networks and Bayesian networks, and continuous models of neural networks and difference/differential equations. A common challenge for all these models is the scarcity of the data, since a typical gene expression dataset consists of relatively few time points (often less than 20) with respect to a large number of genes (generally over thousands). In other words, the number of genes far exceeds the number of time points for which data are available, making the problem of determining gene regulatory network structure a difficult and ill-posed one (D'Haeseleer et al. 2000).

Two General Reverse-Engineering Strategies

The first strategy is the "physical" strategy for reverse-engineering transcription regulation using mRNA expression data (Gardner and Faith 2005). The physical approach seeks to identify the protein factors that regulate transcription, and the DNA motifs to which the factors bind. In other words, it seeks to identify true physical interactions between regulatory proteins and their promoters. An advantage of this strategy is that it can reduce the dimensionality of the reverse-engineering problem by restricting possible regulators to TFs. It also enables the use of genome sequence data, in combination with mRNA expression data, to enhance the sensitivity and specificity of predicted interactions. The limitation of this approach is that it cannot describe regulatory control by mechanisms other than transcription factors.

A second strategy, which we call the "influence" approach, seeks to identify regulatory influences between RNA transcripts (Yeung et al. 2002). In other words, it looks for transcripts that act as "inputs" whose concentration changes can explain the changes in "output" transcripts. Each transcript may act as both an input and an output. The input transcripts can be considered the regulators of transcription. By construction, such a model does not generally describe physical interactions between molecules since transcription is rarely controlled directly by RNA (and never by messenger RNA, which is the type of RNA predominantly measured by DNA microarrays). Thus, in general, the regulator transcripts may exert their effect indirectly through the action of proteins, metabolites, and effects on the cell environment. Nevertheless, in some cases, the regulator transcripts may encode the TFs that directly regulate transcription. In such cases, the influence model may accurately reflect a physical interaction. An advantage of the influence strategy is that the model can implicitly capture regulatory mechanisms at the protein and metabolite level that are not physically measured. That is, it is not restricted to describing only

transcription factor/DNA interactions. As described in the section on differential equation models, an influence model may be advantageous when trying to predict the global response of the cell to stimuli. The limitation of this approach is that the model can be difficult to interpret in terms of the physical structure of the cell, and therefore difficult to integrate or extend with further research. Moreover, the implicit description of hidden regulatory factors may lead to prediction errors.

Linear Differential Equations for Gene Regulatory Network

In general, a genetic network can be expressed by a set of nonlinear differential equations. Almost all of the existing approaches for gene regulatory network inference use linear or additive models, primarily due to the complex structures of biological systems and the scarcity of data (Wang et al. 2006a, b, 2007). Furthermore, linear equations can capture the main features of the network near the steady state, and can provide a good starting point for further modeling and analysis.

Assume that there are N microarray datasets X_1, X_2, …, X_N with m_1, m_2, …, m_N time points, respectively, for one organism. These time-course datasets may be measured under various environments or stimuli by different labs. Let us first consider one time-course dataset with m time points. A linear differential equation can be used to represent the rate of synthesis of a transcript as a function of the concentrations of other transcripts in a cell and the external perturbations:

$$\frac{dx(t)}{dt} = Jx(t) + Pc(t), \quad t = t_1, t_2, ..., t_m \qquad (1)$$

where $x(t) = (x_1(t), \ldots, x_n(t))\ T \in R^n$, $x_i(t)$ is the expression level (mRNA concentrations) of gene i at time point t. $J = (J_{ij})_{n \times n}$ is an $n \times n$ connectivity matrix with elements J_{ij} representing the effect of gene j on gene i with a positive, zero, or negative sign, indicating activation, no interaction, and repression, respectively. $P = (P_{ij})_{n \times s}$ is an $n \times s$ matrix representing the effect of the s perturbations or s small molecules on x, and $c(t) \in R^s$ represents the external perturbations with s compounds at time t. (In principle, the external perturbation can be of virtually any type, for example, an external environmental factor, a small molecule, an enzyme, a microRNA, or a post-translationally modified protein.) A non-zero element P_{ij} of P implies that the i-th gene is a direct target of the j-th perturbation or compound. Identifying P is an important first step toward biological function discovery of small molecules and drug design.

We can rewrite Eq. 1 in a compact form for all time points of one dataset by matrix notation:

$$\frac{d\mathbf{X}}{dt} = J\mathbf{X} + P\mathbf{C} \qquad (2)$$

where $X = (x(t_1),\ldots,x(t_m))$ and $dX/dt = (dx(t_1)/dt,\ldots, dx(t_m)/dt)$ are $n \times n$ matrices with the first derivative of mRNA concentration $dx_i(t_j)/dt = [x_i(t_j + 1) - x_i(t_j)]/[t_{j+1} - t_j]$ for $i = 1,\ldots,n$; $j = 1,\ldots,m$. Although the forward difference approximation here is utilized for numerical computation of dx/dt, backward or other difference approximation methods can be applied similarly. Suppose that there are s external perturbation compounds, then $C = (c(t_1),\ldots,c(t_m))$ is an $s \times m$ matrix representing the s perturbations. The unknowns to be calculated are connectivity matrix J and P.

Cross-References

- Combinatorial Transcription Regulatory Network
- Continuous Model
- Linear Model
- MicroRNA-embedding Regulation Networks, Logical Modeling
- Ordinary Differential Equation (ODE)
- Regulation
- Reverse Engineering

References

Brazma A, Jonassen I, Vilo J, Ukkonen E (1998) Predicting gene regulatory elements in silico on a genomic scale. Genome Research 8(11):1202

Chen L, Wang RS, Zhang XS (2009) Biomolecular networks: methods and applications in systems biology. Wiley, Hoboken

De Hoon MJL, Imoto S, Kobayashi K, Ogasawara N, Miyano S (2003) Inferring gene regulatory network from time-ordered gene expression data of bacillus subtilis using differential equations. Pac. Symp. Biocomput 17–28

Demeter J, Beauheim C, Gollub J, Hernandez-Boussard T, Jin H, Maier D (2007) The Stanford Microarray Database: implementation of new analysis tools and open source release of software. Nucl. Acids Res. 35(suppl_1):D766–770

Dewey TG, Galas DJ (2001) Dynamic models of gene expression and classification. Functional & Integrative Genomics 1(4):269–278

D'Haeseleer P, Liang, S, Somogyi R (2000). Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics 16(8):707-726
Friedman N (2004) Inferring cellular networks using probabilistic graphical models. Science 303(5659):799–805
Gardner TS, Faith JJ (2005) Reverse-engineering transcription control networks. Phys Life Rev 2(1):65–88
Gardner TS, di Bernardo D, Lorenz D, Collins JJ (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301(5629):102–105
Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, Danford TW (2004) Transcriptional regulatory code of a eukaryotic genome. Nature 431:99–104
Holter NS, Maritan A, Cieplak M, Fedoroff NV, Banavar JR (2001) Dynamic modeling of gene expression data. Proc Natl Acad Sci US A 98(4):1693
Husmeier D (2003) Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. Bioinformatics 19(17):2271–2282
Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK (2002) Transcriptional Regulatory Networks in Saccharomyces cerevisiae. Science 298(5594):799–804
Nachman I, Regev A, Friedman N (2004) Inferring quantitative models of regulatory networks from expression data. Bioinformatics 20(90001)
Tegner J, Yeung, MK, Hasty J, Collins JJ (2003) Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. Proc Natl Acad Sci USA 100(10):5944
Teixeira MC, Monteiro P, Jain P, Tenreiro S, Fernandes AR, Mira NP (2006) The YEASTRACT database: a tool for the analysis of transcription regulatory associations in Saccharomyces cerevisiae. Nucl. Acids Res. 34(suppl_1):D446–451
Wang Y, Joshi T, Zhang X-S, Xu D, Chen L (2006) Inferring gene regulatory networks from multiple microarray datasets. Bioinformatics 22(19):2413–2420
Wang Y, Joshi T, Xu D, Zhang, X, Chen L (2006) Supervised inference of gene regulatory networks by linear programming. Lecture notes in computer science 4115:551
Wang RS, Wang Y, Zhang XS, Chen L (2007) Inferring transcriptional regulatory networks from high-throughput data. Bioinformatics 23:8
Yeung MKS, Tegner J, Collins JJ (2002) Reverse engineering gene networks using singular value decomposition and robust regression. Proc Natl Acad Sci USA 99:6163–6168

Gene Set and Protein Set Expression Analysis

Roger Higdon
Seattle Children's Research Institute, Seattle, WA, USA

Synonyms

Gene set enrichment analysis

Definition

Gene set expression analysis (GSEA) often referred to as gene set enrichment analysis is based upon determining whether predefined sets of genes differ in their expression patterns in some way. This is opposed to conventional ▶ relative expression analysis which examines differences on a gene-by-gene by basis. Similarly, these methods can be applied to protein expression data from mass spectrometry (MS) proteomic analysis.

Characteristics

Utilizing information about predefined sets, typically taken from biochemical pathway databases such as KEGG, Panther, or Metacyc or from the Gene Ontology (GO), increases the ability to infer biological meaning from gene expression differences and can increase the power to detect differences by combining data across related genes. Originally, analyses were based on data from gene expression microarrays; more recently other technologies such next-gen sequencing are being used to measure gene expression. A number of comparisons and reviews of GSEA analyses have been published (Nam and Kim 2008; Emmert-Streib and Glazko 2011). Additionally, the methodology is being applied to protein set expression analysis (PSEA) through the use of MS proteomics data.

The simplest approach to GSEA analysis was based on chi-square or ▶ Fisher's tests. A criteria was chosen for differential expression (e.g., ER > 2, P-value $< .05$) and the proportion of differentially expressed (or over or under expressed) genes was compared across gene sets as can be seen in Table 1. ▶ Fisher's test could then be used to calculate a p-value for comparing the proportion of differentially expressed genes in a particular gene set to the proportion in the remaining sets based on the hypergeometric distribution in (1).

$$P - value = P(X) = \frac{\binom{P}{X}\binom{N-P}{D-X}}{\binom{N}{D}} \quad (1)$$

This approach has two major flaws: The first, Fisher's Exact test assumes expressions from genes within

Gene Set and Protein Set Expression Analysis, Table 1 Using Fisher's Exact test for pathway enrichment

	Differentially expressed	Not differentially expressed
Genes in pathway	X	P-X
Genes not in pathway	D-X	N + X-D-P

gene sets are uncorrelated with each other. This is unlikely since gene sets are chosen precisely because the genes are related in some biological manner. Hence, there is usually positive correlation among genes and therefore Fisher's Exact test is often highly anticonservative. In addition, Fisher's Exact test dichotomizes a continuous variable (Expression ratio or t-statistic) and therefore is less powerful than methods that take advantage of the continuous data.

One of the most widely used approaches for gene set analysis is the gene set enrichment analysis (GSEA) approach (Subramanian et al. 2005). In this approach, genes are ordered by a differential expression statistic (typically a ▶ Student's t-Test), then for each gene set the distributions of expression statistics are compared between those in the set and those not in the set using the Kolmogorov-Smirnov test statistic. To calculate p-values and ▶ false discovery rates (FDR), a permutation test of samples is used in order to account for effects of correlation within sets. The Kolomogorov-Smirnov test has been criticized for lacking power; therefore, others have proposed gene set analysis approaches that utilize other statistical tests, such as the GSA approach of Efron and Tibshirani (2007).

More generally, GSEA approaches have been broken down into self-contained and competitive approaches. Self-contained approaches compare individual gene sets across conditions without regard for other gene sets. On the other hand, competitive methods measure relative differences between gene sets. Examples of competitive tests are enrichment tests such as Fisher's exact test described above, e.g., find sets with more or fewer differentially expressed genes. A number of methods are a mixture of these two approaches. Although, there are a few methods based on parametric tests, most GSEA approaches are based on permutation or randomization tests of samples or genes. Randomizing samples preserves correlation structures and therefore leads to p-values that are less biased than gene randomization. However, sample randomization is not useful with very small sample sizes. Some argue that using self-contained methods with sample randomization is the only approach to ensure statistically valid results (Goeman and Buhlmann 2007). Others contend it is better to use multiple approaches in order to get more information out the analyses since competitive and self-contained methods test different hypotheses (Tian et al. 2005).

Although methods are applicable to proteomics data, PSEA has unique difficulties of its own. Typically, sample sizes are quite small, so the use of sample permutation or randomization tests is problematic. Also, unlike microarray data which assays an entire genome, MS proteomics usually assay only a fraction of the proteome. This results often in sparse coverage of proteins sets such as biochemical pathways.

Cross-References

- ▶ False Discovery Rate (FDR)
- ▶ Fisher's Test
- ▶ Relative Expression Analysis
- ▶ Student's t-Test

References

Efron B, Tibshirani R (2007) On testing the significance of sets of genes. Ann Appl Stat 1:107–129

Emmert-Streib F, Glazko GV (2011) Pathway analysis of expression data: deciphering functional building blocks of complex diseases. PLoS Comput Biol 7(5):e1002053

Goeman JJ, Buhlmann P (2007) Analyzing gene expression data in terms of gene sets: methodological issues. Bioinformatics 23:980–987

Nam D, Kim S (2008) Gene-set approach for expression pattern analysis. Brief Bioinform 9:189–197

Subramanian A, Tamayo P, Mootha VK et al (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci USA 102:15545–15550

Tian L, Greenberg SA, Kong SW et al (2005) Discovering statistically significant pathways in expression profiling studies. Proc Natl Acad Sci 102:13544–13549

Gene Set Enrichment Analysis

- ▶ Gene Set and Protein Set Expression Analysis

Gene Sets for Pathways

Michael F. Ochs
Department of Oncology, Johns Hopkins University, Baltimore, MD, USA

Definition

A gene set is a grouping of genes that function in a coordinated manner to provide some biological behavior. Many biological processes are controlled by pathways, where proteins encoded by genes create a directional though potentially circular flow, such as creation of a series of metabolites (metabolic pathway) or an ordered series of post-translational modifications (signaling pathway). Gene sets that serve as surrogate indicators for activity in biological pathways must measure the appropriate targets, which differ for each type of pathway.

Characteristics

Biological phenotypes arise from the coordinated actions of numerous biomolecules and structures in an organism. The coordination, such as during organismal development, is controlled by master regulators, often cell signaling networks responding to environmental and endocrine clues coupled to transcriptional and translational controllers. Often the change that drives a phenotype, such as the development of a specific tissue type, is the activation of a set of genes, thus a coordinated change in the transcript levels of many genes simultaneously. Alternatively, the change could be coordinated post-translational modifications of a set of proteins modifying their activity or localization, the increase in flux through a metabolic pathway leading to a buildup or excretion of a metabolite, or the breakdown of cellular structures as occurs during apoptosis.

Gene sets provide an approach for an analysis to gain insight into these drivers of phenotype through identification of small changes in many biomolecular levels or states rather than through identification of a change in a single gene, protein, or metabolite. For example, all the transcriptional products produced by the activation of the RAS-RAF-MEK-ERK signaling pathway could be used as a gene set of transcriptional products, i.e., messenger RNA (mRNA) levels, that provide a surrogate measurement for pathway activation. The fundamental assumption for any gene set is that changes in the measured biomolecules will be coordinated.

The most common gene sets presently comprise mRNA levels expected to undergo coordinated change and measurable on a gene "expression" microarray. The inherent assumption therefore is that these genes are coregulated by a series of transcriptional activators. A gene set could also be created to monitor activation of a signaling pathway through phosphoprotein levels, as most signaling proteins change activity based on phosphorylation of specific amino acid residues. This would effectively be a protein post-translational modification set. A further "gene" set could be created from metabolomic measurements, where the amounts of metabolites along a metabolic pathway could provide a surrogate measurement for enzyme activity. This type of gene set could provide insight into metabolic syndromes and diseases, including diabetes.

For a gene set to serve as a surrogate for measurement of activity on a biological pathway, changes in the levels measured by the set must be linked to changes in the pathway. Unfortunately, the term "pathway" is used quite loosely biologically, primarily for historical reasons. Initially, the term pathway was used quite literally, indicating nerve conduction, a viral invasion course, or auditory path. Later, it referred to the metabolic pathways that created biomolecules in cells (Carson and Frischer 1966), effectors for hormonal control (Samuels 1964), and gene interactions and epistasis (Avery and Wasserman 1992). For gene set enrichment analysis, we typically take a pathway to indicate a linked series of biomolecules: (1) a series of proteins that are enzymes and the metabolic products produced by them (a metabolic pathway), (2) a series of proteins that act together to transduce a signal when post-translationally modified (a signaling pathway), or (3) a series of transcriptional regulators produced in series (a transcriptional regulatory network). For each of these, different gene sets are needed to serve as surrogate markers of activity.

A gene set for a metabolic pathway could take three forms, depending on the biomolecule measured. The most common measurement remains the mRNA levels from a microarray. In this case, a pathway, such as provided by the Kyoto Encyclopedia of Genes and Genomes (KEGG), would be converted to just the

enzymes and then to the genes encoding the enzymes. The gene set would comprise all the genes whose protein products (i.e., enzymes) are necessary to take the metabolic precursor chemical to its final form. The assumption is that if the cell needs to produce more of this final chemical form, it will coordinately upregulate all the genes to produce all enzymes in the pathway. This is an indirect measurement of the levels of the proteins, where the protein levels are assumed to track the mRNA levels. It is worth noting that this is not true in eurkaryotic organisms in general, although it may be valid in this limited case of metabolic enzymes. Alternatively, the gene set could be proteomic measurements of the enzymes themselves, indicating the concentration of the catalysts that drive the chemical changes. Finally, the gene set could be the set of metabolites produced, measured by metabolomic technologies such as mass spectrometry. Here, the direct metabolic flux through the pathway would be measured. Note that this progression is from less direct to more direct measurements of the pathway activity, but also from more mature to less mature technological platforms.

A gene set for a signaling pathway could take two forms. The first would be the direct measurement of the amount of phosphoproteins and unmodified proteins along the pathway. As with the metabolites in a metabolic pathway, this would effectively be a measurement of the flux through the pathway, here providing the strength of the signal. The second is more complex, involving mRNA levels. The complexity arises from the fact that signaling is driven not by protein levels but by post-translational modification of proteins and the duration of these modifications, the specific modifications made, and the number of proteins with these modifications. As such, the mRNA levels for genes encoding the proteins become an inadequate surrogate for signaling activity. However, most, though not all, signaling pathways lead to changes in the activity of transcription factors (TFs). These TFs have transcriptional gene targets, and the mRNA levels of these targets are directly modified by the transcription factors, either increased (activated) or decreased (repressed). The coordinated changes of the transcription factors downstream of a signaling pathway then provide a surrogate measure of pathway activity. It is critical to note that the gene sets available in databases ignore this complication, however, so that the gene set for a KEGG signaling pathway is typically just the genes encoding the signaling proteins, which is the incorrect set if the goal is a surrogate of pathway activity.

A gene set for a transcriptional regulatory network (TRN) also relies on the relationship between a transcription factor (TF) and its target genes. A TRN is essentially a cascade of TFs linked through the gene of a TF being a transcriptional target of an upstream TF. Then the initial activation of a single TF can lead to activation of additional TFs, which then may lead to activation of further TFs, etc. The relationship of genes sets for this series and for a signaling pathway is obvious. In both cases, the gene targets of a TF are the key to constructing an appropriate gene set. In the case of a TRN, there is an additional complication that arises due to the timing of the measurements. If time resolution is high, then the data can potentially separate the first round (i.e., primary) of transcription from later (i.e., secondary) transcription, and a true regulatory network of relationships could be used, with each TF's signature isolated. However, if a time resolution is low or, in the extreme case, only a single measurement is made, then the appropriate gene set may be all the genes regulated by any member in the TRN. Effectively, the superset of all TF gene sets in the TRN would be the appropriate gene set to serve as a surrogate for TRN activity.

A gene set is typically tested for significance using a rank sum test upon a statistic generated for each gene, protein, or metabolite during preprocessing. This initial individual biomolecular (e.g., gene) statistic relies on comparing two or more phenotypic groups, such as through a *t*-test or Wilcoxon rank sum test. The members of the set are compared to the non-set members also measured in the experiment, and significance is determined by permutation tests on the set labels or by the assumption of a distribution on all sets. The final significance for the gene set must be adjusted for multiple testing, using either family-wise error rate or false discovery rate corrections. Family-wise error rates are more strict, but if the goal of a study is hypothesis-generation for more focused testing, then false discovery rate methods provide a greater chance of finding a novel hypothesis.

The complexity of defining gene sets argues that sets must be chosen with an understanding of biology and with a determined goal in mind. The set must conform to known biological behavior, so that the set is an appropriate surrogate for the pathway of interest. For example, using the genes encoding the proteins in a signaling pathway is an incorrect set if the goal is

a measure of pathway activity. Limiting the sets tested to those that are of interest is also useful for minimizing the effect of multiple testing, since testing all existing sets can lead to thousands of tests, reproducing the multiple testing problem of individual gene measurements.

Future gene sets are likely to involve integration of different molecular species. For instance, studies have already integrated mRNA levels, protein levels, and metabolite levels into a coherent biological picture. Gene sets that comprise different types of measurements and molecules will likely replace the present gene sets as such measurements become more ubiquitous. It will be critical when constructing such sets to carefully reproduce the biological relationships between molecular species, in order to construct meaningful sets.

Cross-References

- ▶ Biological Disease Mechanism Networks
- ▶ Biomarkers
- ▶ Co-expression
- ▶ Disease Classification or Discrimination
- ▶ Functional Enrichment Analysis
- ▶ Gene Expression Biomarkers
- ▶ Inference
- ▶ KEGG PATHWAY
- ▶ MicroRNA-mRNA Regulation Networks
- ▶ Network-Based Biomarkers
- ▶ Pathway, Functional Units
- ▶ Post-translational Modifications
- ▶ Protein Interaction Network
- ▶ Signal Transduction Pathway
- ▶ Transcriptional Regulatory Network
- ▶ Transcriptional Reprogramming

References

Avery L, Wasserman S (1992) Ordering gene function: the interpretation of epistasis in regulatory hierarchies. Trends Genet 8(9):312–316

Carson PE, Frischer H (1966) Glucose-6-phosphate dehydrogenase deficiency and related disorders of the pentose phosphate pathway. Am J Med 41(5):744–761

Samuels LD (1964) Actinomycin and its effects. Influence on an effector pathway for hormonal control. N Engl J Med 271:1301–1308, CONCL

Gene Silencing

Yan Zhang
Key Laboratory of Systems Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai, China

Definition

Gene silencing refers to a mechanism by which cells shut down large sections of chromosomal DNA. It is generally used to describe the "switching off" of a gene by a mechanism other than genetic modification. That is, a gene which would be expressed (turned on) under normal circumstances is switched off by machinery in the cell. Gene silencing is done by incorporating the DNA to be silenced into a form of DNA called heterochromatin that is already silent.

Characteristics

Gene silencing is a general term describing epigenetic processes of gene regulation. This process is important for the differentiation of many different types of cells.

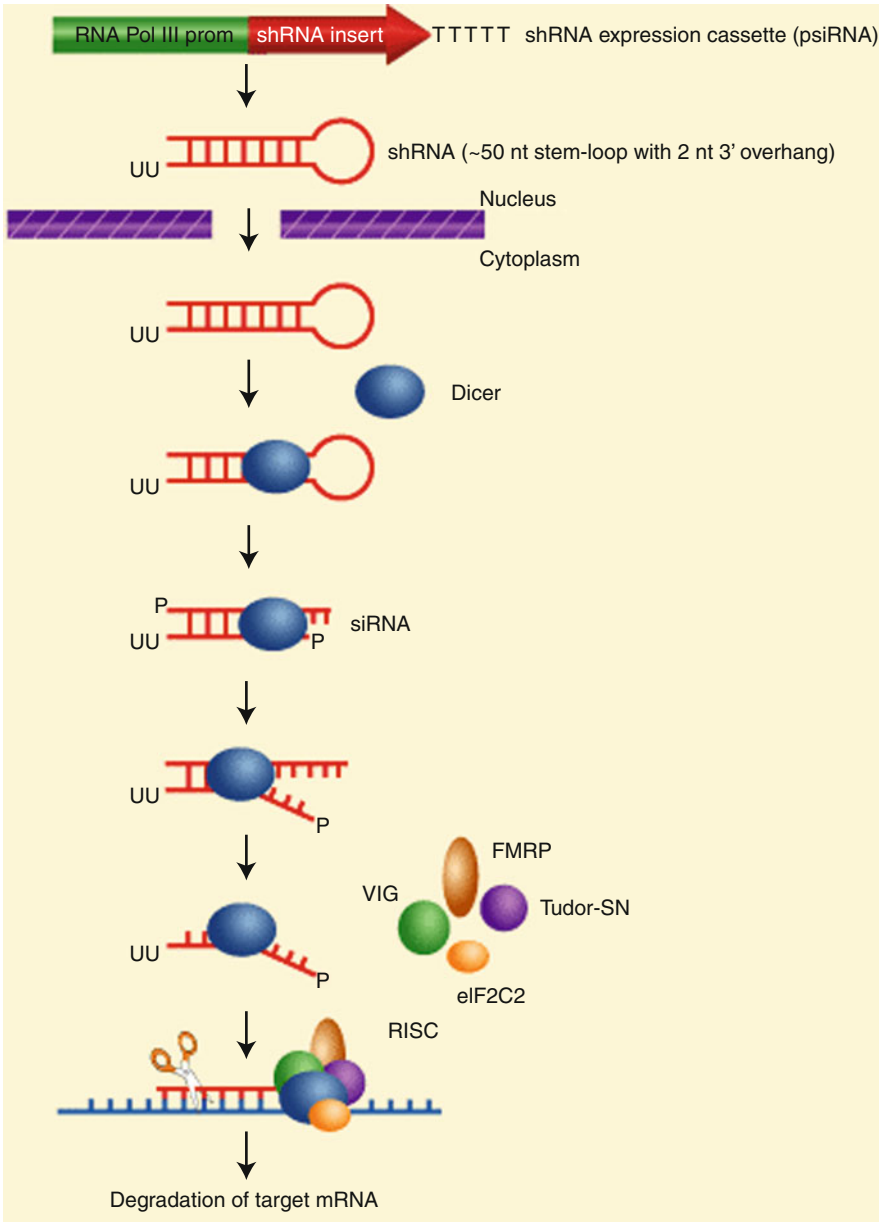
Genes are regulated at either the transcriptional or post-transcriptional level.

Transcriptional gene silencing is the result of histone modifications, creating an environment of heterochromatin around a gene that makes it inaccessible to transcriptional machinery (RNA polymerase, transcription factors, etc.).

Post-transcriptional gene silencing is the result of mRNA of a particular gene being destroyed or blocked. The destruction of the mRNA prevents translation to form an active gene product (in most cases, a protein). A common mechanism of post-transcriptional gene silencing is RNAi (Fig. 1).

Both transcriptional and post-transcriptional gene silencing are used to regulate endogenous genes. Mechanisms of gene silencing also protect the organism's genome from transposons and viruses. Gene silencing thus may be part of an ancient immune system protecting from such infectious DNA elements.

Genes may be silenced by DNA methylation during meiosis, as in the filamentous fungus *Neurospora crassa*.



Gene Silencing, Fig. 1 RNAi-mediated gene silencing in mammals using short hairpin RNA genes

Cross-References

► Computational microRNA Biology

References

Bass BL (2000) Double-stranded RNA as a template for gene silencing. Cell 101(3):235–238

Cogoni C, Macino G (2000) Post-transcriptional gene silencing across kingdoms. Curr Opin Genet Dev 10(6):638–643

Selker EU (1999) Gene silencing: repeats that count. Cell 97(2):157–160

Gene Type

► IMGT-ONTOLOGY, GeneType

Gene-centered Information Resource, GoGene

Conrad Plake
Biotechnology Center (BIOTEC), Technische Universität Dresden, Dresden, Germany

Definition

GoGene is a publicly accessible web server supporting the task of searching for genes and gene-related molecular functions, biological processes, cellular components, single amino acid substitutions, and diseases (Plake et al. 2009). Searching is carried out either via the literature database Pubmed (▶ MEDLINE and PubMed), the Blast service at the European Bioinformatics Institute (EBI), or directly via the NCBI Entrez Gene database. The resulting list of genes is presented together with the relevant parts of the ▶ Gene Ontology and Medical Subject Headings, two controlled vocabularies that cover a broad variety of biomedical research. GoGene is located at: http://www.gopubmed.org/gogene.

Characteristics

Gene Annotation by Automated Literature Mining

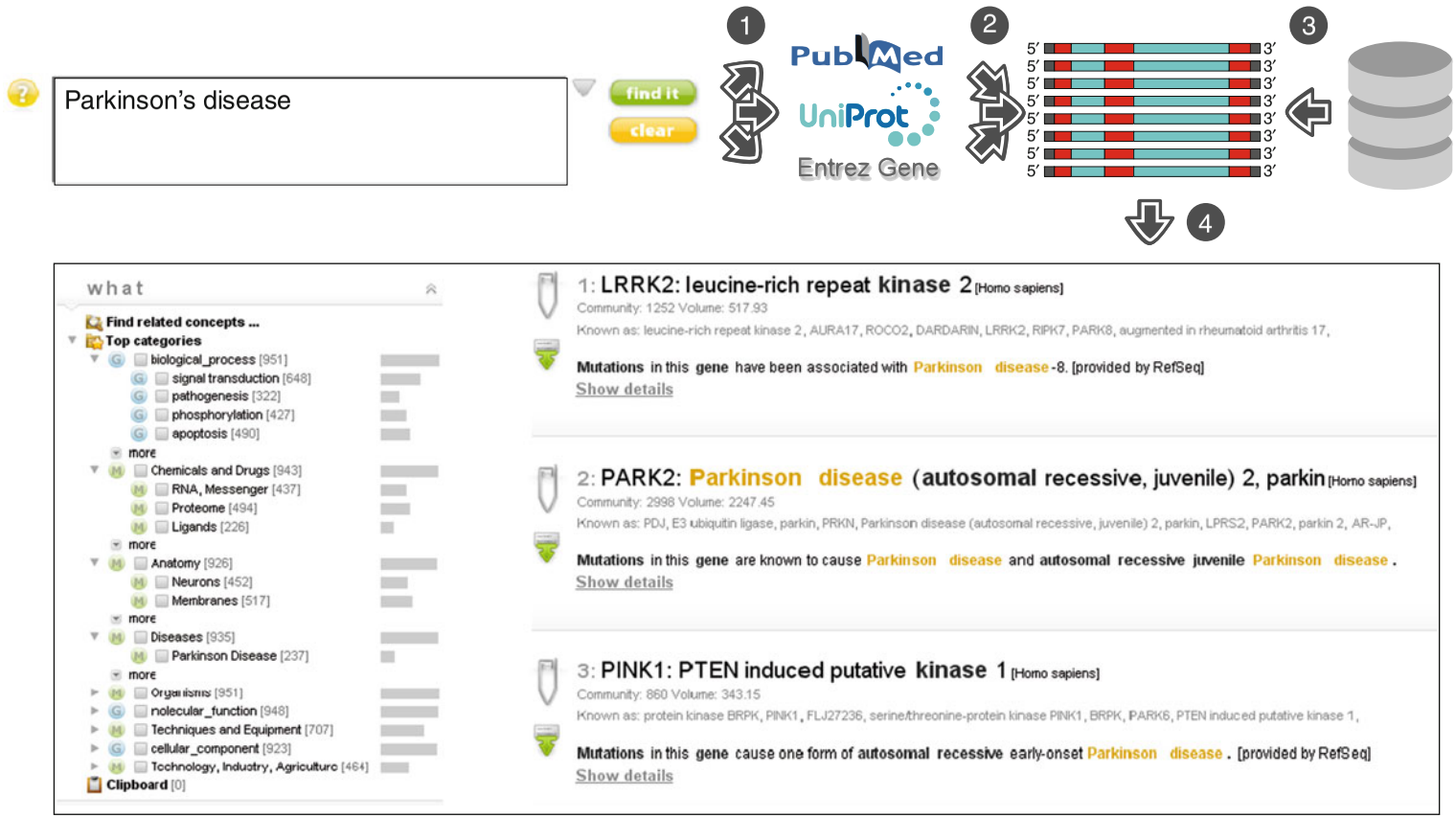
Sequence databases are growing and many entries are lacking proper annotation (Baumgartner et al. 2007). The low coverage of gene and protein annotation has significant impact on turning experimental data into knowledge such as of mechanisms governing biological processes within cells and organisms, or of pharmacodynamic pathways determining the action mechanisms of drugs. With thousands of scientific articles published every day, the literature constitutes the main resource of biomedical knowledge. This knowledge is not easily accessible as it requires human experts to read papers and manually add the relevant pieces of information to a database. GoGene has been developed to support literature-based gene annotation. It employs the gene mention identification tool GNAT (▶ Gene Normalization with GNAT) and the web server GoPubMed (▶ Search Engines with Faceted Search) to associate genes from various model organisms (▶ Model Organism) to concepts of the Gene Ontology (GO) and to diseases based on millions of citations in PubMed. With these associations, GoGene supports interpretation of results from high-throughput experiments or simply searching the literature for discussed genes. A sequence query lets users retrieve genes with similar gene or protein sequences and explore their GO and disease annotation, for example, to find functional hints for yet uncharacterized genes.

Usage

Users of GoGene can choose between three types of queries: (1) lists of gene identifiers or gene names as in Entrez Gene, (2) keywords that are forwarded to PubMed to find genes mentioned in the resulting list of citations, and (3) nucleotide or amino acid sequences that are forwarded to the EBI, where a remote Blast search against all sequences in the Swiss-Prot database is invoked. The query type can either be specified by the user or GoGene tries to automatically find the best result. First, it checks if the query resembles a nucleotide or amino acid sequence, in which case the BLASTX or BLAST service is invoked, respectively. Otherwise, the query is forwarded to both PubMed and Entrez Gene and the larger gene list is returned. For Entrez Gene results, genes are ranked as in the original result list. Results from a PubMed search are ranked by occurrence frequency in the matching abstracts in descending order. A gene list resulting from a Blast search is ranked by sequence similarity, with the most similar gene listed first. Figure 1 shows the flow of data happening after the user submits a query. The resulting list of genes is presented together with the relevant parts of the GO and MeSH (Medical Subject Headings), which support navigation similar to a hyperlinked table of contents. A gene list, including all GO and disease annotations, can also be downloaded as a file in different standard formats.

Use Case

Please note that the following search results might not be reproducible anymore due to updated database contents. Consider a biologist who is interested in rat genes related to osteoporosis and bone resorption. A keyword search for “osteoporosis bone resorption” in the Rat Genome Database gives no results. The same



Gene-centered Information Resource, GoGene, Fig. 1 GoGene accepts three types of queries: keywords, sequences, and gene identifiers. Queries are sent either to PubMed, UniProt, or Entrez Gene (1). If the destination for a query was not specified by the user, GoGene determines automatically which database gives the best result. After a list of genes has been retrieved (2), more details about each gene, taken from a local annotation store, are added such as descriptive summaries, literature references, and ontological concepts describing gene functions, and their role in biological processes and diseases (3). The list of genes is then displayed in GoGene (4). Concepts from the GO and MeSH for all genes are displayed as a tree to the left

query in Entrez Gene results in two rat genes (*Pth* and *Tnfsf11*). In the literature database PubMed, the query "rats osteoporosis bone resorption" returns 857 citations. Reading their abstracts to identify the relevant genes is cumbersome. GoGene alleviates this task by automatically searching these abstracts for mentioned genes and displaying the resulting gene list to the user. In the tree, the biological process, bone resorption, the organism, rat, and the disease, osteoporosis, are listed as top categories. Selecting each as mandatory augments the query and eventually leaves five rat genes (*Pth, Tnfsf11, Ctsk, Tnfrsf11b*, and *Csf1*) for which literature references state their role in the specified disease and biological process.

Methods

Prerequisite for annotating genes with concepts from biomedical ontologies by automated literature analysis is the correct identification of genes and concepts in text. For gene identification, GoGene employs GNAT, a tool for gene mention recognition and normalization to a gene database (▶ Named Entity Recognition, ▶ Entity Mention Normalization). Having defined the gene literature in PubMed using GNAT, GoGene then associates these publications with biomedical concepts provided by the GoPubMed web server. This also includes all MeSH concepts assigned by the US National Library of Medicine for indexing. For each concept found, the corresponding citation is also associated to all its ascendants according to the GO or MeSH disease hierarchy. For instance, if a PubMed citation is associated with pancreatic neoplasms, then it is also associated with digestive system neoplasms, neoplasms by site, neoplasms, and diseases. A relationship between a gene and a GO or disease concept is established based on co-occurrences within PubMed citations. The confidence in such a relationship is computed as the log ratio of observed co-occurrence probability to the co-occurrence probability expected under independence, or pointwise ▶ mutual information (Manning and Schuetze 1999).

Related Tools

Tools and web servers performing tasks similar to GoGene are ▶ Alibaba, EBIMed (▶ Retrieving and Extracting Entity Relations from EBIMed), and iHOP.

Cross-References

- ▶ Alibaba
- ▶ Entity Mention Normalization
- ▶ Gene Normalization with GNAT
- ▶ Gene Ontology
- ▶ MEDLINE and PubMed
- ▶ Model Organism
- ▶ Mutual Information
- ▶ Retrieving and Extracting Entity Relations from EBIMed
- ▶ Search Engines with Faceted Search

References

Baumgartner WA, Cohen KB, Fox LM, Acquaah-Mensah G, Hunter L (2007) Manual curation is not sufficient for annotation of genomic databases. Bioinformatics 23(13):i41–i48

Manning CD, Schuetze H (1999) Foundations of Statistical Natural Language Processing, 1st edn. The MIT Press, Cambridge, MA, USA. ISBN 0262133601

Plake C, Royer L, Winnenburg R, Hakenberg J, Schroeder M (2009) GoGene: gene annotation in the fast lane. Nucleic Acids Res 37(Web server issue):W300–W304. doi:10.1093/nar/gkp429. http://dx.doi.org/10.1093/nar/gkp429

Gene-External Type of Promoter

- ▶ Type 3 Promoters

General Transcription Factors

Tetsuro Kokubo
Department of Supramolecular Biology, Graduate School of Nanobioscience, Yokohama City University, Yokohama, Kanagawa, Japan

Synonyms

Basal transcription factors

Definition

In eukaryotes, RNA synthetic activities are performed by three distinct types of RNA polymerases

(RNAPI, II, and III) that demonstrate different sensitivities to α-amanitin, a toxic substance from the mushroom (*Amanita phalloides*). These RNAPs alone cannot bind to the core promoter, a DNA region surrounding the transcription initiation site of a given gene. Thus, other proteins are essential for transcription initiation at the specific site of the core promoter. The minimal set of such proteins was purified by chromatography for each RNAP and designated as GTFs (general transcription factors) (Hampsey 1998; Orphanides et al. 1996; Thomas and Chiang 2006). The main functions of GTFs are to recognize core promoter structures, recruit RNAP to the core promoter, and regulate transcription in response to activators and repressors.

Cross-References

▶ Transcription in Eukaryote

References

Hampsey M (1998) Molecular genetics of the RNA polymerase II general transcriptional machinery. Microbiol Mol Biol Rev 62(2):465–503

Orphanides G, Lagrange T, Reinberg D (1996) The general transcription factors of RNA polymerase II. Genes Dev 10(21):2657–2683

Thomas MC, Chiang CM (2006) The general transcription machinery and general cofactors. Crit Rev Biochem Mol Biol 41(3):105–178

Generalization

▶ Abstraction

Generalized Additive Models

Roger Higdon
Seattle Children's Research Institute, Seattle, WA, USA

Definition

Generalized additive models are an extension of additive ▶ generalized linear models where the linear predictors are replaced by smooth nonlinear functions.

Characteristics

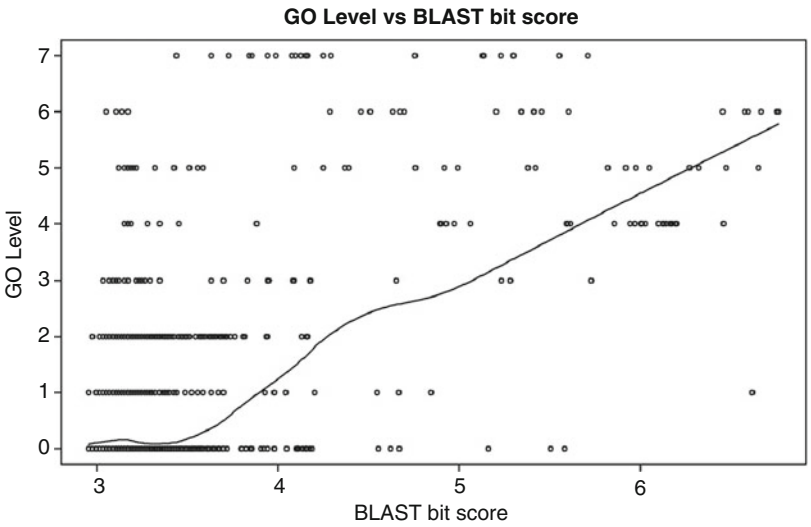
Generalized additive models (GAMs) were developed by Hastie and Tibshirani (1990) and presented in a similar manner to ▶ generalized linear models (GLMs) where a function of the mean (the link function) is modeled as a linear combination of smooth functions of explanatory or predictor variables.

$$g(\mu) = \beta + f_1(x_1) + \ldots + f_m(x_m) \quad (1)$$

GAMs typically use the same underlying distributions for the data as GLMs, most commonly normal, binomial, or Poisson. The functions $f_i(x_i)$ are most commonly fit using non-parametric curve estimating methods such as local-regression (lowess, Cleveland and Devlin 1988), splines, or smoothing splines. These methods potentially provide better fits to the data than linear or polynomial models and allow the detection of features in the data that may be missed. The R package GAM is one of the most popular software tools for estimating GAMs.

An example of the use of GAMs in systems biology is modeling functional similarity as a smooth function of sequence similarity (Louie 2009). In the example shown in Fig. 1, sequence similarity is measured using the BLAST bit score. Functional similarity is based on using the Gene Ontology (GO, Ashburner et al. 2000) and determining the shared level for two experimentally validated proteins.

One drawback of GAMs is the possibility of over-fitting the data; therefore, cross-validation methods should be used when fitting GAMs. Other drawbacks include difficulty interpreting the models and the inability to add interactions between variables.



Generalized Additive Models, Fig. 1 Example of generalized additive models. A comparison of shared GO annotation level as function sequence similarity for experimentally validated proteins. Sequence similarity is measured by the BLAST bit score. The relationship is model using generalized additive model where the curve is estimated by lowess regression

Cross-References

► Generalized Linear Models

References

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H et al (2000) Gene ontology: tool for the unification of biology. Nat Genet 25:25–29

Cleveland WS, Devlin SJ (1988) Locally-weighted regression: an approach to regression analysis by local fitting. J Am Stat Assoc 83(403):596–610

Hastie TJ, Tibshirani RJ (1990) Generalized additive models. Chapman & Hall/CRC, New York/Boca Raton

Louie B et al (2009) A statistical model of protein sequence similarity and function similarity reveals overly-specific function predictions. PLoS One 4:e7546

Generalized Linear Models

Larissa Stanberry
Bioinformatics and High-throughput Analysis Laboratory, Seattle Children's Research Institute, Seattle, WA, USA

Synonyms

GLM

Characteristics

Definition

In classical Linear Regression, the data mean is modeled as a linear combination of the covariates. The error is assumed to be normally distributed with constant variance. Consequently, the mean as well as the combination of the covariates can take any value on the real line and the modeling approach is plausible. However, for skewed or categorical data, the model is not always appropriate. For example, count or proportion data has the mean restricted to a certain interval so the additive model for the mean is inadequate.

Generalized linear models (GLMs) describe the mean as a function of the linear combination of the covariates. The function maps the range of the covariate combinations into the domain of the mean. Generalized linear models are characterized by three components: a random component, a systematic component, and a link function. The random component of the GLM model is given by independent realizations $(y_1, \ldots, y_n)$ of response Y and its probability distribution. The distribution of Y is assumed to belong to the exponential family and can be written as:

$$f(y; \theta, \phi) = \exp(y\theta - b(\theta))/a(\phi) + c(y, \phi),$$

for some functions $a(\phi)$, $b(\theta)$, $c(y,\phi)$. For example, normal, Poisson, binomial, and gamma distributions

belong to the exponential family. For the distributions in the exponential family, the mean and variance can be expressed as $E(Y) = \mu = b'(\theta)$ and $var(Y) = b''(\theta)a(\phi)$.

The systematic component refers to explanatory variables and constitutes a linear predictor:

$$\eta_i = \sum_{j=1}^{p} \beta_j x_{ij}, \quad i = 1, \ldots, n,$$

where $(x_{i1}, \ldots, x_{ip})$ are the covariates for observation y_i and $(\beta_1, \ldots, \beta_p)$ is the vector of model parameters.

The link function $g(\cdot)$ connects the systematic component, η_i, to the random component, $\mu_i = E(Y_i)$:

$$\eta_i = g(\mu_i) = \sum_{j=1}^{p} \beta_j x_{ij}.$$

The canonical link satisfies $g(\mu) = \theta$.

Examples

Linear Regression is an example of the generalized linear model with random components Y assumed to be independently normally distributed with means μ_i and variance σ^2; the systematic component $\eta_i = \sum_{j=1}^{p} \beta_j x_{ij}, \quad i = 1, \ldots, n$ and the identity link $g(z) = z$ so that $\mu_i = \eta_i$. The identity is also the canonical link for the normal model.

For the binomial model, the link functions include the logit $\eta = \log\mu/(1-\mu)$, the complementary log-log $\eta = \log\{-\log(1-\mu)\}$, and the probit given by the inverse of the normal distribution function. The logit function is the canonical link for the binomial model.

The Poisson and gamma models are also examples of GLMs with canonical links given by the log and inverse functions, respectively.

Model Estimation

The maximum likelihood estimates of the GLM parameters β_i, $i = 1, \ldots, p$ are obtained using iterated weighted least squares. The response variable is given by the first-order linear approximation of the link function at Y and the weights depend on the fitted values $\hat{\mu}$. The procedure iterates between computing a new estimate for η and β. The iterations stop when the change in model parameters is sufficiently small.

Cross-References

▶ Linear Regression

References

McCullagh P, Nelder JA (1989) Generalized linear models, 2nd edn. Chapman & Hall/CRC, Boca Raton

Generalized Mass Action System

Eberhard O. Voit
The Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology and Emory University, Atlanta, GA, USA

Definition

▶ Biochemical Systems Theory permits several variants. The format of a *Generalized Mass Action* (GMA) system models every process within a system as one product of power-law functions. As a result, every differential equation describing a dependent variable in a GMA system contains as many power-law terms as processes producing or degrading the variable.

Cross-References

▶ Biochemical Systems Theory (BST)
▶ Metabolic Systems Modeling, Power-Law Functions

General-Purpose Computation, Graphics Processing Units

Giuseppe Agapito
Department of Experimental Medicine and Clinic, University Magna Graecia of Catanzaro, Catanzaro, Italy

Synonyms

Data parallel; Distributed data access; Distributed data management; Integration technologies; Parallel computing

Definition

Graphical Processing Unit computing, or briefly GPU computing, is a methodology that aims at taking advantage from the computational power of graphics processor in the development of high-performance software for a wide range of scientific and engineering fields. GPUs can be considered as a computer device operating as a coprocessor to the main CPU (host). This cooperation between the GPU and CPU allows to increase the performance of CPU through the execution of time-consuming and computing intensive parts of the code on the GPU.

Characteristics

Both scientists and computer graphics designers need to process large volumes of data very quickly while modeling sets of many interrelated equations to produce results that are as realistic and reliable as possible. Many biological phenomena are extremely difficult to study experimentally and researchers must rely on computational simulations, i.e., to determine the protein 3D structure. This is one of the greatest challenges in computational biology. Nuclear magnetic resonance (NMR) spectroscopy and X-ray crystallography are the most popular methods for structure prediction. If all distances are known exactly, the coordinates for each atom can be easily computed. However, only a small fraction of all distance bounds are obtained by experiments conducted with these methodologies; thus, it is necessary to use heuristics to infer 3D structures from data. For this reason, it is needed to use massively parallel computing capability in order to obtain significant models. The high parallelism expressed by the interactions among the atoms leads to the idea of using parallel computing techniques to tackle the complexity of biological systems. Parallel computing techniques require dedicated architectures that can be classified into the following classes:

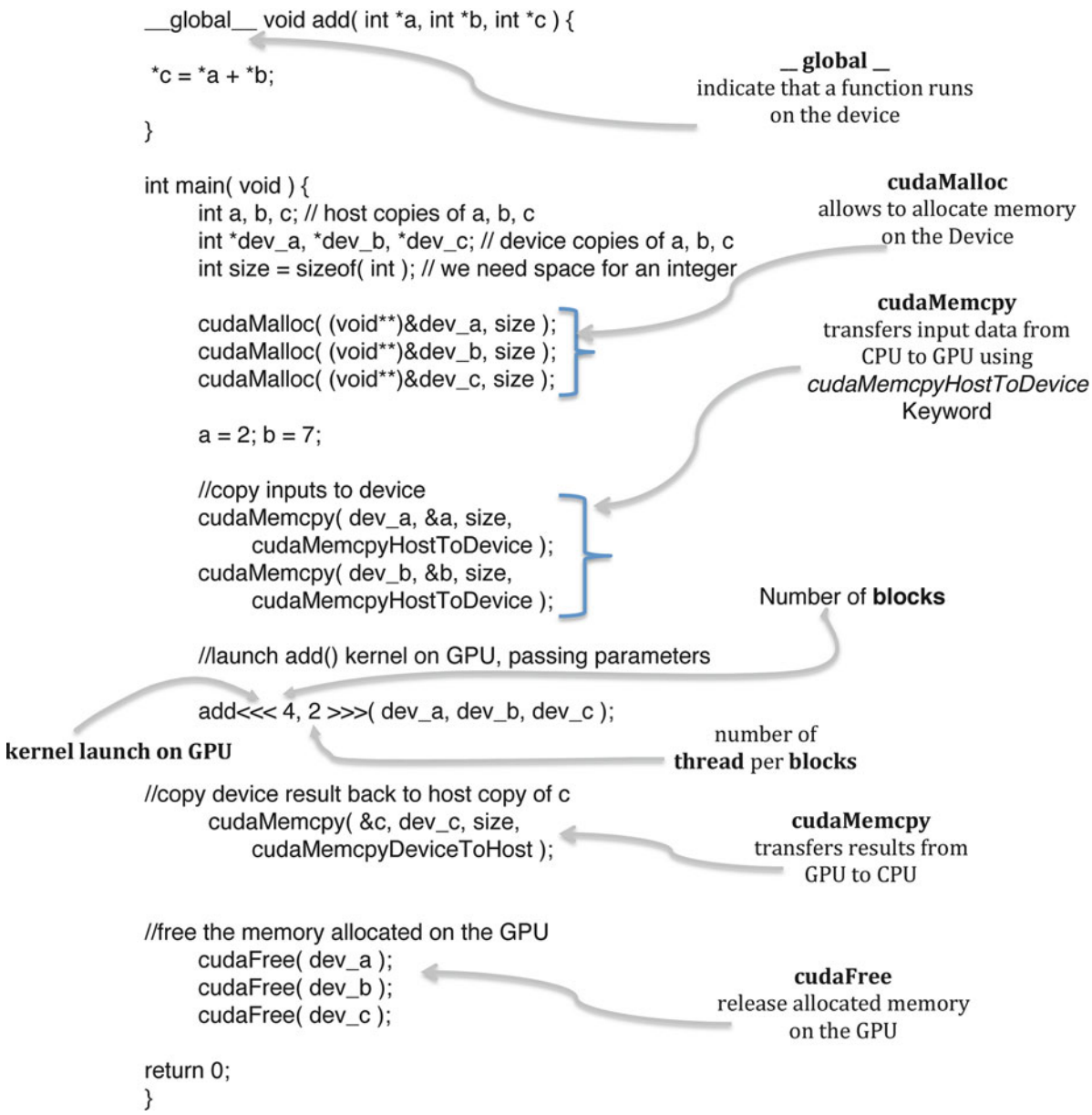
1. Single Instruction, Single Data (SISD) is a monoprocessor architecture that is able to execute exactly one instruction at a time.
2. Single Instruction, Multiple Data (SIMD) is a type of architecture in which many processing units execute the same instruction on different data elements.
3. Multiple Instruction, Single Data (MISD) is a type of architecture where independent processors perform different operations on the same data, this architecture does not have any practical application today.
4. Multiple Instruction, Multiple Data (MIMD) consists of multiple independent processors simultaneously executing different instructions on different data.

The modern GPUs could provide the high computational power needed to manage these huge amounts of data efficiently and relatively inexpensive compared to the number of multi-core CPUs needed to achieve a similar number of cores (Nickolls and Dally 2010).

The GPU's power comes from the sheer number of parallel processing cores on each chip which is typically higher respect to that of a CPU; moreover, the memory of a GPU is typically faster due to a larger bus width. This means GPUs can transfer information to and from their memory more quickly than CPUs, a frequent operation that could create a bottleneck for several applications. Finally, the GPU frequency is currently rapidly increasing, suggesting that GPUs will be able to process data at even faster rates in the future. These benefits are leading many scientists to choose GPUs over clusters of multi-core CPUs. The high parallelism of GPU is a feature needed to render complex graphical effects in 3D in the animations and games; in fact, GPU initially was developed to graphical purposes (Dematté and Prandi 2010).

Over the past few years, the GPU has evolved from a fixed-function processor into a general-purpose parallel programmable processor with additional special-purpose functionality.

The recent introduction of programming environments for the development of non-graphics applications on GPU facilitated the use of GPU for high-performance computations. The most widespread programming environment is Compute Unified Device Architecture (CUDA) introduced by NVIDIA. CUDA is a hardware and software co-processing architecture for parallel computing that enables NVIDIA GPUs to execute programs written in C, C++, Fortran, OpenCL, DirectCompute, and other languages, with the goal to exploit the parallelism of the GPU. CUDA programs are explicitly divided into code that runs on the CPU (host) and code that runs on the GPU (device). The data parallel portions of an application are executed on the device as kernels (functions). This is possible as all the architectural details (threads, warps, multiprocessor, etc.) are hidden to the end user. The notions available



General-Purpose Computation, Graphics Processing Units, Fig. 1 Example of a CUDA program

for the user in CUDA are blocks, grids, and threads, to ease the decomposition of the problem domain.

A CUDA program is organized into a host program, consisting of one or more sequential threads running on a host CPU, and one or more parallel kernels suitable for execution on a parallel computing GPU. In Fig. 1 some basic features of parallel programming with CUDA are shown where it is possible to note the affinity with C language.

A program can be organized into the following parts: (a) set up input data on the CPU; (b) transfer the data to the GPU by cudaMemcpy(dev_a, &a, size, cudaMemcpyHostToDevice), a function that allows to transfer input data from CPU to GPU by using as last parameter the cudaMemcpyHostToDevice keyword; (c) run the kernel on the GPU by the __global__ modifier indicating that the procedure is a kernel entry point, but the execution on

the GPU is done by the extended function call i.e., add < <<4, 2>> > (dev_a, dev_b, dev_c) launches the kernel add() in parallel across 4 blocks of 2 threads each; (d) finally, transfer the result back to the CPU by the cudaMemcpy(dev_a, &a, size, cudaMemcpyDeviceToHost) defining the direction from GPU to CPU by using the cudaMemcpyDeviceToHost keyword.

Cross-References

▶ Multicore Computing

References

Dematté L, Prandi D (2010) GPU computing for systems biology. Brief Bioinform 3:323–333
Nickolls J, Dally WJ (2010) The GPU computing era. IEEE 2:56–69

General-Purpose GPU

▶ GPU Computing

GENESIS: General Neural Simulation System

▶ Database of Quantitative Cellular Signaling (DOQCS)

Genes-to-Systems Breast Cancer Database

▶ Data Integration, Breast Cancer Database

Genetic Algorithm

▶ Evolutionary Algorithm, Transcription Regulatory Network Construction

Genetic Algorithms

Feng-Sheng Wang and Li-Hsunan Chen
Department of Chemical Engineering, National Chung Cheng University, Chiayi, Taiwan

Synonyms

Evolutionary algorithm

Definition

Genetic algorithms (GA) are adaptive heuristic search algorithms based on the conjecture of natural selection and genetics (Zhilinskas and Žilinskas 2008). The basic concept of genetic algorithms is designed to simulate processes in a natural system necessary for evolution, specifically those that follow the survival of the fittest inspired by Darwin's principle. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. Algorithm is started with a population of strings (represented by chromosomes or the genotype of the genome), which encodes candidate solutions (called individuals or phenotypes) to an optimization problem, and evolves toward better solutions. The typical steps are:

1. Choose an initial population of candidate solutions.
2. Calculate the fitness, how well the solution is, of each individual.
3. Perform crossover from the population. The operation is to randomly choose some pair of the individuals as parents and exchange so parts from the parents to generate new individuals.
4. Mutation is to randomly change some individuals to create other new individuals.
5. Evaluate the fitness of the offspring.
6. Select the survive individuals.
7. Proceed from 3 if the termination criteria have not been reached.

Example. Genetic algorithms have been applied in the fields of bioinformatics, computational biology, and systems biology (Larranaga et al. 2006; Handl et al. 2007).

References

Handl J, Kell DB, Knowles J (2007) Multiobjective optimization in bioinformatics and computational biology. IEEE/ACM Trans Comput Biol Bioinform 4(2):279–292

Larranaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, Lozano JA, Armananzas R, Santafe G, Perez A, Robles V (2006) Machine learning in bioinformatics. Brief Bioinform 7(1):86–112

Zhilinskas A, Žilinskas A (2008) Stochastic global optimization. Springer, New York, pp 111–145

Genetic Association

▶ Gene Association and Linkage Analysis

Genetic Control

▶ Regulation

Genetic Epidemiology

▶ Gene Association and Linkage Analysis

Genetic Factor in Parkinson's Disease

Rajeswara Babu Mythri[1], Shireen Vali[2] and M. M. Srinivas Bharath[1]
[1]Department of Neurochemistry, National Institute of Mental Health and Neurosciences (NIMHANS), Bangalore, Karnataka, India
[2]Cell Works Group Inc., Bangalore, India

Definition

It has been documented that only 5–10% of the confirmed cases of PD involve genetic factors. These genetic factors include duplication or point mutations in specific genes. Till date, many genes with both autosomal recessive and dominant inheritances linked to familial PD have been discovered. Such discoveries have been carried based on exhaustive studies on large cohorts of patients including pedigree analysis, genetic mapping, and correlation of disease symptoms. Analysis of these genes and their functions reveal that mutations leading to disruption of their function cause one or more of the following cellular effects: (1) increased oxidative stress, (2) mitochondrial damage, (3) disruption of protein degradation machinery, (4) neurotoxic protein aggregation, and (5) abnormal DA metabolism. Consequently, mutations in two genes might have synergistic effect in dopaminergic neurons. Based on these familial mutations, many cellular and animal models that mimic PD pathology have been developed to understand the molecular mechanisms and to test potential drugs. Interestingly, the molecular effects of PD mutations in dopaminergic neurons are similar to the effects following exposure to environmental toxins. Therefore, PD pathology is an intricate interplay between genetic predisposition and environmental factors that could be summarized as follows: "genetic factors load the gun, environmental factors pull the trigger" (Abeliovich and Beal 2006).

A list of all the reported genes related to PD pathology is shown in Table 1. Among the listed genes, α-syn and leucine-rich repeat kinase (LRRK2) exhibit autosomal dominant pattern and are associated with intracellular protein aggregation. Whereas, Parkin, DJ-1, and Pten-induced kinase1 (PINK1) exhibit autosomal recessive inheritance and are connected with mitochondrial dysfunction and oxidative stress pathways. These gene products have also been shown to interact with each other in different models thereby modulating each other's function.

These gene products have either cytoplasmic or mitochondrial localization. However, they differ in their degradation pathways which could be either via proteasome or autophagy or lysosome. Some proteins possess distinct features and modulate unique functions. For example, α-syn has a role in synaptic function, chaperone activity, microtubule organization and DA metabolism. DJ-1 codes for a protein with chaperone and protease activities with a role in transcriptional regulation. PINK 1 has a protective function against cellular autophagy. Interestingly, toxic mutations in certain proteins might impinge on a variety of cellular functions. For example, α-syn aggregation elicits proteasomal inhibition, mitochondrial damage, oxidative stress and neurotoxicity.

Genetic Factor in Parkinson's Disease, Table 1 List of genes with known familial mutations in PD

Sl. No.	Gene	Park	Locus	Function
1.	α-syn	Park1 and 4	4q21–22	Presynaptic, chaperone
2.	Parkin	Park 2	6q25–27	E3 ubiquitin ligase
3.	UCH-L1 (Ubiquitin carboxy-terminal hydrolase L1)	Park 5	4p14	Ubiquitin recycling enzyme 10
4.	PINK-1	Park 6	1p25–36	Mitochondrial
5.	DJ-1	Park 7	1p36	Oxidative stress response
6.	LRRK2	Park 8	12p11.2q13.1	Kinase

Cross-References

► Disease System, Parkinson's Disease

References

Abeliovich A, Beal MF (2006) Parkinsonism genes: culprits and clues. J Neurochem 99:1062–1072

Genetic Marker

Wentian Li
The Robert S. Boas Center for Genomics and Human Genetics, Feinstein Institute for Medical Research, Manhasset, NY, USA

Synonyms

Genetic variation; Polymorphism

Definition

Genetic markers are detectable variations of DNA sequence with known chromosome locations. A genetic marker may or may not have a biological function. Each possible state among the various number of genetic variants is an allele. The chromosome location of a genetic marker is referred to as a locus. A polymorphism is a genetic variant that appears in at least 1% of the population.

Genetic variations can occur at many different DNA length scales, such as single nucleotide base, oligonucleotides, and long segment of bases. Single nucleotide polymorphism (SNP) is a single base variant, microsatellite markers involve various number of repeats of short DNA sequences (2–6 nucleotides), insertion–deletion variant and inversion variants may cover a range of ~10 bases, and copy number variants occur at between multiple of 10^3 bases (kb) and multiple of 10^6 bases (Mb).

The names of genetic markers are yet to be standardized. However, copy number variants, insertion–deletion variants, inversion variants, and other genomic scale variants are called structural variants.

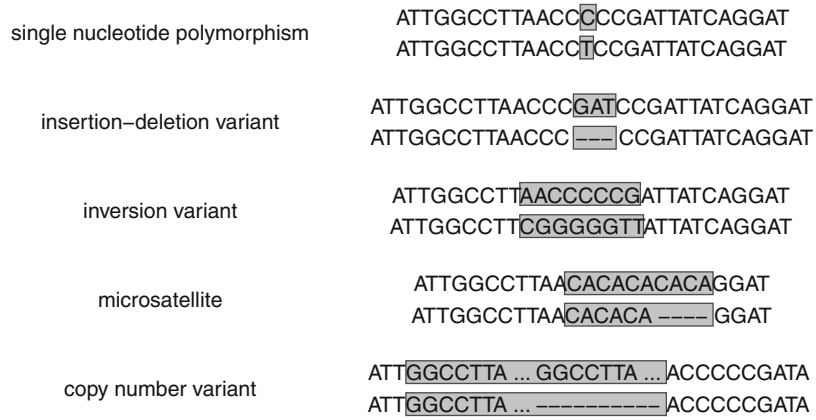
The nature of detection of genetic markers evolves with the development of biotechnology. An early type of genetic marker during 1980s, restriction fragment length polymorphism (RFLP), is less often used now as there are more efficient technologies to detect other types of genetic markers.

Genetic markers are essential for genetic studies (mapping genotype–phenotype relationships), disease gene mapping, and the resulting medical applications (Strachan and Read 2010). It has also been increasingly used in population genetics and evolutionary biology. As genetic markers become surrogates of genes or gene products, they can also be used in studies of gene network and system biology.

Characteristics

Single Nucleotide Polymorphism

Single nucleotide polymorphism (SNP) is a prototypic example of genetic marker (Fig.1). Due to the higher mutation rates between nucleotide adenine (A) and guanine (G), A ↔ G, and between thymine (T) and cytosine (C), T ↔ C, at a given base position on a chromosome, there are typically only two types of nucleotide (e.g., A or G) present. These are the two alleles of the SNP. For diploid species including human, a pair of chromosomes could exhibit at most



Genetic Marker, Fig. 1 Examples of the main DNA variants or genetic markers (modified from (Frazer et al. 2009)). Each DNA sequence segment represents one possible state (allele) Single nucleotide polymorphism (SNP), insertion–deletion variant, and inversion variant typically have only two alleles, whereas microsatellite markers have many possible states depending on the number of simple repeats. Copy number variants (CNVs) involve DNA segments of much longer lengths (represented by dots in the graph), and it is common to have only two alleles

four ordered genotypes (e.g., A|A, A|G, G|A, G|G). When the parental origin of the pair of chromosomes is ignored, there are at most three genotypes for a SNP (e.g., A/A, A/G, G/G).

Genotyping is an experimental determination of the genotype of a genetic marker. The genotyping technology evolves constantly. Many are based on hybridization: single-stranded probe set binds the single-stranded DNA segment containing a specified SNP, and the binding strength differs between the two alleles. The chip technology allows such hybridization-based detection to be carried out for hundreds of thousands of probes, making it possible to genotype a large fraction of SNPs in human genome in one run.

SNP has been a popular choice of genetic marker since mid-1990s. The total number of SNPs in human genome that differentiate unrelated individuals is in the range of several millions (Frazer et al. 2009). The exact number of SNPs depends on a particular ethnic population (e.g., Caucasian, Asian, African), and depends on whether SNPs with very low minor-allele-frequency are counted as polymorphism. An evenly distributed set of 10 million SNPs in human genome corresponds to one SNP per 300 bases on average.

Microsatellite Markers

Microsatellite markers are one type of variable number of tandem repeat markers of which the repeat sequence is simple (1–6 bases) (Fig.1). The common genotyping method for microsatellite markers is to first identify the location of the repeat sequence by its flanking sequence, then amplify the amount of repeats-containing DNA segment by polymerase chain reaction (PCR), and finally the repeat length is determined by a gel electrophoresis. Microsatellite markers played an important role in the construction of genetic map of human genome during 1980s–1990s. The roughly 30,000 known microsatellite markers in human genome cover on average 100 kb per marker.

Copy Number Variants

Copy number variants are also variable number of repeats markers with a much longer repeating sequence (1 kb or larger) (Feuk et al. 2006). The normal number of copies of any DNA segment in a diploid genome is 2, and a deletion (duplication) leads to a reduced (increased) copy number of 1 (3). Several mechanisms for the generation of copy number variants have been proposed, such as nonhomologous recombination.

More than a thousand CNVs have been observed in human genome. The exact number of CNVs should again depend on a particular ethnic population, and depend on whether rare or newly created de novo CNVs are counted. Even though the number of CNVs might be low, more than 90% of base difference between two individual's genomes are due to CNVs and other structural variants (Frazer et al. 2009).

Genetic Marker, Table 1 An illustration of a genetic association test. A total of 1,000 normal and 1,000 disease affected samples are genotyped, and the sample counts for three genotypes (A/A, A/G, G/G) are listed in the table. The row–column correlation in the 2-by-3 genotype table can be tested by Pearson's chi-square test: $\chi^2 = 72.516$ leading to p-value $= 2.22 \times 10^{-16}$. The Cochran–Armitage trend test statistic of the genotype table is CAT = 66.366, corresponding to p-value $= 3.33 \times 10^{-16}$. For the 2-by-2 allele count table, $\chi^2 = 69.829$ leads to p-value $= 1.11 \times 10^{-16}$. Two more quantities can be obtained from the 2-by-2 allele count table: The minor allele frequency difference between the diseased and the normal group is 0.095, and odds-ratio is 2.13

	Genotype count				Allele count		
Phenotype	A/A	A/G	G/G	total	A	G	total
Normal	$n_{0,AA} = 20$	$n_{0,AG} = 170$	$n_{0,GG} = 810$	$n_0 = 1{,}000$	$n_{0,A} = 210$	$n_{0,G} = 1{,}790$	$2n_0 = 2{,}000$
	2%	17%	81%	100%	10.5%	89.5%	100%
Diseased	$n_{1,AA} = 40$	$n_{1,AG} = 320$	$n_{1,GG} = 640$	$n_1 = 1{,}000$	$n_{1,A} = 400$	$n_{1,G} = 1{,}600$	$2n_0 = 2{,}000$
	4%	32%	64%	100%	20%	80%	100%

Association Between Phenotypes and Genetic Markers

The main application of genetic markers is as a surrogate of genes for studying genotype–phenotype relationship based on statistical correlation. One such study is disease gene mapping, where the phenotype is the disease status. Investigation of genotype–phenotype relationship can be carried out on family data (linkage analysis (Ott 1999)), on a randomly selected dataset in a population (association analysis (Balding 2006; Li 2008)), or on an exhaustive collection of people in an isolated population. The latter contains elements from both linkage and association analyses.

For a genetic marker to be surrogate of a gene, an allele of the marker has to be in linkage disequilibrium (LD) with that of a gene in proximity. LD is simply a nonrandom statistical association between alleles at two loci (Slatkin 2008). The presence or absence of LD between neighboring markers partitions the human genome into discrete LD blocks, and all markers in a LD block can potentially be surrogates of genes within the same block.

Genetic association carried out on the whole genome level with a large number of genetic markers is called a genome-wide association study (GWAS) (▶ Genome-wide Association Study). GWAS has generated a long list of genetic markers associated with various human diseases. Statistical analysis plays an important role in marker-based genetic association studies (Balding 2006; Li 2008). Table 1 shows how a typical association test can be carried out for a single SNP.

Gene–Gene and Gene–Environment Interaction

Genetic markers can be used in studying gene–gene and gene–environment interaction, based on the assumption that these can be surrogates of genes at or near the same chromosome location. The term "interaction" used in genetics, statistics, epidemiology, and biochemistry has conflicting meanings, and has been a source of confusion. We may loosely distinguish two types of meanings of interaction.

Bateson's interaction (or epistasis) (Cordell 2002; Phillips 2008), after William Bateson (1861–1926), refers to the situation when the nature of phenotype–genotype relationship for gene 1 is modified by a change of allele in gene 2. This is often discussed when gene 1 is the major effect gene, and gene 2 is a modifier gene.

Fisher's interaction (or statistical interaction), after Ronald Aylmer Fisher (1890–1962), refers to any deviation from the linear phenotype–genotype relation: $y = c_1 g_1 + c_2 g_2$ where y is a quantitative phenotype, g_1 and g_2 are numerical coding of two genes (e.g., number of minor alleles), and c_1, c_2 are regression coefficients. If another model with a nonlinear term, e.g., $y = c_1 g_1 + c_2 g_2 + c_{12} g_1 g_2$ fits the data better than the linear relationship (either in the sense of model selection or in the sense of significant c_{12} coefficient), a statistical interaction between genes 1 and 2 is claimed.

The term "interaction" in biochemistry, biological pathway, and system biology has a more intuitive meaning of a physical contact by chemical bonds, to be in the same pathway, or being linked in a gene network. The purpose to study Fisher's or Bateson's interaction in genetic marker data is to discover the true physical interaction among gene products.

The concept of gene–gene interaction can be extended by gene–environment interaction (Thomas 2010). The Bateson's interaction is particularly easy to be translated in this context: where certain environment factor plays the role of modifier gene.

Cross-References

► Genome-wide Association Study

References

Balding DJ (2006) A tutorial on statistical methods for population association studies. Nat Rev Genet 7:781–791
Cordell HJ (2002) Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. Hum Mol Genet 11:2463–2468
Feuk L, Carson AR, Schere SW (2006) Structural variation in the human genome. Nat Rev Genet 7:85–97
Frazer KA, Murray SS, Schork NJ, Topol EJ (2009) Human genetic variation and its contribution to complex traits. Nat Rev Genet 10:241–251
Li W (2008) Three lectures on case-control genetic association analysis. Brief Bioinfom 9:1–13
Ott J (1999) Analysis of human genetic linkage, 3rd edn. The Johns Hopkins University Press, Baltimore
Phillips PC (2008) Epistasis - the essential role of gene interactions in the structure and evolution of genetic systems. Nat Rev Genet 9:855–867
Slatkin M (2008) Linkage disequilibrium – understanding the evolutionary past and mapping the medical future. Nat Rev Genet 9:477–485
Strachan T, Read A (2010) Human molecular genetics, 4th edn. Garland Science, New York, Part 4
Thomas D (2010) Gene-environment-wide association studies: emerging approaches. Nat Rev Genet 11:259–272

Genetic Material

► Genome

Genetic Network

► Gene Regulatory Networks

Genetic Polymorphisms

Vani Brahmachari and Shruti Jain
Dr. B. R. Ambedkar Center for Biomedical Research, University of Delhi, Delhi, India

Synonyms

Genetic Variations

Definition

The stable coexistence of two or more variations in the genotype in a population at a high frequency is called genetic polymorphism. Based on these variations populations can be divided into subgroups and variability is seen in the gene pool. These polymorphisms lead to multiple alleles for a particular locus, and in disease-associated genes, selective advantage makes one allele protective and the another susceptible to a disease. In enzyme coding genes, genetic polymorphism, may affect the biological activity or affinity of the enzyme to the substrate. The fact that they coexist stably in the population indicates that most often they do not have drastic effects on the phenotype.

Cross-References

► Epigenetics, Drug Discovery

Genetic Redundancy

Diana Ascencio and Alexander DeLuna
Laboratorio Nacional de Genómica para la Biodiversidad, CINVESTAV-IPN, Irapuato, Guanajuato, Mexico

Synonyms

Gene redundancy

Definition

Genetic redundancy is when two or more genes perform the same biochemical function. Genetic redundancy is usually defined at the phenotypic level, that is, to describe situations in which mutations in one of these genes has little or no effect on the organism's fitness. Duplicate (paralogous) genes are the most important source of genetic redundancy. Since duplicated genes may diverge in function, genetic redundancy does not necessarily provide functional redundancy or overlap. The prevalence of genetic redundancy represents a paradox, since redundant functions should be

evolutionarily unstable. Yet, particular scenarios explain why genetic – and even functional – redundancy is frequently generated and maintained in genetic systems.

Characteristics

Prevalence of Genetic Redundancy in Biological Systems

Genetic redundancy is a salient feature of living organisms. Genomes in all three domains of life – bacteria, archaebacteria, and eukaryotes – have a large proportion of duplicated genes, ranging from over 15% and up to 65% of their genes (Zhang 2003). Duplicate genes are constantly originated through small-scale duplication or by whole-genome duplication. The particular mechanism of duplication and biological role are relevant for the process of fixation and conservation of duplicate genes: genes that are part of molecular complexes or cellular networks are more likely retained if duplicated by whole-genome duplication. Likewise, genes with certain functions such as metabolic enzymes, transporters, and transcription factors are often preserved in duplicate (Conant and Wolfe 2008).

Duplicate genes have the potential for yielding novel functions; however, not all gene duplications involve functional innovation. In fact, most gene duplicates do not confer novel functions and are probably preserved in the genome by passive mechanisms such as partitioning of the ancestral function (subfunctionalization) or selection for dosage (Conant and Wolfe 2008). Phylogenetic analyses of genes in model eukaryotes have shown that the functional overlap between duplicate genes is not only a transient state after the duplication, but is often a stable state across long evolutionary scales (Vavouri et al. 2008).

The occurrence of synergistic (negative) genetic interactions is used as a direct indication of functional overlap between duplicate genes. For any two genes, a synergistic interaction occurs when the phenotype of the double-deletion mutant is quantitatively more severe than the expected combined effects of the single deletions. In the particular case of duplicate genes, a synergistic interaction between the duplicates indicates that they can compensate for each other's loss. Up to 55% of duplicate gene pairs in yeast metabolism interact synergistically with each other, which indicate that they carry out important and overlapping functions (DeLuna et al. 2008). Moreover, duplicate genes show on average a lower number of genetic interactions with other genes, which also reflects functional compensation and overlap between the duplicates (VanderSluis et al. 2010). Taken together, these observations suggest that duplicate genes tend to maintain a substantial functional overlap.

The Problem of Genetic Redundancy

It is challenging to account for genes with redundant functions that have been conserved for long evolutionary times. Natural selection, in particular purifying selection, acts to conserve genes that impact fitness of an organism, thus a fully redundant function should in principle not be under any kind of selective pressure. This rationale makes genetic redundancy evolutionarily unstable, which apparently contradicts the prevalence of genetic redundancy across biological species (Nowak et al. 1997).

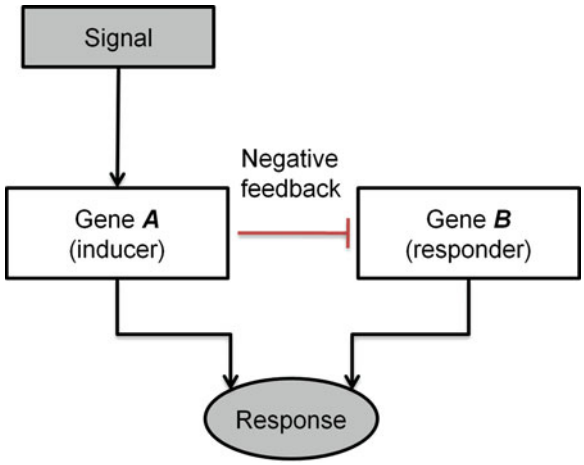
Nowak et al. (1997) modeled several scenarios that could explain why genetic redundancy is common and even evolutionarily stable. Let us consider a population of organisms that have two loci A and B that perform one essential function F. If both genes perform the function with similar efficacy and mutation rates operate equally on both genes, genetic redundancy is evolutionarily unstable. In a different scenario, one of the genes, B, performs the function with slightly less efficacy than its paralog A; redundancy is maintained because B has lower mutation rate than A. Nowak et al. (1997) also considered situations where genes perform more than one function, the overlap between both genes affecting only one function, as well as instances where developmental errors provide a selective pressure to maintain genetic redundancy. In addition, both genes performing exactly the same function needed in high amounts would result in deletions of any of the two genes causing fitness defects; hence both genes would be under selective pressure (Conant and Wolfe 2008).

Genetic Robustness, Genetic Redundancy, and Functional Innovations

Living organisms are remarkably robust to genetic perturbations. Genome-wide surveys in model organisms have shown that complete gene inactivation typically has little or no phenotypic effect. There are two main mechanisms that are responsible for such functional compensation to mutations (genetic robustness). Genetic

redundancy is one of these mechanisms, whereby deletion of one gene has little effect due to the presence of its duplicate and functionally overlapping paralogous gene. The second mechanism of genetic robustness relies on the distributed nature of genetic networks; interactions between genes with unrelated functions also provide functional compensation (Wagner 2008).

Early after gene duplication the sister copies may experience a relaxed selection pressure; sequences of duplicate genes tolerate more nucleotide changes than their single-copy ancestral genes. Hence, genetic redundancy provides not only functional backup and robustness, but also the chance to accumulate sequence changes that may diversify their functions. This innovation mechanism has evolutionary potential. In fact, there are several examples of the important role of gene duplication in vertebrate radiation, flowering plant evolution, and heart development (Wagner 2008). There is thus an important link between genetic redundancy, mutational robustness, and functional innovations, as their constant interplay allows organisms to survive environmental perturbations and yet have the ability to generate phenotypic diversity.



Genetic Redundancy, Fig. 1 The utilization of genetic redundancy in a responsive backup circuit. Two duplicate genes, *A* and *B*, perform the same function, but they are regulated in a different ways. Levels of protein *B* are negatively regulated by its duplicate *A* that is positively regulated by an external signal. The response is the sum of the two products. Even though the signal upregulates *A*, this will prompt downregulation of *B*, reducing fluctuations in overall protein and activity levels (Adapted from Kafri et al. (2009))

Dosage Effects and Increase of Metabolic Flux

Gene redundancy may increase gene expression level. The process of gene duplication has an immediate effect on gene dosage and this feature can be beneficial for the organism, which in turn drives the conservation of functionally overlapping pairs of genes. Redundant genes encoding ribosomal proteins and some metabolic isoenzymes in yeast have likely been retained in duplicate due to selection for changes in gene dosage (Conant and Wolfe 2008).

In yeast, about 25% of duplicate genes originated from the whole-genome duplication are metabolic enzymes. Analyses of metabolic networks and flux-models indicate that genetic redundancy is not more frequently associated with reactions with higher flux and not to indispensable metabolic reactions, suggesting that the reason for their retention is to increase metabolic flux rather than providing functional compensation (Papp et al. 2004). Most genes in the glycolytic and fermentation pathways of *Saccharomyces cerevisiae* are present in more than one copy. Following whole-genome duplication, an increase in the products of these genes gave yeast a growth advantage during adaptation to glucose-rich environments; the appearance of angiosperms on earth opened an ecological niche for microorganisms with the ability to consume glucose and produce ethanol rapidly through fermentation (Conant and Wolfe 2008). Genetic redundancy may thus provide a beneficial dosage-mediated effect underlying adaptation of an organism to a novel environment.

Responsive Backup Circuits

Genetic redundancy could provide a mechanism to reduce noise caused by the direct interaction between a signal and the response. From a series of studies in vertebrate developmental pathways, it has been noticed that redundant duplicates are typically cross-regulated by negative feedback, allowing one of the redundant proteins, the responder, to respond to an alteration in the expression of its partner, the inducer Fig. 1. This mutual repression can be used to reduce stochastic fluctuations in protein expression and to reduce the effects of noise in environmental or developmental signals (Kafri et al. 2009). For example, MyoD and Myf5 are two functionally overlapping regulators of skeletal muscle

development, which are expressed in separate cell lineages. Mutations in MyoD induce increased proliferation of the Myf5-positive cell lineage thus increasing expression of the Myf5 isoform. In this case, extracellular signals regulate a "responsive circuitry" comprising MyoD and Myf5 that effectively buffers against fluctuations in MyoD levels (Kafri et al. 2009).

In yeast, a modest fraction of proteins show significant upregulation upon deletion of their duplicate genes. Metabolic enzymes are over-represented among such paralog-responsive proteins that match almost exclusively duplicate pairs whose overlapping function is required for growth. Moreover, media conditions that add or remove requirements for the function of a duplicate gene pair specifically eliminate or create responsiveness of duplicate genes (DeLuna et al. 2010). These observations suggest that responsiveness between duplicate genes could indeed provide an important mechanism for compensation of genetic, environmental, or stochastic perturbations in protein abundance.

References

Conant GC, Wolfe KH (2008) Turning a hobby into a job: how duplicated genes find new functions. Nat Rev Genet 9:938–950

DeLuna A, Vetsigian K, Shoresh N, Hegreness M, Colón-González M, Chao S, Kishony R (2008) Exposing the fitness contribution of duplicated genes. Nat Genet 40:676–681

DeLuna A, Springer M, Kirschner MW, Kishony R (2010) Need-based up-regulation of protein levels in response to deletion of their duplicate genes. PLoS Biol 8: e1000347

Kafri R, Springer M, Pilpel Y (2009) Genetic redundancy: new tricks for old genes. Cell 136:389–392

Nowak MA, Boerlijst MC, Cooke J, Smith JM (1997) Evolution of genetic redundancy. Nature 388:167–171

Papp B, Pal C, Hurst LD (2004) Metabolic network analysis of the causes and evolution of enzyme dispensability in yeast. Nature 429:661–664

VanderSluis B, Bellay J, Musso G, Costanzo M, Papp B, Vizeacoumar FJ, Baryshnikova A, Andrews B, Boone C, Myers CL (2010) Genetic interactions reveal the evolutionary trajectories of duplicate genes. Mol Syst Biol 6:429

Vavouri T, Semple J, Lehner B (2008) Widespread conservation of genetic redundancy during a billion years of eukaryotic evolution. Trends Genet 24:485–488

Wagner A (2008) Gene duplications, robustness and evolutionary innovations. Bioessays 30:367–373

Zhang J (2003) Evolution by gene duplication: an update. Trends Genet 18:292–298

Genetic Regulation Mechanisms

Pramod R. Somvanshi and Kareenhalli V. Venkatesh
Department of Chemical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai, Maharashtra, India

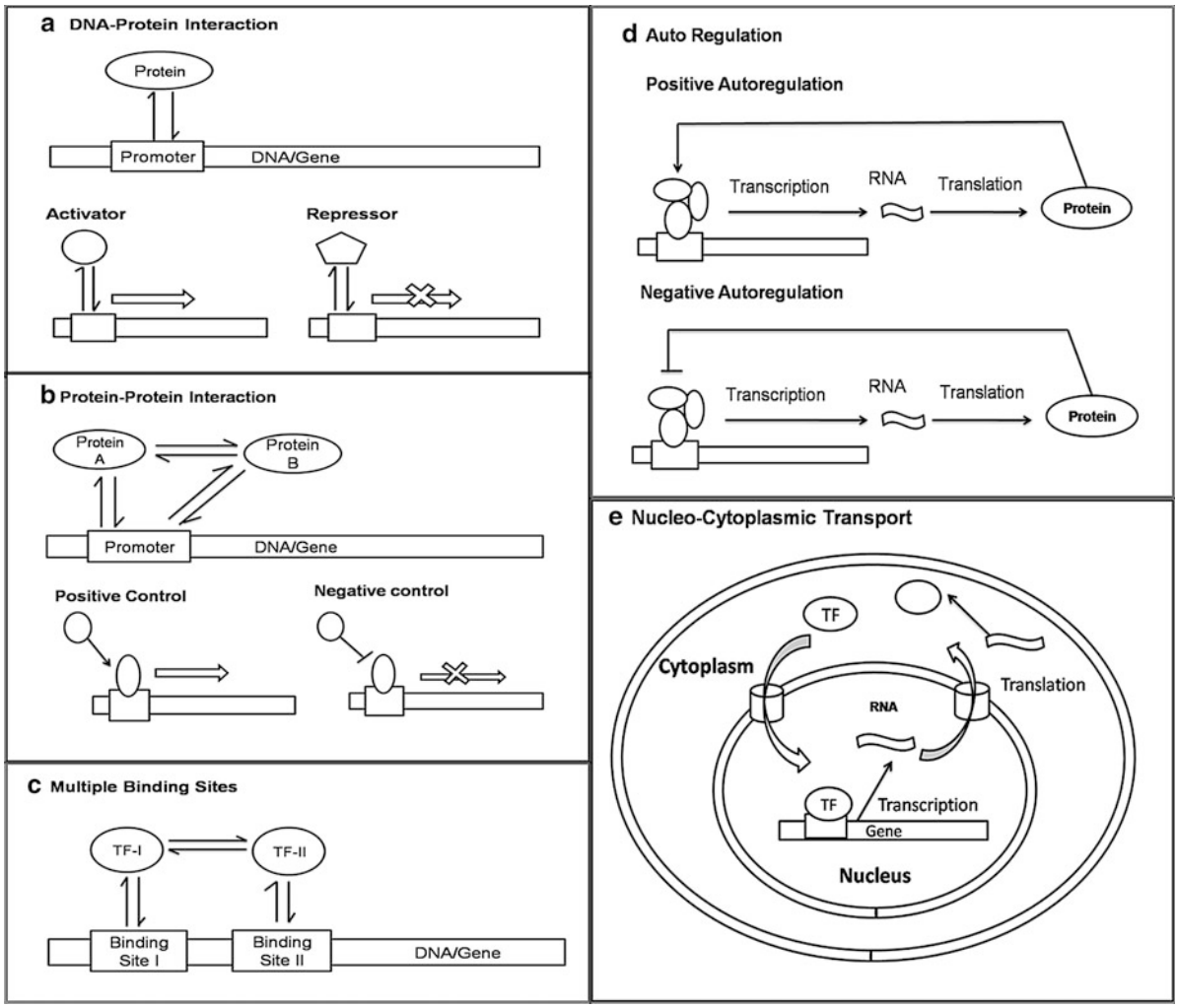
Definition

Genetic information is carried in the form of genetic code in the DNA of an organism and corresponding genes need to be expressed to yield a specific response or phenotype. Gene expression is a multistep process which is precisely coordinated and controlled by various complex regulatory mechanisms for an optimal performance and is termed as genetic regulation (Perdew et al. 2006).

Characteristics

The gene expression is a process by which the genetic information is transferred from DNA to RNA and RNA to proteins via ▶ transcription and translation, respectively. The expression is triggered by the preceding signaling cascades which activate the ▶ transcription factors for the initiation of the transcription. RNA polymerases, along with other protein complexes, facilitate transcription process. The gene products of the transcription are RNA molecules, which after further processing and modification (capping, splicing, termination, cleavage, and polyadenylation) are transported from nucleus to cytoplasm for translation into functional proteins.

Prokaryotic and eukaryotic gene expression have several regulatory mechanisms in common; however, they differ in their cellular structures, wherein prokaryotic cells lack the nuclear envelope and the chromatin structures. Hence, the events of nucleocytoplasmic transport and chromatin remodeling are very specific to eukaryotic systems In prokaryotes, genes are clustered together to form an operon which encodes the proteins required for coordinated functioning and follow certain order of expression. Further, the multiple operons coordinate together to form a regulon, which is typically regulated through common regulatory mechanisms. Gene expression is regulated at various stages from transcription of RNA to the post-translational



Genetic Regulation Mechanisms, Fig. 1 Gene regulatory mechanisms

modification of the proteins. Usually, the regulation is achieved by altering the rates of various steps mentioned above. The various mechanisms that elicit such kind of regulations are briefed below (see Fig. 1) (Perdew et al. 2006; White and Sharrocks 2010).

Protein-DNA Interactions

Protein-DNA interactions are the most fundamental interactions in regulation of the gene expression where various regulatory proteins bind to the DNA (Fig. 1a) to accomplish transcription. A gene contains a core promoter region and contains sequence called as TATA box where the proteins complexes including different transcription factors (TFIID,-B,-E,-F,-H), TBPs (TATA-binding proteins), and TAFs (TBP-associated factors) can bind to facilitate transcription. The process of gene expression is initiated with the activation and binding of ▶ transcription factors (TF) to the promoter regions. TFs also help in activating and recruiting RNA polymerases (enzymes involved in RNA synthesis) to the transcriptional initiation site. A single TF can act as an activator or repressor for two different genes or multiple genes at a time. There are certain enhancer and silencer regions on the DNA, which when bound by the respective transcription factors drastically increase and decrease the rate of transcription, respectively. Moreover, binding of certain activator or repressor molecules to the promoter regions increases or decreases the rate of transcription, respectively. Similarly in prokaryotes, certain operons are catabolite-regulated operons (e.g., *lac*-operon) where catabolite acts as an activator for energy metabolism whereas for attenuated operons

(e.g., *trp*-operon) the proteins act as a repressor for its own synthesis (Orphanides and Reinberg 2002; Levine and Tjian 2003; Kornberg 1999).

Protein-Protein Interactions

Various protein molecules and protein complexes are known to interact with the TFs and RNA polymerases to modulate the regulation of the gene expression (Fig. 1b). The rate of gene expression can be regulated by modulating either the probability of the TF binding or the strength (affinity) of the TF binding to the promoter region. It is found that the increase in TF binding strength is achieved by the formation of dimeric or multimeric protein-DNA complex. Multimeric protein-DNA interactions have been shown to yield steep sensitive responses through the effect of stoichiometry as compared to a binding of single TF. Several proteins act as specificity factors, inducers, activators, repressors, corepressors, coactivators, and mediator complexes to regulate gene expression. Specificity factors are the proteins that alter (increase or decrease) the specificity of the RNA polymerases to the promoter regions. Inducers are the molecules that interact with the activators and repressors to induce the transcription through positive and negative controls, respectively. Activators enhance the rate of transcription by increasing the affinity of the RNA polymerase to the respective promoter regions. Repressors intervene the association of the RNA polymerases and TF binding to the promoter regions and decrease the rate of transcription. The mediator protein complex is an important interface that compounds the activity of RNA polymerases with the activator protein complex to facilitate the transcriptional regulation. Moreover, there are various interactions of coactivators and corepressors with activators and repressors leading to positive and negative control mechanisms (Perdew et al. 2006; Orphanides and Reinberg 2002; Levine and Tjian 2003; Kornberg 1999).

Protein Modifications and Stability

The cell has to respond to changing environmental stimuli and correspondingly regulates the net protein abundance at a given instance. This is achieved by controlling the rates of synthesis and degradation of the proteins and altering the stability of the mRNAs and proteins as the requirement. Various cofactors assign the functional groups to the amino acid residues and help in catalyzing the enzymatic reactions for protein stabilization and degradation. Several modifications in TFs and proteins are ubiquitination, phosphorylation, acetylation on lysine residues, methylation on arginine and lysine residues, glycosylation, fatty acylation, disulfide bond formation, and proteolysis (Orphanides and Reinberg 2002).

Multiple Binding Sites and Cooperativity

Certain promoter regions possess multiple binding sites for the TFs which can modulate the genetic responses (Fig. 1c). It is observed that higher promoter occupancy and higher strength of TF binding leads to increased rate of transcription and vice versa. The availability of the multiple binding sites provides a room for altering the rates of transcription based upon the combinatorial effect of the upcoming signals. The bound subunit has the cooperative effect on the binding of the next subunit by increasing or decreasing its affinity toward the binding region exhibiting positive or negative cooperativity, respectively. The stoichiometry of interaction is drastically affected and the equilibrium is so shifted that a steep rise in the rate of expression is obtained with increasing number of binding sites. Occupancy of multiple binding sites has shown to yield ultrasensitive and subsensitive responses by the binding of activators and repressors, respectively. Such responses also facilitate initial delays and are operational only by certain activation or repression thresholds (Chin et al. 1999; Sacketta and Saroff 1996).

Auto-Regulation and Feedback Mechanisms

Auto-regulation is a phenomenon in which the rate of gene expression is directly or indirectly regulated by its own gene product with certain feedback mechanisms (Fig. 1d). The process by which a gene product upregulates its own production by increasing the rate of its gene expression as a result of positive feedback is called as positive auto-regulation (PAR). PAR exhibits slower response time due to the delay in activation and is sensitive to the inherent noise leading to cell-to-cell variability in the gene product concentrations. It helps in signal amplification and pattern generation. The process by which the gene product downregulates its own production by the decreasing rate of its gene expression as a result of negative feedback is called as negative auto-regulation (NAR). NAR fastens the response time and is resistant to the inherent noise leading to reduced cell-to-cell variability of gene product concentrations. Various combinations of positive and negative feedback loops have shown to elicit memory, bistability,

oscillations, and robustness in the gene expression profiles. Moreover, such feedback mechanisms allow the cells to filter out transient input signals and aids in appropriate decision making by responding to the multiple regulatory pathways (Alon 2007).

Nucleocytoplasmic Transport

In eukaryotes, a potential way of regulating gene expression is by controlling nucleocytoplasmic transport (Fig.1e) through the nuclear pore complexes and the selective exchange of RNA (export) and protein molecules (import) in the two compartments. The nuclear and cytoplasmic environment is spatially separated by the nuclear envelope, which facilitates the separation of transcriptional and translational machinery. The transport of the RNA and proteins are mediated by the specialized transport motifs called as nuclear export signals (NES) and nuclear localization signals (NLS), respectively. The shuttling of the heterodimeric complex (to and fro) from nucleus to cytoplasm serves as a quality control mechanism by selective transport of the only mature RNAs to the translational machinery. This transport can also be regulated by the covalent modifications of the NES and NLS molecules. Furthermore, the event of RNA export is coupled with the transcription and pre-mRNA processing by the capping and splicing mechanisms. Many proteins shuttle continuously between cytoplasm and nuclease based on their requirement and localization at the respective sites (Orphanides and Reinberg 2002; Dimaano and Ullman 2004).

Chromatin Remodeling and Histone Modification

In eukaryotes, DNA is not easily accessible to the transcription interactions as it is supercoiled around the core of octameric histone proteins in the nucleosomes leading to a highly compact structure called as chromatin. Various coregulatory proteins are recruited along with the transcriptional factor to facilitate chromatin decompaction and conformational changes that provide access to the promoter regions for recruitment of RNAPII and general transcriptional machinery. This process is coupled with the transcription elongation. The chromatin structure can be temporarily modified by the phosphorylases from signaling cascades and permanently modified by methylation of DNA, termed as gene silencing. For the activaton of the genes, activator proteins recruit HATs (Histone acetyltransferse) and HMTs (Histone methyltransferase) to the promoter regions of the genes, which leads to acetylation and methylation of N-terminal residues of histone tails. On the contrary, the transcriptional repressors recruit HDACs (Histone deactylases) that leads to deacetylation of the histone tails and subsequent repression of the transcription (Orphanides and Reinberg 2002; Kornberg 1999).

The above discussed mechanisms work in coordination with different levels to regulate gene expression. Cells have engineered gene regulatory motifs involving these mechanisms in combination with feed forward and feedback loops to elicit a system level regulation. Moreover, the global regulatory pathways from signaling and metabolic networks also coordinate to govern the gene expression profiles. With advances in research in this area, newer interactions and mechanisms are being explored that elicit precise regulation.

Cross-References

- ▶ Transcription
- ▶ Transcription Factor

References

Alon U (2007) Network motifs: theory and experimental approaches. Nat Rev Genet 8:450–461

Chin JW, Kohler JJ, Schneider TL, Schepartz A (1999) Gene regulation: protein escorts to the transcription ball. Curr Biol 9:R929–R932

Dimaano C, Ullman KS (2004) Nucleocytoplasmic transport: integrating mRNA production and turnover with export through the nuclear pore. Mol Cell Biol 24(8):3069–3076

Kornberg RD (1999) Eukaryotic transcriptional control, multilayered transcription mechanisms, millenium issue. TCB 9(12):(0962–8924)

Levine M, Tjian R (2003) Transcription regulation and animal diversity. Nature 424:147–151

Orphanides G, Reinberg D (2002) A unified theory of gene expression. Cell 108:439–451

Perdew GH, Vanden Heuvel JP, Peters JM (2006) Regulation of gene expression-molecular mechanisms. Human Press, Totowa, Book

Sacketta DL, Saroff HA (1996) The multiple origins of cooperativity in binding to multi-site lattices. FEBS Lett 397:1–6

White RJ, Sharrocks AD (2010) Coordinated control of the gene expression machinery. Trends Genet 26(5):214–220

Genetic Variation

- ▶ Genetic Marker

Genetic Variations

▶ Genetic Polymorphisms

Genome

Vani Brahmachari and Shruti Jain
Dr. B. R. Ambedkar Center for Biomedical Research, University of Delhi, Delhi, India

Synonyms

Genetic material

Definition

The genome is defined as the entire genetic makeup of an organism. It can be either DNA or RNA (as is in the case of some viruses). It includes the nuclear as well as organelle DNA. The complexity and the size of the genome vary between organisms. The average human genome size is 3.2×10^9 base pairs and corresponds to one copy of each of the 23 chromosomes (haploid set) as opposed to 4.6×10^6 base pairs for *Escherichia coli*.

Cross-References

▶ Epigenetics

Genome Annotation

Akos Dobay
Institute of Evolutionary Biology and Environmental Studies (IEU), University of Zurich, Zurich, Switzerland

Definition

Genome annotation is the process of attaching higher-level information to primary sequences. The whole process consists of starting with raw DNA sequences and giving a biological meaning to its content (Stein 2001). The first step in annotating a raw sequence will require the mapping of structural elements in the genome by comparing the latter against a library of already known sequences. This especially includes genetic markers, genetic polymorphisms, protein-coding regions (▶ Protein Structure Comparison, High-Performance Computing, ▶ Proteome, ▶ Proteome Analysis Pipeline, ▶ Proteomics), as well as other functional elements such as non-translated RNAs (▶ RNA-seq), repetitive elements, duplicated genes, and regulatory regions (▶ Gene Regulation, ▶ Regulation, ▶ Regulation and Autoregulation, ▶ Regulation Function). The structural elements are aligned using a specialized search tool, such as BLAST (http://blast.ncbi.nlm.nih.gov/Blast.cgi) developed at the National Center for Biotechnology Information Database (Altschul et al. 1990), or GENSCAN (http://genes.mit.edu/GENSCAN.html) by Burge and Karlin (1997). The second step in annotation involves the association of biological processes to the sequences identified from the first step. These processes include metabolic functions, regulation (▶ Gene Regulation), interactions (▶ Binding Affinity), and gene expression (▶ Gene Expression). In 1999, a consortium created the gene ontology (http://www.geneontology.org) to standardize the vocabulary used for describing genes, gene products, and genomes.

References

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215:403–410
Burge C, Karlin S (1997) Prediction of complete gene structures in human genomic DNA. J Mol Biol 268:78–94
Stein L (2001) Genome annotation: from sequence to biology. Nat Rev Genet 2:493–503

Genome Browsers

▶ Genomic Resources

Genome Project

▶ NCBI BioProject Genome Resources

Genome-Scale Metabolic Modeling

► Constraint-based Modeling

Genome-Scale Metabolic Network

Andrzej M. Kierzek
Division of Microbial Sciences, Faculty of Health and Medical Sciences, University of Surrey, Guildford, Surrey, UK

Definition

Genome-scale metabolic network is the list of biochemical reaction formulas implied by the repertoire of enzymes identified in the genome of the organism under investigation.

Cross-References

► Mycobacterium Tuberculosis

Genome-Scale Metabolic Network Inference

Oliver Ebenhöh[1] and Stefan Kempa[2]
[1]Institute for Complex Systems and Mathematical Biology, University of Aberdeen, Kings College, Old Aberdeen, Aberdeen, UK
[2]Integrative Metabolomics and Proteomics, BIMSB (Berlin Institute for Medical Systems Biology), Berlin, Germany

Definition

The metabolic network is the biochemical backbone for all chemical reactions in a cell, organ, or organism. Metabolism is a basic sign of life and metabolic dysfunctions may cause or be the result of many diseases. The structure of the network and the identity of the proteins (here enzymes) determine the metabolic behavior of the system. With the availability of genome-scale information and a huge knowledge about identity and specificity of enzymes, first efforts have been undertaken to reconstruct the metabolic network from genome data.

Therefore data from several biological layers, e.g., genome, transcriptome, proteome, and metabolome, can be used to infer the specific metabolic network.

The genome contains the information which proteins a cell can in principle produce. Therefore, a metabolic network can in principle be derived from the genome by assembling all enzymes for which a coding region in the genome has been identified. However, several issues make this task nontrivial. Firstly, to identify a function of a particular gene relies on finding similarities, or homologies, to known and previously annotated genes. This is essentially a statistic process and therefore there is no certainty whether new annotations are correct. But even if this problem could be resolved, the next difficulty arises in defining the abstract, computer-readable description of the metabolic network. Which reactions should be included in the model and where should a "boundary" of metabolism be defined? Clearly, macromolecular assembly, such as protein or RNA synthesis, cannot be included, but it is by no means trivial to define a suitable threshold for the complexity of the included metabolites. For example, small proteins such as ferredoxins or thioredoxins are important cofactors and as such a description of a metabolic network without these compounds is somewhat incomplete. Finally, a network defined from the genome sequence represents the maximal metabolic capabilities of a cell or organism; however, which enzymes are actually expressed and active depends greatly on the environmental conditions or the specific tissue into which a cell has been differentiated.

The following general strategy allows exploiting different types of high-throughput data with heterogeneous origin:

The reconstruction process typically begins with the sequenced and annotated genome. With the help of biochemical databases such as KEGG, BRENDA, Reactome, or MetaCyc, the annotated genes are mapped to enzymes and the catalyzed biochemical reactions. Depending on the envisaged computational analysis, this established list of reactions has to be curated. Most importantly, stoichiometric inconsistencies have to be detected and removed. Reactions in which the numbers of atoms in the substrates and products are not balanced are fatal for any further computational analysis because the models may predict that metabolites are created from nothing or completely annihilated. More difficult to

detect are thermodynamic inconsistencies which may lead to energy-producing cycles which clearly violate fundamental laws of thermodynamics. This curation process leads to a draft metabolic network which can be used for further computational analyses including constraint-based models, such as flux balance analysis (FBA).

The second task is typically the identification of missing reactions and subsequent completion of the draft network. The principle strategy is to query the model and validate whether it agrees with experimental observations. The most fundamental observation is that a cell is actually alive and growing, which implies that the metabolic network must be able to produce all biomass precursors from the available nutrients. Knowledge of the chemical composition of the growth medium and the biomass allows the definition of fluxes which the model must be able to support. Metabolome data are highly useful to refine this analysis. Model consistency demands that, besides biomass precursors, all other experimentally observed metabolites must also be producible by the network. After those experimentally observed metabolic functions have been identified which the draft network is not able to support, candidate reactions must be found which, when added to the draft, provide the network with the missing functionality. Several methods for this gap-filling have been proposed. One approach which allows integrating genomic, proteomic, and metabolomic data is described by Christian et al. (2009). Essentially, the draft network is embedded in a reference network derived from a biochemical database containing enzymatic reactions from a large number of different organisms. With a simple search algorithm, minimal sets of reactions are identified which complete the draft to consistency. Subsequently, the identified reactions are ranked according to how well the genome sequence and proteomic data support the existence of a catalyzing enzyme. In this way, the genome annotation of *Chlamydomonas reinhardtii* could be considerably improved. The caveat of this and similar approaches is that only those reactions can be identified which have been previously reported. A future challenge is to develop computational methods which allow for the identification of hitherto unknown metabolic reactions. In principle, such a method could be based on chemical reaction patterns from which theoretically feasible reactions can be computed that may be catalyzed either by known enzymes or by putative enzymes with functional similarities to known enzymes.

In multicellular organisms the different cell types fulfill specific metabolic functions and therefore each cell displays a specific expression pattern for metabolic enzymes. Also single cellular organisms adapt their molecular repertoire according to external or internal stimulation. Thus, not all genome encoded reactions are expressed all the time in all cells. Using further information from transcriptome analyses or protein expression data, active metabolic subnetworks can be inferred. A strategy which is exemplified on different human tissues has been described by Jerby et al. (2010).

A still unresolved challenge is to combine genome-scale metabolic models with global gene regulatory models. The expressed subnetwork can be viewed as a result of the genetic up- or downregulation of genes encoding the respective enzymes. However, also the metabolic state of the cell is signaling back to the transcriptional regulatory system. An approach how the metabolic and gene regulatory subsystems can be simultaneously described by timescale separation has been proposed by Baldazzi et al. (2010). However, this method has so far only been demonstrated for very small systems and its application to large-scale models seems difficult. To validate these kinds of combined models, however they may be realized in detail, it will be necessary to simultaneously monitor the dynamics of concentrations of metabolites (small molecules), the level of transcripts, and the activity of enzymes.

References

Baldazzi V, Ropers D, Markowicz Y, Kahn D, Geiselmann J, de Jong H (2010) The carbon assimilation network in *Escherichia coli* is densely connected and largely sign-determined by directions of metabolic fluxes. PLoS Comput Biol 6:e1000812

Christian N, May P, Kempa S, Handorf T, Ebenhoh O (2009) An integrative approach towards completing genome-scale metabolic networks. Mol Biosyst 5:1889–1903

Jerby L, Shlomi T, Ruppin E (2010) Computational reconstruction of tissue-specific metabolic models: application to human liver metabolism. Mol Syst Biol 6:401

Genome-Wide Association

▶ Gene Association and Linkage Analysis

Genome-Wide Association Study

Wentian Li
The Robert S. Boas Center for Genomics and Human Genetics, Feinstein Institute for Medical Research, Manhasset, NY, USA

Synonyms

Genome-wide case-control studies; Genome-wide genetic association analysis

Definition

Genome-wide association studies (GWAS) are projects to investigate the statistical association between phenotypes and a dense set of genetic markers (► Genetic Marker) that capture a substantial amount of genetic variations in the genome, using a large number of matched samples.

Phenotypes can be qualitative traits such as disease status or quantitative traits such as blood pressure. Statistical association between disease status and alleles of a genetic marker is carried out by categorical data analysis.

Genetic markers are usually genotyped by microarray chips. Whether a substantial genetic variation in the genome, including common, rare, and structural variations, is captured by the set of markers depends on the number of markers and their chromosome locations.

The typical number of single nucleotide polymorphism (SNP) markers used in a current GWAS is 300 k (300,000), 500 k (500,000), or 1 M (1,000,000). The number of samples collected in a GWAS range from hundreds to tens of thousands. These large number of markers present a particular problem for statistical analysis called multiple testing.

Stringent data quality control procedures are usually required before statistical analysis, as biased genotyping rates between case and control groups could lead to false positives.

The diseased (case) and normal (control) samples in a GWAS have to be appropriately matched by their ethnic or geographic origin, to avoid true signal being overwhelmed by genetic variations unrelated to the disease. Using GWAS genotyping data, it is possible to uncover unmatched samples and outliers by statistical methods such as cluster analysis. Depending on the study, matching can also be done by other criteria, such as gender and environmental exposure. Lack of proper sample matching is a common cause of false positive results.

References

Hirschhorn JN, Daly MJ (2005) Genome-wide association studies for common diseases and complex traits. Nat Rev Genet 6:95–108

Ioannidis JPA, Thomas G, Daly MJ (2009) Validating, augmenting and refining genome-wide association signals. Nat Rev Genet 10:318–329

McCarthy MI, Abecasis GR, Cardon LR, Goldstein DB, Little J, Ioannidis JPA, Hirschhorn JN (2008) Genome-wide association studies for complex traits: consensus, uncertainty and challenges. Nat Rev Genet 9:356–369

Genome-Wide Case-Control Studies

► Genome-Wide Association Study

Genome-Wide Genetic Association Analysis

► Genome-Wide Association Study

Genomic

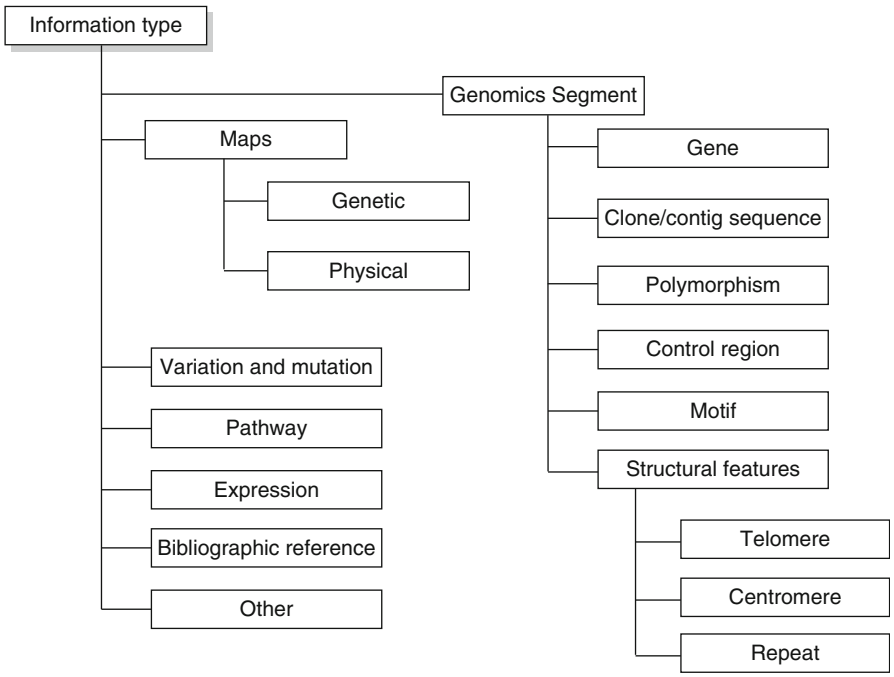
► Microbiome

Genomic Databases

Erika De Francesco[2], Giuliana Di Santo[1], Luigi Palopoli[1] and Simona E. Rombo[1,3]
[1]Dipartimento di Elettronica, Informatica e Sistemistica Università della Calabria, Rende, Italy
[2]Exeura, Rende (CS), Italy
[3]Istituto di Calcolo e Reti ad Alte Prestazioni, Consiglio Nazionale delle Ricerche, Rende (CS), Italy

Synonyms

Non-coding intergenic sequences; Genomic Datasets



Genomic Databases, Fig. 1 Information typologies

Definition

Genomic databases store datasets related to the genomic sequencing of different organisms and gene annotations. Differently from gene databases, containing only coding DNA sequences, genomic databases contain also non-coding intergenic sequences. Genomic databases are listed among the data resources useful in systems biology.

FASTA Format

Each entry of a FASTA document consists of three components: (1) a comment line that is optional and reports brief information about the sequence and the GenBank entry code; (2) a sequence that is represented as a string on the alphabet *{A, C, G, T}* of the nucleotide symbols; and (3) a character denoting the end of the sequence.

Characteristics

Systems biology focuses on studying (complex) interactions involving various components (among others, genes, proteins, enzymes, pathways, and such) in biological systems. Therefore, studies in systems biology often rely on (sometimes massive) bunches of data that are stored nowadays in biological databases.

The number of relevant biological data sources available to date can be estimated in about 970 units (Galperin 2007). Among them, there are about 100 *genomic database*s that can be classified by considering different characteristics.

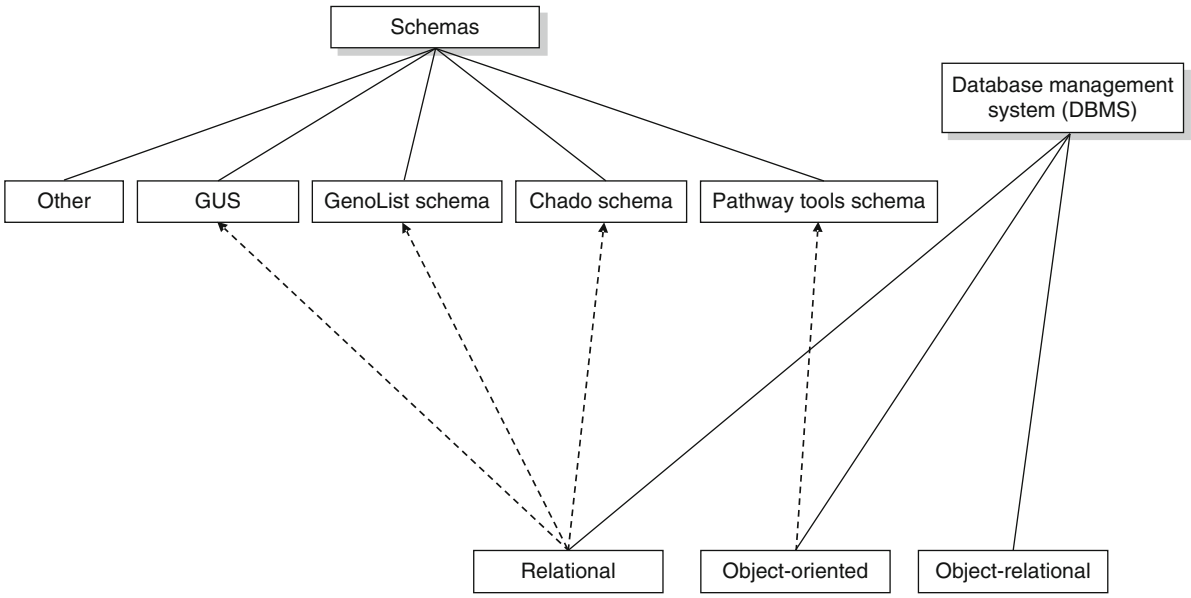
One way to go is by employing five different and orthogonal characteristics (De Francesco et al. 2009), that are:

Typologies of recoverable information (e.g., genomic segments or clone/contig regions)
Database schema types (e.g., Genolist schema or Chado schema)
Query types (e.g., simple queries or batch queries)
Search methods (e.g., graphical interaction–based or query language–based methods)
Result formats (e.g., flat files or XML formats)

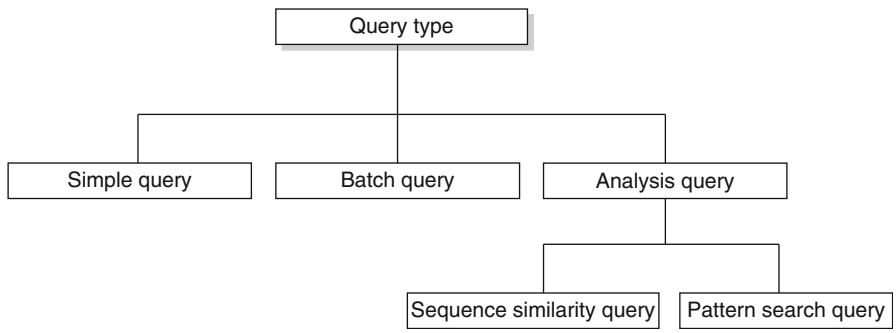
These characteristics are more thoroughly discussed next.

Recoverable Information

Genomic databases contain a large set of data types. Some archives report only the sequence, the function, and the organism corresponding to a given portion of genome, and other ones contain also detailed information useful for biological or clinical analysis. Figure 1 shows a taxonomy of the main classes of information



Genomic Databases, Fig. 2 Database schemas



Genomic Databases, Fig. 3 Query types

that are recoverable from genomic databases. For example, *Genomic segments* include all the nucleotide subsequences that are meaningful from a biological point of view, such as *genes*, *clone*/*clontig sequences*, *polymorphisms*, *control regions*, *motifs*, and *structural features* of chromosomes.

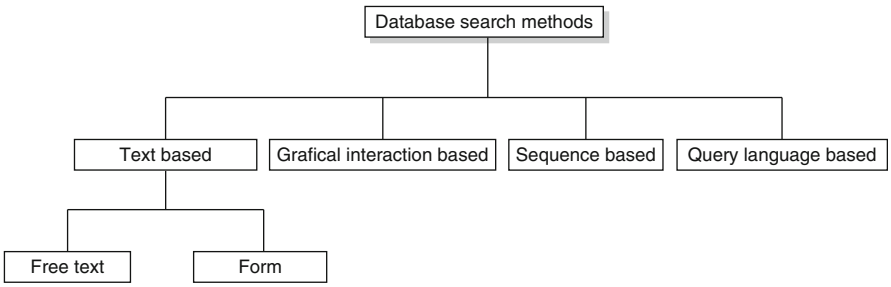
Database Schemas

Most genomic databases are relational, even if relevant examples exist based on the object-oriented or the object-relational model (see e.g., (WormBase 2006; Twigger et al. 2002)). Four different types of database schema that are designed "specifically" to manage biological data can be distinguished from other unspecific database schemas, which are mostly generic relational schemas, whose structure is anyways independent from the biological nature of data (see Fig. 2). For example, the *Genomics Unified Schema (GUS)* (GUS 2006) is a relational schema suitable for a large set of biological information, including genomic data, genic expression data, and protein data, while the *Pathway Tools Schema* (Karp 2000) is an object schema used in Pathway/Genome databases.

Query Types

In Fig. 3, a taxonomy of query types supported by most genomic databases is illustrated.

By *simple querying*, it is possible, for example, to recover data satisfying some standard search parameters such as gene names, functional categories, and others.



Genomic Databases, Fig. 4 Search methods

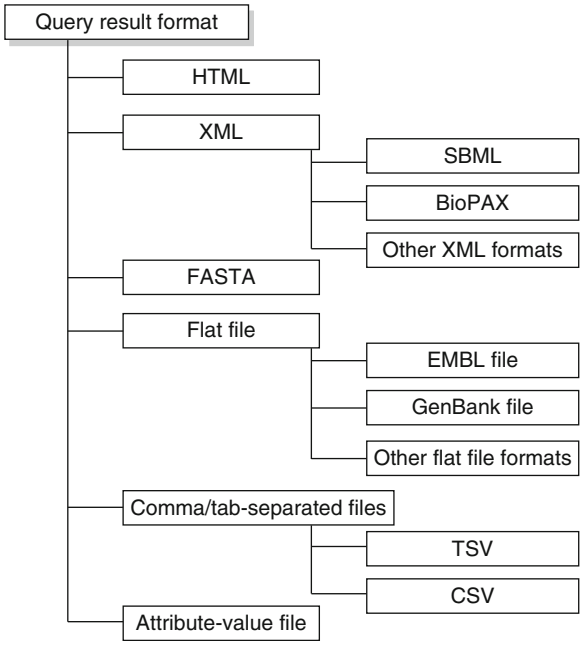
Analysis queries are more complex and, somehow, more typical of the biological domain. They consist in retrieving data based on *similarities* (similarity queries) and *patterns* (pattern search queries). The former ones take as input a DNA or a protein (sub)sequence and return those sequences found in the database that are the most similar to the input sequence. The latter ones take as input a pattern p and a DNA sequence s and return those subsequences of s which turn out to be most strongly related to the input pattern *p*.

Search Methods

A further classification criterion is related to methods used to query available databases, as illustrated in Fig. 4 where four main classes of query methods are distinguished: text-based methods, graphical interaction–based methods, sequence-based methods, and query language–based methods. The most common methods are the text-based ones, further grouped in two categories: *free text* and *forms*. In both cases, the query can be formulated specifying some keywords. With the free text methods, the user can specify sets of words, also combining them by logical operators. With forms, searching starts by specifying the values to look for that are associated to attributes of the database.

Result Formats

In genomic databases, several formats are adopted to represent query results (see the taxonomy in Fig. 5). Web interfaces usually provide answers encoded in HTML, but other formats are often available as well. For example, *flat files* are semistructured text files where each information class is reported on one or more consecutive lines, identified by a code used to characterize the annotated attributes. Often, special formats have been explicitly conceived for biological data. An example is the *FASTA* format (Mount 2004), commonly used to represent sequence data. More recently, in order to facilitate the spreading of information in heterogeneous contexts, the XML format is sometimes supported.



Genomic Databases, Fig. 5 Result formats

Cross-References

▶ Metabolic Networks, Databases

References

De Francesco E, Di Santo G, Palopoli L, Rombo SE (2009) A Summary of genomic databases: overview and discussion. In: Sidhu AS, Dillon TS (eds) Biomedical data and applications, studies in computational intelligence. Springer, Berlin, pp 37–54

Galperin MY (2007) The molecular biology database collection: 2007 update. Nucl Acids Res 35:D3–D4
GUS (2006) The Gen. Unified Schema docum. http://www.gusdb.org/documentation.php
Karp PD (2000) An ontology for biological function based on molecular interaction. Bioinformatics 16:269–285
Mount DW (2004) Bioinformatics: sequence and genome analysis, 2nd edn. Cold Spring Harbor Laboratory Press, Cold Spring Harbor
Twigger S, Lu J, Shimoyama M et al (2002) Rat genome database (RGD): mapping disease onto the genome. Nucl Acids Res 30(1):125–128
WormBase (2006) Db on biology and genome of C. Elegans http://www.wormbase.org/

Genomic Datasets

► Genomic Databases

Genomic Imprinting

Vani Brahmachari and Shruti Jain
Dr. B. R. Ambedkar Center for Biomedical Research, University of Delhi, Delhi, India

Synonyms

Differential expression of homologous genes; Parental-origin-effect; Transcription repression

Definition

Genomic imprinting is an epigenetic phenomenon due to which only one copy of the gene, inherited from the mother or the father is expressed. The term imprinting is borrowed from behavioral science, implying parental-origin effect on the genes. The transcriptional activity of genes inherited from the maternal side through the egg or oocyte could be different from that of the same genes inherited through the sperm, even if they have identical DNA sequence. Thus genomic imprinting is a special case of epigenetic inheritance where DNA methylation and histone modifications are responsible for differential regulation of the homologous genes or chromosomes and the epigenetic marking is transmitted through meiosis. Only a small subset of genes in the human genome are subjected to genomic imprinting. If a gene is said to be maternally imprinted it means that the copy of the gene coming from the maternal side is repressed, as per the convention in the field (Ideraabdullah et al. 2008).

Cross-References

► Epigenetics

References

Ideraabdullah FY, Vigneau S, Bartolomei MS (2008) Genomic imprinting mechanisms in mammals. Mutat Res 647(1–2): 77–85

Genomic Resources

Ricardo Cruz-Herrera del Rosario
Genome Institute of Singapore, Singapore, Singapore

Synonyms

Genome browsers

Definition

Genome browsers are graphical interfaces that allow the viewing of the DNA of sequenced species at different scales, ranging from a few bases, to thousands of bases at the level of genes, up to whole chromosomes. To display a large amount of heterogeneous data, genome browsers use the genomic coordinates as the main axis and data are displayed as different tracks that run parallel to it.

Characteristics

The data provided by genome browsers include gene annotations, transcript evidence, regulatory information, repeat elements, alignment with other genomes, epigenetic marks, genomic variants, etc. Users can select

specific tracks that are only relevant to the biological question they are trying to answer. Genome browsers may allow the retrieval of the data that they provide.

Genome browsers can be web-based or can be downloaded as a stand-alone program. Since their use involves huge amounts of raw data, most popular genome browsers are web-based as they do not require the user to download data locally. Another advantage of web-based genome browsers is that they can automatically update the data that they provide. Genome browsers can also be classified into multi-species or species-specific browsers. A short description of some of the most popular multi-species web-based browsers is given below.

The UCSC Genome Browser (www.genome.ucsc.edu; Kent et al. 2002) is one of the most widely used genome browsers worldwide. It provides vertebrate, deuterostome, insect, nematode and microbial genomes. It also provides a large amount of annotation data and functional data from next generation sequencing experiments. It allows the retrieval of almost all data that it provides.

The Ensembl browser (www.ensembl.org; Flicek et al. 2011) provides the genome and annotation of vertebrate genomes, but also has sister sites that provide metazoan, plant, fungi, and unicellular eukaryote and prokaryote genomes.

The VISTA browser (www.genome.lbl.gov/vista; Frazer et al. 2004) consists of a suite of programs and databases for comparative genomics.

References

Flicek P, Amode MR, Barrell D, Beal K, Brent S, Chen Y, Clapham P, Coates G, Fairley S, Fitzgerald S, Gordon L, Hendrix M, Hourlier T, Johnson N, Kähäri A, Keefe D, Keenan S, Kinsella R, Kokocinski F, Kulesha E, Larsson P, Longden I, McLaren W, Overduin B, Pritchard B, Riat HS, Rios D, Ritchie GR, Ruffier M, Schuster M, Sobral D, Spudich G, Tang YA, Trevanion S, Vandrovcova J, Vilella AJ, White S, Wilder SP, Zadissa A, Zamora J, Aken BL, Birney E, Cunningham F, Dunham I, Durbin R, Fernández-Suarez XM, Herrero J, Hubbard TJ, Parker A, Proctor G, Vogel J, Searle SM (2011) Ensembl 2011. Nucleic Acids Res 39(Database issue):D800–D806

Frazer KA, Pachter L, Poliakov A, Rubin EM, Dubchak I (2004) VISTA: computational tools for comparative genomics. Nucleic Acids Res 32(Web Server issue):W273–279

Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D (2002) The human genome browser at UCSC. Genome Res 12:996–1006

Genomics

▶ Disease System, Malaria

Genomics Network

▶ Functional/Signature Network Module for Target Pathway/Gene Discovery

Geometric Networks

▶ Stem Cell Networks

Gibbs Sampler

▶ Markov Chain Monte Carlo

Gillespie Algorithm

▶ Chemical Master Equation
▶ Stochastic Simulation Algorithm

Gillespie Stochastic Simulation

Ruiqi Wang
Institute of Systems Biology, Shanghai University, Shanghai, China

Definition

Consider the master equation

$$\frac{\partial p(X;t)}{\partial t} = \sum_{k=1}^{M} \{w_k(X-\theta_k)p(X-\theta_k;t) - w_k(X)p(X;t). \tag{1}$$

Although the analytical solution of the master equation is rarely available, the density function can be constructed numerically using the stochastic simulation

algorithm (SSA). Generally, the SSA first constructs numerical realizations and then averages the results of many realizations. The goal of stochastic simulation is then to describe the evolution of the state $X(t)$ from some given initial state $X(0)$.

The reaction probability density function $P(\Delta t, \mu|X;t)$ is the joint probability density function of two random variables, i.e., the time to the next reaction Δt and the index of the next reaction μ, given X. The reaction probability density function for the master equation takes the form

$$P(\Delta t, \mu|X;t) = w_\mu exp(-a_0(X)\Delta t) \tag{2}$$

with

$$a_0(X) = \sum_{j=1}^{M} w_j(X),$$

where $\Delta t \geq 0$ and $\mu = 1, \cdots M$.

The reaction probability density function provides the basis for the SSA.

According to the joint density function (Eq. 2), the next reaction and the time of its occurrence can be generated through the direct method. Draw two random numbers r_1 and r_2 from a uniform distribution in the unit interval [0, 1]. The time to the next reaction Δt and the index of the next reaction μ, given X, can be taken as follows:

$$\Delta t = \frac{1}{a_0(X)} ln(\frac{1}{r_1}), \tag{3}$$

$\mu =$ the smallest integer satisfying

$$\sum_{j'}^{\mu} w_{j'}(X) > r_2 a_0(X). \tag{4}$$

The Gillespie direct method for exact simulation of the master equation is as follows:

Step 1. Initialization: set $t = 0$ and fix the initial numbers of molecules $X(0)$.

Step 2. Calculate the propensity function w_k, $k = 1, \cdots M$.

Step 3. Generate two random numbers r_1 and r_2 in [0, 1].

Step 4. Determine Δt and μ according to (Eqs. 3 and 4).

Step 5. Execute reaction μ and advance time Δt, i.e., $t \leftarrow t + \Delta t$. If t reaches T_{max}, terminate the computation. Otherwise, go to Step 2.

References

Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. J Phys Chem 81:2340–2361

Glazier–Graner–Hogeweg Model

▶ Cellular Potts Model

GLM

▶ Generalized Linear Models

Global Linearization

▶ Quasilinearization

Global Maximum

▶ Global Optimum

Global Minimum

▶ Global Optimum

Global Network Alignment

Shihua Zhang[1] and Zhenping Li[2]
[1]National Center for Mathematics and Interdisciplinary Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
[2]School of Information, Beijing Wuzi University, Beijing, China

Synonyms

Local network alignment; Network alignment

Definition

Global network alignment problem is used to find the best overall alignment between the input networks. The mapping for it should cover all of the input nodes. Each node in an input network is either matched to one or more nodes in the other network(s) or explicitly marked as a gap node which has no match in another network (Singh et al. 2008; Zaslavskiy et al. 2009).

Cross-References

- ▶ Local Network Alignment
- ▶ Multiple Network Alignment
- ▶ Network Alignment

References

Singh R, Xu J, Berger B (2008) Global alignment of multiple protein interaction networks with application to functional orthology detection. Proc Natl Acad Sci USA 105(35): 12763–8

Zaslavskiy M, Bach F, Vert JP (2009) Global alignment of protein-protein interaction networks by graph matching methods. Bioinformatics 25(12):i259–67

Global Optimum

Lin Wang
School of Computer Science and Information Engineering, Tianjin University of Science and Technology, Tianjin, China

Synonyms

Global maximum; Global minimum

Definition

Optimum indicates minimum and maximum, respectively. Let $f : R^n \rightarrow R$ be a real valued function and $M \subset R^n$. A point $\bar{x} \in M$ is called a global minimum (respectively, global maximum) if there exists $f(\bar{x}) \leq f(x)$ (respectively, $f(\bar{x}) \geq f(x)$) for all $x \in M$.

References

Jongen HT, Meer K, Triesch E (2004) Optimization theory. Kluwer, Dordrecht

Global Sensitivity Analysis

Yunfei Chu and Juergen Hahn
Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX, USA

Definition

Global sensitivity analysis is an alternative to the widely used local sensitivity analysis to quantify parameter effects on the output. The advantage of global sensitivity analysis over the local counterpart is that parameter uncertainty is taken into account in the calculation of the sensitivity value and the sensitivity is not dependent on the nominal values of the parameters. Furthermore, global sensitivity analysis allows to vary several parameters simultaneously and over a large range to investigate their effects on the output. Global sensitivity analysis returns more detailed information about parameter effects and can also capture interactions among parameters.

Various techniques for global sensitivity analysis have been developed (Saltelli et al. 2008); some popular ones include the Morris' screening method, sampling-based approaches, and variance-based indices. All these techniques identify the effect of parameter variations on the output; however, each one looks at this problem from a different perspective.

The main challenge of global sensitivity analysis is the required computational effort. Many global sensitivity measures need to evaluate a multidimensional integral, and computationally intensive approaches such as the Monte Carlo method have to be applied to compute the integral.

Cross-References

- ▶ Optimal Experiment Design
- ▶ Signal Transduction Pathway

References

Saltelli A, Ratto M, Andres T, Campolongo F (2008) Global sensitivity analysis: the primer. Wiley, New York

Global Stability

Tianshou Zhou
School of Mathematics and Computational Sciences, Sun Yet-Sen University, Guangzhou, Guangdong, China

Definition

Global stability means that the attracting basin of trajectories of a dynamical system is either the state space or a certain region in the state space, which is the defining region of the state variables of the system. In other words, global stability means that any trajectories finally tend to the attractor of the system, regardless of initial conditions. Most of biological systems, e.g., gene regulatory systems, are needed to be globally stable.

To help understand global stability, here we give an example. Consider the famous Lorenz system:

$$\begin{aligned}\frac{dx}{dt} &= a(y-x)\\ \frac{dy}{dt} &= cx - xz - y\\ \frac{dz}{dt} &= xy - bz\end{aligned}$$

If $a = 10, b = 3, c = 28$, then we know that it has an attractor (called the Lorenz attractor). This attractor is globally stable since any trajectories beginning at initial points in the state space of the system finally tend to the attractor of the system.

Global stability is different from both the local stability of a steady state and structural stability, where the former describes how the trajectories near the steady state respond to a perturbation and the latter describes how the trajectories are changes when the parameters of a system are changes. Global stability belongs to a kind of asymptotic stability.

Global State, Boolean Model

Xi Chen, Wai-Ki Ching and Nam-Kiu Tsing
Advanced Modeling and Applied Computing Laboratory, Department of Mathematics, University of Hong Kong, Hong Kong, China

Synonyms

State

Definition

In a Boolean model, a global state is a gene state vector consisting of all the gene expression states. A Boolean model actually consists of a set of n nodes (where each node corresponds to a gene):

$$V = \{v_1,\ v_2,\ \ldots,\ v_n\}$$

and a list of Boolean functions (which represent the regulatory rules for nodes). Define $v_i(t)$ to be the state (0 or 1) of the node v_i at time t. Here we let:

$$\mathbf{v}(t) = (v_1(t),\ v_2(t),\ \ldots,\ v_n(t))^T$$

which is called the Gene Activity Profile (GAP). The GAP can take any possible form from the set:

$$S = \left\{(v_1,\ v_2,\ \ldots,\ v_n)^T :\ v_i \in \{0,\ 1\}\right\}$$

where each element in the set S is a global state, and thus totally there are 2^n possible (global) states in the network. An example of a two-gene network is given in Table 1. From the table, there are four global states in this network: (0, 0), (0, 1), (1, 0), and (1, 1).

Global State, Boolean Model, Table 1 The truth table

State	$v_1(t)$	$v_2(t)$	$f^{(1)}$	$f^{(2)}$
1	0	0	1	1
2	0	1	1	0
3	1	0	1	0
4	1	1	0	0

References

Shmulevich I, Dougherty E, Kim S, Zhang W (2002) From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. Proc IEEE 90:1778–1792

Glutathione

Rajeswara Babu Mythri[1], Shireen Vali[2] and M. M. Srinivas Bharath[1]
[1]Department of Neurochemistry, National Institute of Mental Health and Neurosciences (NIMHANS), Bangalore, Karnataka, India
[2]Cell Works Group Inc., Bangalore, India

Definition

Glutathione, a tripeptide (γ-L-glu-L-cys-gly; GSH), is the most abundant nonprotein thiol biomolecule in mammalian tissues. GSH also occurs as GSSG, the oxidized form of GSH and as GSSR representing GSH-cysteine disulphides linked to proteins. GSH mainly functions as a cellular antioxidant involved in detoxification of toxic free radicals (namely, peroxynitrite) and xenobiotics. GSH acts as an electron donor in the reduction of peroxides. Apart from this, GSH is also involved in maintenance of redox potential, transportation and storage of cysteine and as a cofactor. GSH has a role in signal transduction, cell proliferation, storage of nitric oxide, and regulation of gene expression. GSH also plays an important role in DNA metabolism, protein synthesis, activation of certain enzymes, and enhancement of immune function.

De novo synthesis of GSH in vivo occurs in the cytosol from the constituent amino acids in two consecutive steps catalyzed by γ-glutamyl cysteine ligase (GCL), the rate limiting enzyme and GSH synthase. Cellular GSH level is also contributed by reduction of GSSG to GSH. Conversely, GSH levels could be decreased when it is converted to GSH conjugates, GSSG, or by release from cells. Approximately 10% of cellular GSH is transported to the mitochondria by an energy-dependent mechanism. There exists a steady-state balance between synthesis and depletion of cellular GSH.

Compared to other organs in the human body, the brain is more susceptible to oxidative damage due to various biochemical and physiological factors. To prevent oxidative damage, GSH is present in millimolar concentrations in the brain; however, the concentrations are higher in the astrocytes compared to neurons. Age-dependent decline in GSH in the brain and cerebrospinal fluid has been observed in many organisms including humans. Further, the SN region of the midbrain has lower levels of GSH compared to other anatomical areas. However, during PD, there is a further decrease in SN GSH levels. GSH depletion is one of the first known indicators of oxidative stress and neurodegeneration in PD prior to selective inhibition of CI activity and DA loss. GSH depletion exacerbates the neurotoxicity of PD causing chemical toxins such as MPTP and 6-hydroxydopamine suggesting that disruption of the redox homeostasis of the cells triggers disease pathways in PD (Bharath et al. 2002).

Cross-References

► Disease System, Parkinson's Disease

References

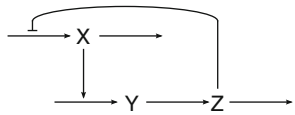
Bharath S, Hsu M, Kaur D, Rajagopalan S, Andersen J (2002) Glutathione, iron and PD. Biochem Pharmacol 64: 1037–1048

Goodwin Oscillator

Jinzhi Lei
Zhou Pei-Yuan Center for Applied Mathematics, Tsinghua University of Beijing, Beijing, China

Definition

Goodwin oscillator is a quintessential example of a biochemical oscillator based on negative feedback alone that was invented by Brain Goodwin (Goodwin 1965, 1966). The oscillator consists of mRNA, protein, and protein product (repressor). MRNA is the controlling factor for protein synthesis. Protein is the controlling factor for the production of protein product. The repression of mRNA synthesis by protein product follows the same law of surface adsorption as does protein inhibition (Fig. 1) (Goodwin 1965, 1966).



Goodwin Oscillator, Fig. 1 A schematic representation of the Goodwin oscillator

The kinetic equations describing the Goodwin oscillator are

$$\begin{aligned} \frac{dX}{dt} &= \frac{v_0}{1+(X/K)^p} - k_1 X \\ \frac{dY}{dt} &= v_1 X - k_2 Y \\ \frac{dZ}{dt} &= v_2 Y - k_3 Z \end{aligned} \tag{1}$$

Here X, Y, Z are concentrations of mRNA, protein, and protein product, respectively; v_0, v_1, and v_2 determine the rates of transcription, translation, and catalysis; k_1, k_2, and k_3 are rate constants for degradation of each component; $1/K$ is the binding constant of protein product to transcription factor; and p is a measure of the cooperativity of the repressor.

▶ Hopf bifurcation analysis in Goodwin's model shows that to obtain biochemical oscillation, the cooperativity of the negative feedback must be very high, say $p > 8$, and the degradation rate constants of the three components have to be nearly equal.

Bliss, Painter, and Marr (1982) fixed these problems by a slight modification of Goodwin's model:

$$\begin{aligned} \frac{dX}{dt} &= \frac{v_0}{1+X} - k_1 X \\ \frac{dY}{dt} &= v_1 X - k_2 Y \\ \frac{dZ}{dt} &= v_2 Y - k_3 \frac{Z}{1+Z/K}. \end{aligned} \tag{2}$$

Now the feedback step is no longer cooperative, and the uptake of protein product has a form of Michaelis-Menten function.

References

Bliss R, Painter P, Marr A (1982) Role of feedback inhibition in stabilizing the classical operon. J Theor Biol 97:177–193

Goodwin B (1965) Oscillatory behavior in enzymatic control processes. Adv Enzyme Regul 3:425–438

Goodwin B (1966) An entrainment model for timed enzyme synthesis in bacteria. Nature 209:479–481

GPGPU

▶ GPU Computing

GPU

Luca Lombardi and Piercarlo Dondi
Department of Computer Engineering and Systems Science, University of Pavia, Pavia, Italy

Synonyms

Visual processing unit (VPU)

Definition

A Graphic Processing Unit (GPU) is a specialized processor dedicated to the creation of the images visualized on the screen. It implements a set of the most frequently used graphics primitive operations, so as the CPU is not involved in the time-expensive graphical computation. GPUs, originally designed for personal computers, are currently used in many other kinds of devices, such as mobile phones, games consoles, tablets, or embedded systems. In a PC a GPU can be included as a standalone graphic card (usually in the mid-high-level solutions) or embedded in the motherboard (in low-level models). Finally, at the end of 2010, CPUs with integrated GPU were released: a solution particularly efficient to reduce communication times among the processors and to limit battery consumption in mobile devices (as notebooks or netbooks).

Cross-References

▶ High-Performance Computing, Structural Biology

GPU Computing

Fransisco Vázquez, José Antonio Martínez and Ester M. Garzón
Department of Computer Architecture and Electronics, University of Almería, Almería, Spain

Synonyms

General-Purpose GPU; GPGPU

Definition

The use of GPUs (Graphics Processing Units) to accelerate the computation of scientific and engineering applications.

Characteristics

At first the goal of the Graphics Processing Units (GPUs) was to accelerate the specific computation related to the visualization in the computer; in this way the CPU could be entirely dedicated to the computation of general applications. So GPU hardware resources were specialized in graphic computation and included several specific Arithmetic Units and a particular memory hierarchy connected to the central process unit (CPU). Later, the role of GPUs has been more relevant and the graphic software to exploit the GPUs was based on graphics-specific programming languages like OpenGL and Cg with a substantial cost of programming.

In the last decade the evolution of GPU technology has been based on the following:

1. The amount of hardware resources has been significantly increased.
2. The GPU was only focused on graphic computation, so it became an underutilized resource in the computer.
3. A wide range of applications in Science and Engineering share the characteristics of the graphic processes.

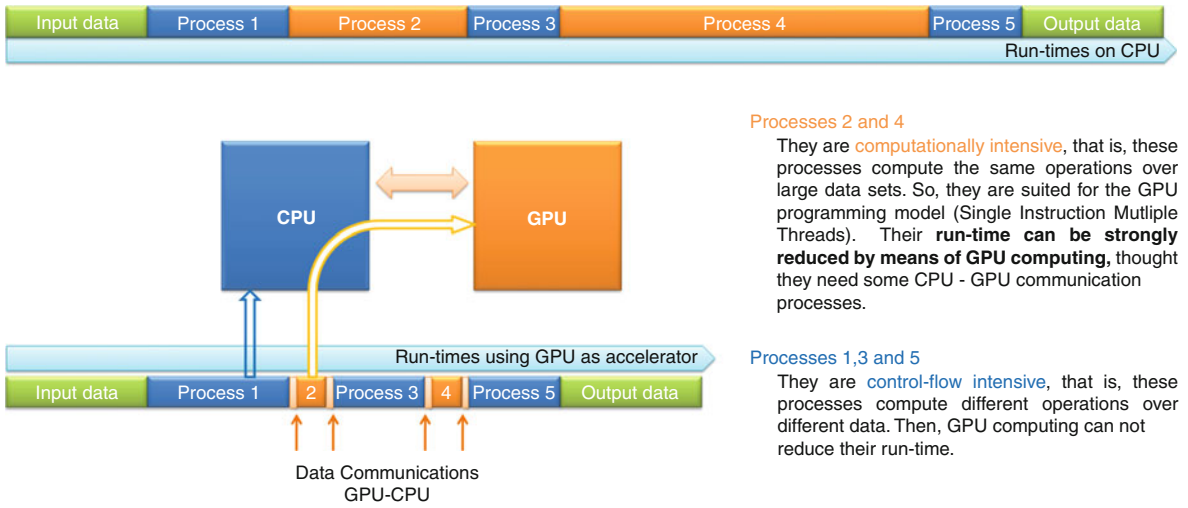
Thus, the main recent advances in GPU technology have focused on the development of Application Programming Interfaces (APIs), such as Compute Unified Device Architecture (CUDA) of NVIDIA, that greatly facilitate the programming of applications targeted at GPUs. The use of GPUs for general-purpose applications has exceptionally increased in the last few years due to the evolution of both GPU programming resources and the semiconductors technology. In this way, GPUs have emerged as new computing platforms that offer massive parallelism and provide incomparable performance-to-cost ratio for scientific computations (Kaeli and Leeser 2008). Currently, NVIDIA is leading the GPU computing and its interface CUDA is the key of thousands of applications which are accelerated by the NVIDIA GPUs; a relevant percentage of these applications are referred to the systems biology. CUDA is based on the programming model SIMT (Single Instruction, Multiple Threads), that is, every instruction in the program is executed by hundreds of threads with different data. The mapping of the threads on the GPU cores is automatically carried out by CUDA (Kirk and Hwu 2010).

An approach to facilitate the GPU programming is based on the use of basic routines or libraries which (1) compute the most used operations in the applications and (2) are optimally accelerated by GPU. In this line, NVIDIA supplies a wide set of routines related to several kinds of applications such as CuBLAS, CuFFT, and so on.

It should be noted that the computation related to every application can be classified in two kinds:

- *Computationally intensive*, if it includes large sequences of arithmetic-logic operations over large data sets. So, these operations can be computed by a large set of threads which are mapped on the high number of cores into the GPU and all of them execute the same sequence of instructions. Then, the SIMT programming model is suitable for the computation and it can be accelerated by the massive parallelism of the GPU (see Fig. 1).
- *Control-flow intensive*, if the process is dominated by control-flow operations, that is, decision points, in order to drive different instructions over different data. This computation is not suitable for the SIMT programming model and it is not accelerated by the GPU. This kind of computation can achieve better performance on the multicore CPU.

Consequently, the GPU computing cannot substitute the "CPU computing" because both can improve the performance of different kinds of computations included in the applications. So, currently, the



GPU Computing, Fig. 1 GPU computing can strongly accelerate the computationally intensive procedures in the program

high-performance computation (HPC) is based on heterogeneous computation (GPU-multicore computing) and the effort of programming is relevant in this context, because the programmers have to identify both types of computations and develop the program combining two parallel interfaces. However, if one kind of computation dominates the application, then, only one parallel interface can be considered.

GPU Computing for Systems Biology

Electron Tomography (ET) is taken as an illustrative example related to the systems biology in order to analyze the role played by GPU computing in this field. ET has emerged as the leading technique for the structural analysis of unique complex biological specimens. It combines electron microscopy with the power of 3D imaging. ET has made it possible to directly visualize the molecular architecture of organelles, cells, and complex viruses. Furthermore, ET allows the identification of the macromolecular assemblies in their native cellular environment and the study of their distribution in 3D as well as their interactions. ET has been crucial for recent breakthroughs in life sciences (see Lucic et al. (2005); Frank (2006) for reviews).

Computer-automated data collection has been essential for the advent of ET as a structural technique in cellular biology. It allows to automate specimen tilting, area tracking, focusing, and recording of images under low electron-dose conditions in order to preserve the specimen from radiation damages. But, as a consequence, the computed images are blurred, that is, they exhibit poor signal-to-noise ratio (SNR). Then, in ET it is necessary to apply either a simple 3D-reconstruction method which produces blurred images plus a sophisticated denoising method, or sophisticated 3D-reconstructions which are capable generating images with high resolution. Both alternatives need long run-times to compute the reconstruction because both are computationally intensive. So, HPC is paramount in this context to cope with those computational needs.

The GPU computing has emerged as a new HPC technique that offers massive parallelism. It is based on the GPUs platforms which are very suitable for their integration in laboratories of structural biology due to their incomparable performance-to-cost ratio and easy maintenance and use.

The specific methods in ET are computationally intensive and can be strongly accelerated on GPU platforms. However, it is necessary to reprogram them or even to propose a new approach to define the ET algorithms, in order to better exploit the parallelism of the GPU platforms. In this line, several ET approaches have already been proposed (Castaño-Diez et al. 2008; Vazquez et al. 2010; Xu et al. 2010).

Tomographic Reconstruction

Tomographic reconstruction can be modeled as a least square problem that can be solved by means of

algorithms based on matrix operations (Herman 1980). Large sparse matrices are involved in these algorithms. However, matrix data structures have not been traditionally included in the implementations due to their large memory requirements. As a consequence, the algorithms usually recompute of matrix coefficients when needed. Nonetheless, modern computers have large memory units available, so it is now possible to improve the performance of reconstruction algorithms by storing large matrices in core.

Assuming the single tilt axis geometry and using voxels to represent the volume to be reconstructed, the 3D reconstruction problem can be decomposed into a set of independent 2D reconstruction subproblems corresponding to the slices perpendicular to the tilt axis. Each of the 2D slices of the volume can then be computed from the corresponding set of 1D projections (usually known as sinogram). The reconstructed 3D volume is obtained by simply stacking the 2D slices.

The standard method to solve this problem is Weighted Back Projection (WBP) (Frank 2006). Briefly, the method uniformly distributes the object mass present at the projection images over computed backprojection rays. The intersection of the backprojection rays from the different images reinforces the density at the points where the mass is in the original structure. Therefore, the mass of the object is reconstructed. Formally, the backprojection can be defined by means of the matrix backprojection operator B as:

$$g = B * p^s \tag{1}$$

where B is a sparse matrix related with the projection geometry and p^s is the vector of sinograms for the different tilt angles of the slice s. When the number of tilt angles is large enough, the vector g is a good estimation of the slice g^*. Therefore, the 3D reconstruction consists of the following set of matrix-vector products:

$$g_s = B * p^s \; with \; 0 \leq s \leq Nslices \tag{2}$$

where *Nslices* is the total number of slices.

It is important to note that the matrix B is sparse and the location of nonzero coefficients exhibits some regularity and several kinds of symmetries. These characteristics are key to develop efficient matrix implementations of 3D WBP. At the beginning of 3D reconstruction process, the nonzeroes of B are computed and stored into one sparse matrix data structure. Afterward, that matrix is used to reconstruct all the slices by Sparse Matrix-Vector products (SpMV). So, the WBP method is computationally intensive and the acceleration of SpMV is the key to achieve a high performance.

This matrix formulation of WBP allows to use CUDA routines previously developed, which efficiently accelerate the operation SpMV on the GPU. The routine based on the format called ELLPACK-R achieves highest performance on GPUs (Vazquez et al. 2011) and facilitates the development based on CUDA of WBP. Additionally, the particular operation SpMV ($B * p^s$) involved in the matrix implementation of WBP has some specific characteristics that can be exploited to accelerate these operations with GPUs. Using the general implementation of SpMV on GPU based on ELLPACK-R as a starting point, the three geometry-related symmetry properties allow a significant reduction of the memory requirements to store the matrix B. The corresponding formats to compress the matrix related to these symmetries are referred as *sym1*, *sym2*, and *sym3*.

The matrix WBP approach has been implemented with CUDA and evaluated on a NVIDIA GeForce GTX 295 GPU. In order to evaluate the use of matrix WBP and GPU computing, the speedup factors against the standard recomputation-based WBP on the CPU were computed. As it is shown in the evaluation results at Fig. 2, matrix WBP on the GPU yield excellent net speedups up to 165x, especially for huge datasets as commonly used in the tomography field.

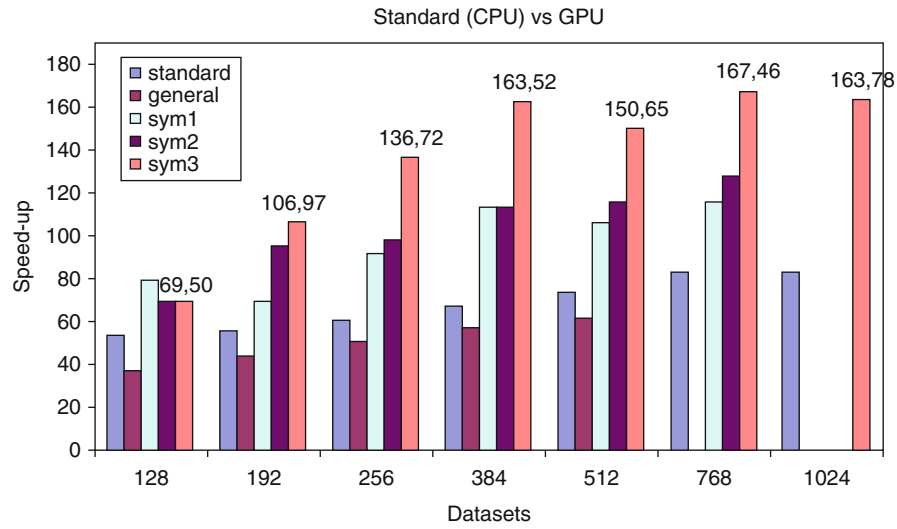
Denoising Method

Several simple linear methods, Gaussian or kernel-based filtering, reduce the noise at reasonable time but at the expense of blurring the structural features; then, in ET it is essential to apply nonlinear methods, such as the Anisotropic Nonlinear Diffusion or Beltrami filter, which preserve or even highlight the structural features. They are computationally intensive and GPU computing can relevantly reduce their run-times.

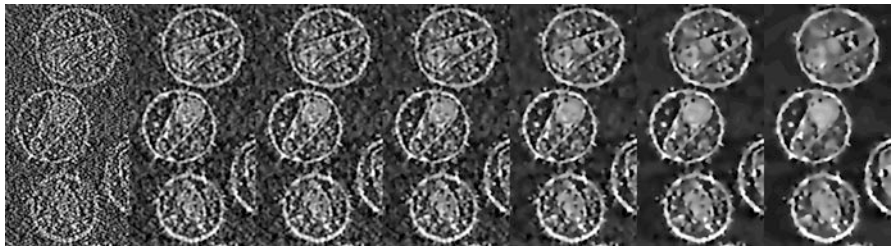
The Beltrami flow is a selective noise filtering method that preserves structural features; its formulation is based on the iterative solution of the following differential equation (Kimmel et al. 2000):

$$I_t = \frac{1}{\sqrt{g}} \mathrm{div}\left(\frac{\nabla \mathrm{I}}{\sqrt{g}}\right) \tag{3}$$

where $I_t = \partial I / \partial t$ denotes the derivative of the image density I with respect to the time t; $\nabla \mathbf{I}$ is the gradient



GPU Computing, Fig. 2 Effective speedup derived from different approaches (the standard based on recomputation and the four matrix approaches) on the GPU compared to the standard approach on the CPU



GPU Computing, Fig. 3 From *left* to *right*, the original HIV-1 reconstruction and the results with the noise reduced at 10, 25, 50, 100, 150, and 200 iterations are shown. Only a representative slice of the 3D reconstruction is presented

GPU Computing, Table 1 Run-times(s) of Beltrami Filter on a core of CPU based on 2 Quad Core Intel Xeon 2,26 Ghz and GPU code on one Tesla C1060 card for 3D images of dimension V × V × V

V	128	192	256	384	512	640	768
Sequential Beltrami	8,2	28,2	98,7	335,5	800,5	1552,1	2729,3
GPU Beltrami	0,5	1,0	2,1	5,7	14,6	22,9	38,7
Speedup	16,4	28,2	47,0	58,9	54,8	67,8	70,5

vector, that is $\nabla \mathbf{I} = (I_x, I_y, I_z)$, $I_x = \partial I/\partial x$ being the derivative of I with respect to x (similar applies for y and z); g denotes the determinant of the first fundamental form of the surface, which is $g = 1 + |\nabla \mathbf{I}|^2$; and div is the *divergence* operator.

Figure 3 is intended to illustrate the performance of this method in terms of noise reduction and feature preservation over a representative ET dataset that was taken from the Electron Microscopy Data Bank (http://emdatabank.org).

To sum it up, the Beltrami Method is translated into an iterative method where every step includes the same arithmetic operations over every voxel in the 3D image. So, GPU computing is capable of accelerating its computation due to a highly computationally intensive nature of the algorithm. This characteristic is frequently shared by most of the advanced filtering methods. The results in Table 1 show that the acceleration of Beltrami filter based on GPU computing is very relevant, specially for larger images.

Conclusions

Currently, the GPU computing is paramount to cope with the high computational needs in the

systems biology field. It is capable of accelerating the processes computationally intensive. However its performance decreases for control-flow intensive processes. There is a wide range of applications in systems biology which can be accelerated by GPU computing, but their software must be reprogrammed or even the algorithms need rethinking in order to adapt them to the GPU platforms.

Cross-References

- ▶ General-Purpose Computation, Graphics Processing Units
- ▶ High-Performance Computing, Structural Biology
- ▶ Multicore Computing

References

Castaño-Diez D, Moser D, Schoenegger A, Pruggnaller S, Frangakis AS (2008) Performance evaluation of image processing algorithms on the GPU. J Struct Biol 164:153–160

Frank J (2006) Electron tomography: methods for three-dimensional visualization of structures in the cell, 2nd edn. Springer, New York, p 455

Herman GT (1980) Image reconstruction from projections: the fundamentals of computerized tomography. Academic, New York

Kaeli DR, Leeser M (2008) Special issue on general-purpose processing using graphics processing units. J Parallel Distrib Comput 68:1305–1402

Kimmel R, Malladi R, Sochen NA (2000) Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images. Int J Comput Vis 39:111–129

Kirk DB and Hwu WW (2010) Programming Massively Parallel Processors: A Hands-on Approach. MorganKaufmann. pp 256

Lucic V, Forster F, Baumeister W (2005) Structural studies by electron tomography: from cells to molecules. Ann Rev Biochem 74:833–865

Vázquez F, Garzón EM, Fernández JJ (2010) A matrix approach to tomographic reconstruction and its implementation on GPUs. J Struct Biol 170:146–151

Vázquez F, Garzón EM, Fernández JJ (2011) A new approach for sparse matrix vector product on NVIDIA GPUs. Concurr Comput-Pract. doi:10.1002/cpe.1658

Xu W, Xu F, Jones M, Keszthelyi B, Sedat J, Agard D, Mueller K (2010) High-performance iterative electron tomography reconstruction with long-object compensation using graphics processing units (GPUs). J Struct Biol 171:142–153

Granular Computing

C. Maria Keet
KRDB Research Centre, Free University of Bozen-Bolzano, Bolzano, Italy

Definition

Granular computing is an emerging paradigm in computing and applied mathematics to process data and information, where the data or information are divided into so-called *information granules* that come about through the process of granulation. An information granule is a collection of entities that, in the scope of granular computing, usually originate from numerical analysis to group entities together at a certain level of granularity, thanks to their similarity, functional or physical adjacency, or indistinguishability.

The principal techniques used in granular computing are machine learning (▶ Identification of Gene Regulatory Networks, Machine Learning), fuzzy sets and logic (▶ Fuzzy Logic), rough sets, ▶ clustering, ▶ data mining, and, to a lesser extent, ontologies. Hence, the principal approach with granular computing is that of scale-based, quantitative, ▶ granularity with a focus on classifying instance data into their appropriate level of granularity as well as attempts to generate the best possible partitioning and, thereby, the optimal hierarchy of levels of granularity.

Meyers (2009) contains several entries for granular computing, and Yao (2010) provides a recent state of the art concerning theoretical foundations and applications of granular computing.

Cross-References

- ▶ Granularity

References

Meyers RA (2009) Encyclopedia of complexity and systems science. Springer, Berlin

Yao JT (ed) (2010) Novel developments in granular computing: applications for advanced human reasoning and soft computation. IGI Global, Hershey

Granularity

C. Maria Keet
KRDB Research Centre, Free University of Bozen-Bolzano, Bolzano, Italy

Definition

Granularity concerns the ability to represent and operate on different levels of detail in data, information, and knowledge that are located at their appropriate level. The entities are described relative to that level, which may be more coarse-grained or concern fine-grained details. Devising these ordered levels of granularity in a granular perspective are either determined by the laws of nature or are a resultant of human cognition to divide the data, information, or knowledge.

Characteristics

Introduction

Multiscale analysis of biological systems, such as in metagenomics, requires one to traverse from the molecular level of detail, to cells, bacterial communities, up to micro- and macro-environments and habitats. Longer established disciplines, such as plant and animal taxonomy, categorize specimens in the tree that contains more (Genus-level) or less (e.g., Family-level) details. That biology concerns different levels of detail and hierarchical systems has been noted widely (Vogt 2010; Salthe 1985), and efforts have gone into the characteristics of granularity, how one can model it, and how to effectively use it to manage the large amounts of data, information, and knowledge.

Granularity is relatively static: once the granular levels and hierarchies in a subject domain are characterized, it is a static structure (a *granularity framework*) either imposed on the data, information, or knowledge being granulated or based on it being inherent in the entities in reality themselves. Entities can undergo changes and thereby move from a finer-grained level to a coarser-grained one, or vice versa, but are not present at two levels in the same hierarchy at the same time; e.g., a single cell at the Cell-level of granularity develops into a multicellular organism at the Organism-level. Entities are somehow "assigned to" a certain level. The first basic questions that arise, then, are: *what are the components that have to do with granularity, and how does one obtain them?* The answers to these questions may depend on the emphasis one takes regarding granularity, where the principal dimensions are:

1. Arbitrary scale versus non-scale-dependent granularity, roughly fitting with quantitative versus qualitative granularity
2. How levels, and its contents, in a granular perspective relate to each other
3. Difference in emphases, being focused on either the entities, their relations, or the criterion for granulation
4. The (mathematical) representation, such as based on set theory, ▶ mereology, or an encompassing framework that accommodates both

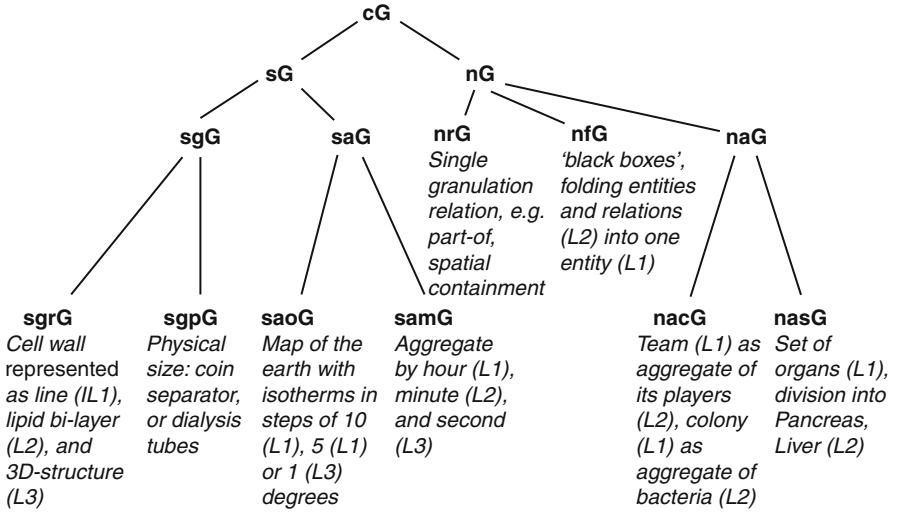
Such different emphases have brought forward various proposals for representing granularity and, to a lesser extent, how one can manage the granulated data, information, and knowledge.

Granularity, Table 1 Sample granular perspective for human structural anatomy (criterion) granulated by parthood (nrG type of granularity), with six granular levels and sample entities residing at each level (which also could be their respective instances)

Name of level	Sample contents of each level
Body	Male human body
Organ	Liver, pancreas
Tissue	Epithelium, smooth muscle
Cell	Erythrocyte, melanocyte
Organelle	Ribosome
Molecule	β-galactosidase, Insulin

Theories of Granularity

Theories of granularity are predominantly informal and subject domain oriented (Salthe 2001), with a few exceptions that focus on the subject domain independent logic-based representation (Bittner and Smith 2003; Keet 2008). Thus far, the most comprehensive effort to find answers to the two aforementioned questions is proposed by Keet (2008), which takes into account the four distinct emphases regarding granularity and it provides a formal characterization informed by ontology. The principal "ingredients" for any granularity *framework*, are the *entities* that are assigned to different *levels of granularity* that, in turn, form a hierarchy of levels so that one obtains



Granularity, Fig. 1 Top-level taxonomy of types of granularity, each with one or more examples, where *Lx* stands for a particular level in the granular perspective

a *granular perspective* on a particular domain of interest, provided the granulation is carried out in a consistent manner using a particular *criterion for granulation* and *type of granularity* (mechanism of granulation, see below). It can be proven that to be able to have an instance of a granularity framework, it requires at least two adjacent levels of granularity in a granular perspective; an informal example of a granular perspective is shown in Table 1. The granular perspectives within one granularity framework can be linked to each other, provided the contents in the levels of different perspectives overlap; e.g., Insulin is located at the Molecule-level as a subtype of Peptide and as a subtype of Hormone in a functional perspective.

Granulation is the act of devising the granular levels and perspectives and assigning contents to those levels, which can be done manually or computationally. For data and information granulation, the automated approach is generally called ▶ granular computing (Yao 2010), whereas at the knowledge layer with its entities at the intensional level (classes, concepts, relationships, etc.), one uses processes such as ▶ modularization, ▶ abstraction, and expansion that are constrained by the criterion for granulation and type of granularity that together determine uniqueness of a granular perspective.

Types of Granularity

Good granular perspectives adhere to underlying principles regarding how the levels are identified. This can be structured in a taxonomy of types of granularity that categorizes the mechanism of granulation, of which the eight principle types are described in Fig. 1 and Table 2 (see Keet 2008, Chap. 2 for motivation and explanation). The main distinction is between quantitative and qualitative granularity. Concerning qualitative granularity, nrG's granulation relations include part-whole relations that are either a type of true parthood (▶ Mereology) or motivated by linguistics (▶ Meronymy), such as structural part-of (e.g., Table 1), contained in, and being a member of something. Examples for nfG are the MAPK cascade and the Second messenger system with its components, and clustering in ER models. nacG's aggregate generally is termed with a meaningful collective noun and such

Granularity, Table 2 Distinguishing characteristics at the branching points of the top-level taxonomy of types of granularity of Fig. 1

Branching point	Distinguishing feature
sG – nG	Scale (quantitative) – non-scale (qualitative)
sgG – saG	Grain size (scale *on* entity) – aggregation (scale *of* entity)
sgrG – sgpG	Resolution – size of the entity
saoG – samG	Overlay aggregated – entities aggregated according to scale
naG – nrG – nfG	Semantic aggregation – one type of relation between entities in different levels – different type of relation between entities in levels and relations among entities in level
nacG – nasG	Parent-child not taxonomic and relative independence of contents of higher/lower level – parent-child with taxonomic inheritance

Granularity, Table 3 Typical classification levels in ecosystems using as criteria for granulation similar spatial scales and subdivisions in biotic and abiotic environments, resulting in seven granular perspectives (depicted as columns) that each contains eight granular levels or less (rows). Each named level contains instances, such as Palearctic and Afrotropic in the Ecozone-level

	Biotic			Abiotic		
Ecosystem	Biogeography	Zoogeography	Phytogeography	Physiography	Geology	Pedology
Ecozone	Biome		Floral kingdom			
Ecoprovince		Zoogeographic province	Floral province		Geoprovince	
Ecoregion	Bioregion		Floral region	Physio-region	Georegion	Pedoregion
Ecodistrict						
Ecosection						
Ecosite						
Ecotope	Biotope	Zootope	Phytotope	Physiotope	Geotope	Pedotope
Ecoelement	Bioelement				Geoelement	

that the instances of the aggregate are different from instances of its members and a change in its members does not change the meaning of the whole. sgrG's grain size with respect to resolution can also be applied in GIS; e.g., the city of Paris is represented on cartographic maps as polygon or as point. With sgpG, one focuses on the physical size of the entities themselves; within one level one can distinguish instances of, say, <5 and ≥ 1 mm, but instances <1 mm are indistinguishable from each other (but are distinguishable in a finer-grained level), such as sieves with different pore sizes or two objects touching each other, like the wallpaper and the wall where, when zoomed in, we also observe the glue that connects the wallpaper to the wall. samG has an associated mathematical function for measured aggregation (1960s in a minute and so forth). An example of granulation motivated by scale is included in Table 3.

Challenges

Currently, there is still a gap between the ▶ knowledge representation of granularity with its (onto)logical foundations and implementations (▶ Granular Computing), being how to relate structured thinking and structured problems solving, respectively (Yao 2005), in particular with respect to dealing with attributes in granular computing and its ontological counterpart of criterion of granulation. In addition, conditional granularity across granular perspectives is to be addressed in applications, such as linking time granularity (Euzenat and Montanari 2005) together with qualitative granularity that already can be modeled unambiguously with the theory of granularity (Keet 2008; Vogt 2010). Examples for its need are abound: for instance, in a multiscale analysis, one has to know

Granularity, Table 4 Two abbreviated granular perspectives (table columns), one for plant anatomy with parthood and one time granularity, which are linked at three levels in the hierarchy (denoted with "⇔") so that cross-granular conditions can be asserted and, e.g., information retrieved from the information system

Plant sample		Time granularity
Tissue	⇔	Day
↕		↕
Cell	⇔	Hour
↕		↕
Molecule	⇔	Millisecond

that, say, biological sample analysis of a cell culture – be it the structural components or the processes it is involved in – yields information at a time granularity of hours (Table 4).

The interaction of granularity with notions such as ▶ emergence, ▶ holism, and ▶ reduction is to be explored further. Granularity can provide a methodological modeling framework to enable structured examination of claims of emergence both from a formal ontological modelling and computational angle by making the complex at least less complex, and aids understanding which levels are essential for explanation of some property of observed behavior.

Cross-References

- ▶ Abstraction
- ▶ Emergence
- ▶ Granular Computing
- ▶ Holism

- ▶ Mereology
- ▶ Meronymy
- ▶ Modularity
- ▶ Modularization
- ▶ Organism State, Lymphocyte
- ▶ Reduction
- ▶ Top-Down Decomposition of Biological Networks

References

Bittner T, Smith B (2003) A theory of granular partitions. In: Duckham M, Goodchild MF, Worboys MF (eds) Foundations of geographic information science. Taylor & Francis, London, pp 117–151

Euzenat J, Montanari A (2005) Time granularity. In: Fisher M, Gabbay D, Vila L (eds) Handbook of temporal reasoning in artificial intelligence. Elsevier, Amsterdam, pp 59–118

Keet CM (2008) A Formal theory of granularity. Ph.D. Thesis, KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy

Salthe SN (1985) Evolving hierarchical systems – their structure and representation. Columbia University Press, New York

Salthe SN (2001) Summary of the Principles of Hierarchy Theory. http://www.nbi.dk/natphil/salthe/hierarchy_th.html. Accessed 10 October 2005.

Vogt L (2010) Spatio-structural granularity of biological material entities. BMC Bioinformatics 11:289

Yao YY (2005) Perspectives of granular computing. IEEE Conf Granul Comput (GrC2005) 1:85–90

Yao JT (ed) (2010) Novel developments in granular computing: applications for advanced human reasoning and soft computation. IGI Global, Hershey

Graph

Jan Ramon
Declarative Languages and Artificial Intelligence Group, Katholieke Universiteit Leuven, Leuven, Belgium

Definition

A graph is an abstract representation of a set of objects, some of which are connected by links. The objects are represented with vertices (also called nodes) and the links with edges or arcs. Several types of graphs are used depending on what is suitable for a particular application.

An undirected graph is a tuple (V,E) where V is a set of vertices and $E \subseteq \{\{x,y\}x,y \in V\}$ is a set of edges.

A directed graph is a tuple (V,E) where V is a set of vertices and $E \subseteq \{(x,y)x,y \in V\}$ is a set of arcs. Therefore, (x,y) and (y,x) are different arcs, while for an undirected graph, $\{x,y\}$ and $\{y,x\}$ are two representations of the same edge.

An edge $\{v,w\}$ (or arc (v,w)) is said to be incident with vertices v and w. v and w themselves are said to be adjacent. The degree of a vertex is the number of edges (arcs) incident with it.

A labeled graph is a tuple (V,E,λ) where (V,E) is a graph (either directed or undirected) and $\lambda: V \cup E \rightarrow \Sigma$ is a labeling function assigning to all vertices and edges (or arcs) labels from some alphabet of labels Σ.

A hypergraph is a graph whose edges (arcs) can link more than two vertices. A multigraph is a graph where there may be more than one edge (arc) between a given pair of vertices.

Graph theory is the study of the properties of graphs (Diestel 2002). Algorithmic graph theory is the study of algorithms to compute properties of graphs (Gross and Yellen 2004). ▶ Graph mining is the study of performing data mining on and machine learning from data represented with graphs.

References

Diestel R (2010) Graph theory. Springer, Heidelberg

Gross JL, Yellen J (2004) Handbook of graph theory. CRC Press, Boca Raton

Graph Algorithms in Network Analysis

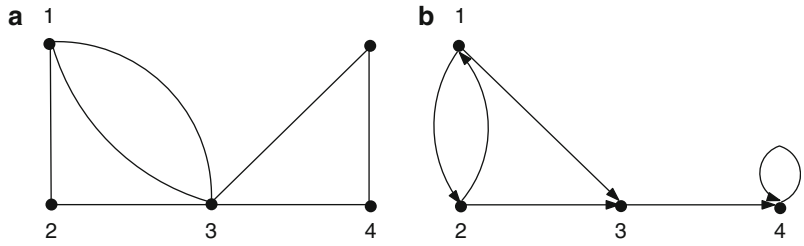
Seok-Hee Hong
School of IT, University of Sydney, Sydney, NSW, Australia

Characteristics

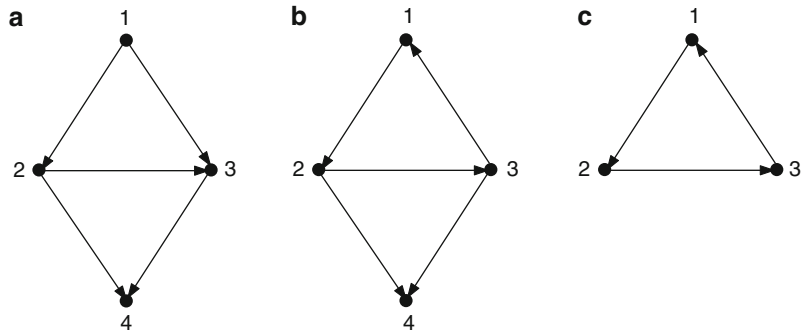
Graph Theory

We first define necessary terminologies in graph theory. For more details, see (Bondy and Murty 1976; West 2001).

A *graph* is a popular mathematical structure for modeling entities and the relationships between them. A graph $G = (V, E)$ consists of a set of



Graph Algorithms in Network Analysis, Fig. 1 Example of (**a**) an undirected graph, and (**b**) a directed graph



Graph Algorithms in Network Analysis, Fig. 2 Example of (**a**) a DAG, (**b**) a directed graph with a cycle, and (**c**) the directed cycle in (b)

vertices (or nodes) V and a set of *edges* E, where an edge $e \in E$ is defined by a pair of vertices $v, w \in V$ and denoted by $e = (u, v)$. We say that an edge $e = (u, v)$ is *incident* to vertices u and v, and that u and v are *adjacent* to each other. The vertices u and v are called the *endvertices* of e. The *degree* of a vertex $v \in V$ in a graph G is the number of edges incident to v. For example, vertex 3 of the graph shown in Fig. 1a has degree 5.

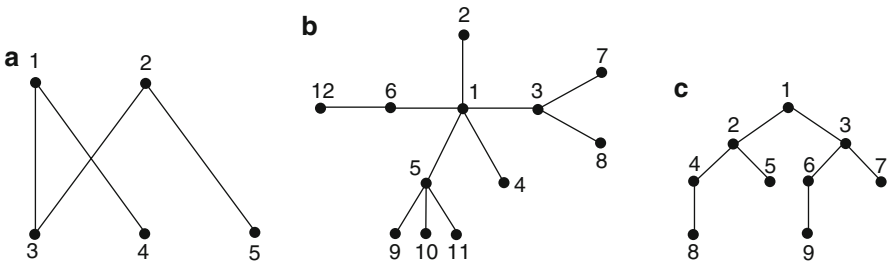
An *undirected* graph is a graph in which the edges have no orientation. A *directed* graph (or *digraph*) is a graph in which the edges have an orientation, i.e., an edge $e = (u, v)$ is defined by an *ordered* pair of two vertices $v, w \in V$. For example, Fig. 1a shows an undirected graph, and Fig. 1b shows a directed graph. A *weighted* graph is a graph in which the edges have weights, such as integers or real numbers.

A *loop* is an edge e whose endvertices are the same vertex, i.e., $e = (v, v)$. An edge $e = (u, v)$ is called a *multiple* edge, if there is another edge $e' = (u, v)$ with the same endvertices u and v. A *multigraph* is a graph with multiple edges. For example, the undirected graph in Fig. 1a has multiple edges between vertex 1 and vertex 3, and the directed graph in Fig. 1b has a loop at vertex 4. A *simple* graph has no multiple edges or loops. In most cases, the graph refers to a simple undirected graph.

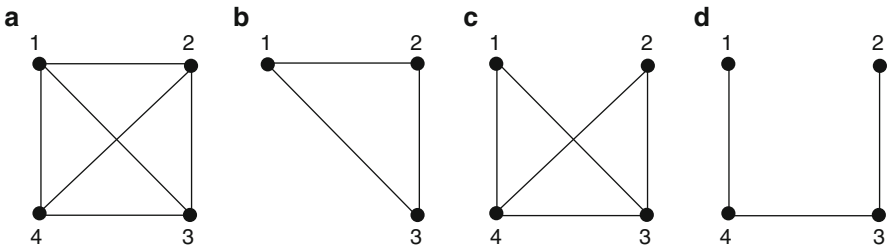
A *path* in a graph is a sequence of vertices such that each vertex in the sequence is connected by an edge to the next vertex in the sequence. The *length* of a path is the number of edges in the path. A *cycle* is a path where the starting vertex is the same as the ending vertex. A path or cycle is called *Hamiltonian* if it visits all the vertices of the graph exactly once. A graph is *acyclic* if it contains no cycles. A directed acyclic graph is called a DAG. For example, the directed graph in Fig. 2a is a DAG, while the directed graph in Fig. 2b is not a DAG, since it contains the directed cycle shown in Fig. 2c.

A *tree* T is a connected simple graph with no cycles. A vertex of degree 1 in a tree is called a *leaf*. A non-leaf vertex is called an *internal* vertex. A *rooted* tree has a special vertex called the *root*, and is often treated as a DAG with the edge directions originating from the root. A *k-ary tree* is a rooted tree in which every internal vertex has at most k children. In particular, a 2-ary tree is called a *binary tree*. For example, Fig. 3b shows a tree, and Fig. 3c shows a rooted binary tree.

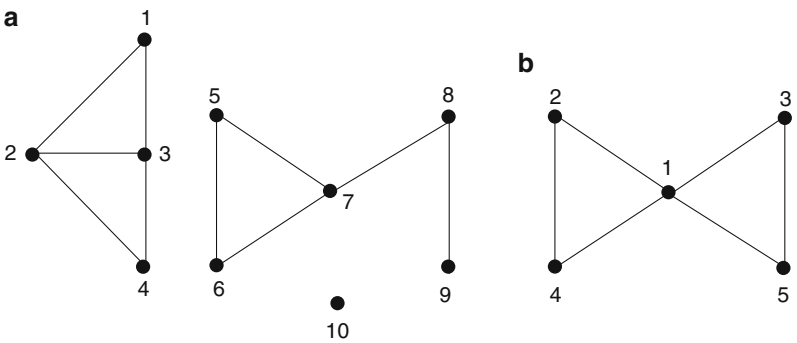
A *bipartite graph* $G = (V, W, E)$ consists of two disjoint vertex sets V and W (i.e., no two vertices in V are adjacent and no two vertices in W are adjacent), and an edge set E, where every edge $e = (v, w)$ has an endvertex $v \in V$ and the other endvertex $w \in W$. Figure. 3a shows an example of a bipartite graph.



Graph Algorithms in Network Analysis, Fig. 3 Example of a (**a**) bipartite graph, (**b**) a tree, and (**c**) a rooted binary tree



Graph Algorithms in Network Analysis, Fig. 4 Example of (**a**) a clique, (**b**) a subgraph of (**a**), (**c**) a spanning subgraph of (**a**), and (**d**) a spanning tree of (**a**)



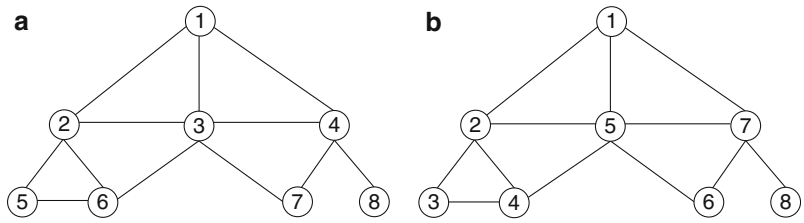
Graph Algorithms in Network Analysis, Fig. 5 Example of (**a**) a disconnected graph, and (**b**) a graph with a cut vertex

A *regular graph* is a graph in which every vertex has the same degree. For each n, the *complete graph*, denoted by K_n, is a simple graph with n vertices in which every vertex is adjacent to all the other vertices.

An (*induced*) *subgraph* $G' = (V', E')$ of a graph $G = (V, E)$ is a graph where V' is a subset of V, E' is a subset of E, and all endvertices of $e \in E'$ are in V'. A subgraph G' is a *spanning* subgraph of G if $V = V'$. A *spanning tree* is a spanning subgraph that is a tree. For example, Fig. 4a shows complete graph K_4, and Fig. 4b shows a subgraph of K_4. Fig. 4c shows a spanning subgraph of K_4, and Fig. 4d shows a spanning tree of K_4.

A pair of vertices v and w in a graph G is *reachable*, if there is a path between v and w in G. A graph G is *connected*, if every pair of vertices is reachable; otherwise, it is *disconnected*. A directed graph is *weakly connected*, if it contains an undirected path for every pair of vertices. It is *strongly connected*, if it contains a directed path for every pair of vertices. A *cut vertex* is a vertex whose removal disconnects the remaining subgraph. A *bridge* (or *cut edge*) is an edge whose removal disconnects a graph. If a graph is still connected after removing any $k-1$ vertices, then the graph is called *k-connected*. For example, the graph in Fig. 5a is disconnected, and the graph in Fig. 5b is connected, with vertex 1 as a cut vertex.

The *distance* $d(u, v)$ between two vertices u and v in a graph G is the length of the shortest path between them. The *eccentricity* of a vertex v in a graph G is the



Graph Algorithms in Network Analysis, Fig. 6 Example of (**a**) Breadth-first search (BFS) and (**b**) Depth-first search (DFS)

maximum distance from v to any other vertex. A *center* is a vertex with the minimum eccentricity. The *diameter* of a graph G is the maximum eccentricity over all vertices in that graph. For example, the graph in Fig. 5b has a diameter 2, and vertex 1 is the center of the graph. The distance between vertex 2 and vertex 5 is 2.

Graph Algorithms

We now briefly explain basic graph algorithms. For more details on the full description of each algorithm, see (Cormen et al. 2001; Goodrich and Tamassia 2001).

Graph Traversal

A graph traversal is a method of visiting all the vertices of a graph G, starting from a given vertex v, where the elementary move is from one vertex to another along an edge connecting them. There are two well-known traversal methods: breadth-first search (BFS) and depth-first search (DFS).

BFS begins at a chosen vertex v, and visits all neighbor vertices $u_1, u_2, \ldots, u_k$ adjacent to v; then it recursively visits all the neighbors of u_i, until it explores all the vertices of G that are reachable from v. In this way, BFS visits all the vertices of distance k from v, before it visits the vertices of distance $k + 1$ from v.

BFS produces a *breadth-first tree* with root v which contains all the reachable vertices u from v. The path from v to u in the breadth-first tree corresponds to the *shortest path* (i.e., the path containing the smallest number of edges) from v to u in G.

DFS explores a graph G from a chosen vertex v by searching deeper in the graph whenever possible; it traverses the *depth* of G before the *breadth* of G. DFS explores unexplored edges from the most recently visited vertex u, until it explores all the edges from u. It then backtracks to a vertex w, that was discovered from v, and explores the other unexplored edges from w. DFS repeats this process until it visits all the vertices that are reachable from v.

Both the DFS and BFS algorithms can be implemented to run in $O(|V| + |E|)$ time, i.e., linear in the size of the graph G. For details, see (Cormen et al. 2001; Goodrich and Tamassia 2001). For example, Fig. 6 depicts examples of BFS and DFS, where the number of a vertex represents the ordering performed by BFS and DFS.

Connected Components

A *connected component* of an undirected graph $G = (V, E)$ is a maximal connected subgraph $G' = (V', E')$ of G in which any two vertices u and v in G' are reachable from each other (i.e., they are connected by a path in G'). For example, the graph shown in Fig. 5a has three connected components: One consists of vertices $\{1, 2, 3, 4\}$, and the others consists of vertices $\{5, 6, 7, 8, 9\}$ and an isolated vertex 10.

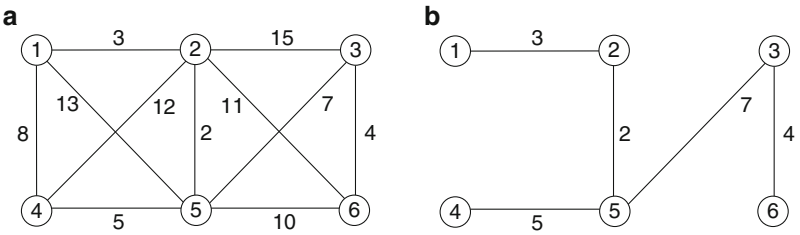
A *strongly connected component* of a directed graph $G = (V, E)$ is a maximal strongly connected subgraph $G' = (V', E')$ of G in which every two vertices u and v in G' are reachable from each other in both directions (i.e., they are connected by a directed path from u to v and a directed path from v to u).

It is easy to compute the connected components of an undirected graph G in linear time, by using either BFS or DFS. One can use DFS to compute the strongly connected components of a directed graph G in linear time. For details, see (Cormen et al. 2001; Goodrich and Tamassia 2001).

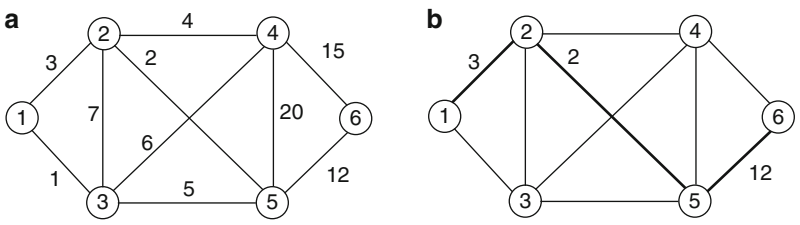
Minimum Spanning Tree

A minimum spanning tree (MST) of a connected weighted undirected graph G is a spanning subtree T of G that connects all the vertices of G with the minimum total edge weight, among all possible spanning trees of G. For example, Fig. 7b shows a minimum spanning tree of a graph in Fig. 7a.

There are two well-known algorithms to compute a MST of a weighted undirected graph: Prim's algorithm and Kruskal's algorithm. Both algorithms are *greedy algorithms*, i.e., they make the best possible



Graph Algorithms in Network Analysis, Fig. 7 (**a**) A weighted connected graph, and (**b**) its minimum spanning tree



Graph Algorithms in Network Analysis, Fig. 8 Example of a shortest path: (**a**) a weighted graph, and (**b**) the shortest path between vertices 1 and 6

available choice at each stage of the algorithm. More specifically, at each step of the algorithm, they add one *safe* edge at a time (i.e., an edge that does not create a cycle), preserving the minimum weight spanning tree property.

Kruskal's algorithm first sorts the edges in a nondecreasing order by their weights. It then finds a safe edge to add to the growing *forest* (i.e., a set of disconnected trees), by finding an edge $e = (u, v)$ with the minimum weight among all the edges that connect any two trees in the forest.

Prim's algorithm maintains a single tree $T = (V_T, E_T)$ at each stage of the algorithm. The tree initially only contains an arbitrary vertex v and grows until it spans all the vertices in the graph. At each step, it finds a safe edge with the minimum weight among all edges $e = (u, v)$, where $u \in V - V_T$ and $v \in V_T$ (i.e., an edge connecting a vertex in T and a vertex in $G - T$).

The running time of Prim's algorithm and Kruskal's algorithm depends on the data structures that are used. For example, Prim's algorithm can be implemented to run in $O(|E|log|V|)$ time, using a binary heap data structure. Kruskal's algorithm can be implemented to run in $O(|E|$ $log|V|)$ time, using a disjoint-set data structure. For faster implementations using more complex data structures, see (Cormen et al. 2001; Goodrich and Tamassia 2001).

Shortest Path

The aim of the shortest path problem is to find a path between two vertices in a weighted graph, such that the sum of edge weight is minimized. For example, Fig. 8b shows a shortest path between vertex 1 and vertex 6 of the graph in Fig. 8a.

There are two problems: the single-source shortest path problem and the all-pair shortest path problem.

Single-Source Shortest Path The aim of the single-source shortest path problem is to find shortest paths from a chosen source vertex v to all the other vertices in the graph. For unweighted graphs, one can use breadth-first search algorithm to compute a shortest path from v to all the other vertices. For weighted graphs, there are two well-known algorithms: Dijkstra's algorithm and the Bellman-Ford algorithm.

Both algorithms produce a shortest path tree, starting from the *source* vertex v to all the other reachable vertices in the graph. Each vertex u has a value on shortest path estimate (initially assigned with a very large value). Each step of the algorithms tries to decrease the estimate value, using the *relaxation* technique. More specifically, relaxing an edge $e = (u, w)$ tests whether one can improve the shortest path from v to w by rerouting the path through u.

Dijkstra's algorithm solves the single-source shortest path problem on a weighted directed graph $G = (V, E)$, with nonnegative edge weights. It also uses a greedy approach, and maintains a set S of vertices whose final shortest-path weights from v have already been determined. The algorithm repeatedly chooses a vertex $u \in V - S$ with the minimum shortest-path estimate, adds u to S, and relaxes all the edges leaving u. Dijkstra's algorithm can be

implemented to run in $O(|V|log|V| + |E|)$ time. For details, see (Cormen et al. 2001; Goodrich and Tamassia 2001).

The Bellman-Ford algorithm solves the single-source shortest path problem on a weighted directed graph $G = (V, E)$, where the edge weights may be negative. It either detects negative-weight cycles, or progressively decreases an estimate on the weight of a shortest path from v until it discovers the actual shortest-path weight. The running time of the algorithm is $O(|V||E|)$. For details, see (Cormen et al. 2001; Goodrich and Tamassia 2001).

All-Pair Shortest Path The aim of the all-pair shortest path problem is to find the shortest paths between every pair of vertices u and v in the graph. The problem can be solved by repeatedly using the algorithm for the single-source shortest path problem, for each vertex $v \in V$ as a source vertex.

Alternatively, the Floyd-Warshall algorithm solves the all-pairs shortest path problem in $O(|V|^3)$ time using a *dynamic programming* approach. The algorithm operates on directed graphs that may have negative-weight edges, but do not contain any negative-weight cycles. For details, see (Cormen et al. 2001; Goodrich and Tamassia 2001).

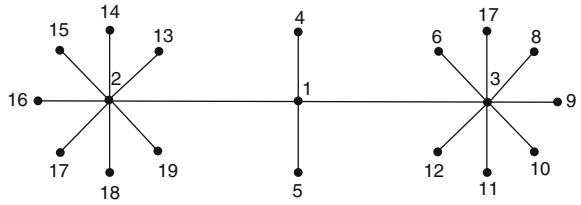
Network Analysis

A network is an extension of the concept of a graph. The network sometimes contains more information, such as vertex attributes and edge attributes. Network analysis spreads from social sciences to complex systems, communication networks, bioinformatics, transportation systems, and project planning.

Roughly speaking, there are three different levels of analysis:

- Individual-level: Examples include centrality measures such as degree centrality and betweenness centrality.
- Group-level: Examples include clique analysis, k-core analysis, structural equivalence, and blockmodeling.
- Network-level: Examples include network statistics (such as degree distribution, diameter, average path length, and clustering coefficient) and network comparison (such as graph isomorphism and similarities).

In the following, we describe some of the basic concepts and methods in network analysis. For details on social network analysis, see (Wasserman and Faust 1994). For algorithmic aspects of network analysis, see (Brandes and Erlebach 2005). For network analysis of biological networks, see (Junker and Schreiber 2008).



Graph Algorithms in Network Analysis, Fig. 9 Example of centrality analysis

Individual-level Analysis

The *centrality* index aims to compute the *importance* of actors in a social network. There are many different centrality measures available, which are based on the definition of importance in specific applications. For definitions of various centrality measures, see (Brandes and Erlebach 2005; Wasserman and Faust 1994). For algorithms for computing each centrality measure, see (Brandes and Erlebach 2005).

Degree Centrality For undirected graphs, the *degree centrality* of a vertex v is defined as the degree of v. For example, in the undirected graph in Fig. 9, vertices 2 and 3 have the highest degree centrality.

For directed graphs, there are two variants of degree centrality: the *in-degree* centrality (i.e., the number of incoming edges) and the *out-degree* centrality (i.e., the number of outgoing edges).

Closeness Centrality In a social network analysis, a vertex with a small total distance maybe more important than a vertex with a high total distance. More formally, *closeness centrality* of a vertex u can be defined as follows:

$$c_C(u) = \frac{1}{\sigma(u)} \tag{1}$$

where $\sigma(u)$ denotes the sum of the distances from a vertex $u \in V$ to all the other vertices v in a graph $G = (V, E)$. For example, vertex 1 of the graph shown in Fig. 9 has the highest closeness centrality.

Eccentricity Centrality Hage and Harary defined the eccentricity $e(u)$ of a vertex u as the maximum distance

from u to a vertex v in the graph (i.e., $e(u) = \max\{d(u, v) : v \in V\}$). More formally, *eccentricity centrality* of a vertex u is defined as follows:

$$c_E(u) = \frac{1}{e(u)} = \frac{1}{\max\{d(u,v) : v \in V\}} \quad (2)$$

Stress Centrality The definition of stress centrality is based on the set of shortest paths in a graph. The formal definition of stress centrality of a vertex u is as follows:

$$c_S(u) = \sum_{s \neq u \in V} \sum_{t \neq u \in V} \sigma_{st}(u) \quad (3)$$

where $\sigma_{st}(u)$ denotes the number of shortest paths between s and t containing u. For example, vertex 1 of the graph in Fig. 9 has the highest stress centrality.

Shortest Path Betweenness Centrality Let $\delta_{st}(u)$ denote the fraction of shortest paths between s and t that contain vertex u; i.e.,

$$\delta_{st}(u) = \frac{\sigma_{st}(u)}{\sigma_{st}} \quad (4)$$

where σ_{st} denotes the total number of shortest paths between s and t.

More formally, the *shortest path betweenness centrality* of a vertex u is defined as follows:

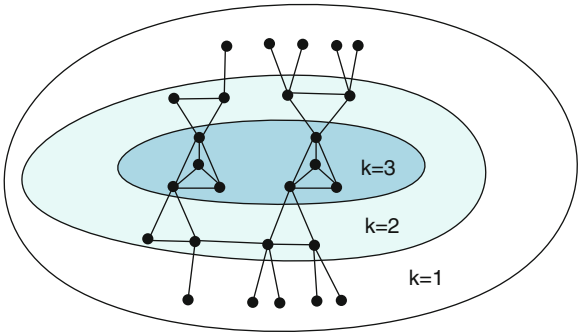
$$c_B(u) = \sum_{s \neq u \in V} \sum_{t \neq u \in V} \delta_{st}(u). \quad (5)$$

For example, the vertex 1 of a graph in Fig. 9 has highest shortest path betweenness centrality.

Group-level Analysis

Clique Analysis A *clique* is a dense cohesive subgroup in a network. Formally, a clique of an undirected graph $G = (V, E)$ is a subset U of V, such that every pair of vertices $u, v \in U$ are adjacent (i.e., the *induced subgraph* $G[U]$ is a complete graph). A *maximum clique* of a graph G is a clique with the largest size.

However, testing whether a given graph G has a clique of size k is an *NP-complete* problem, which is difficult to solve. For details, see (Brandes and Erlebach 2005).



Graph Algorithms in Network Analysis, Fig. 10 Example of k-core analysis

***k*-Core Analysis** Another way to identify dense subgraphs in a graph is k-core analysis. The *k-core* of a graph G is a maximal connected subgraph of G, in which all the vertices have degree at least k. Equivalently, a k-core is a connected component of the subgraph of G, formed by repeatedly deleting all vertices of degree less than k. The k-core of a graph G can be computed efficiently in linear time by repeatedly removing the vertex with the smallest degree.

Note that a k-core is a subgraph of a $(k-1)$-core. For example, in Fig. 10, the dark gray region indicates the 3-core of the graph, and the light gray region shows the 2-core of the graph.

Structural Equivalence and Blockmodel Two vertices u and v are *structurally equivalent*, if they are connected to exactly the same vertices in a graph G (intuitively, they hold identical positions in the network). More formally, two vertices u and v are *structurally equivalent* if, for each edge (u, x) in G, there is an edge (v, x) in G, and for each edge (v, x) in G, there is an edge (u, x) in G.

We can partition the vertex set V of G into *structural equivalence classes* or *blocks* such that if u and v are structurally equivalent, then they belong to the same equivalence class X. For algorithms for computing structural equivalence, see (Brandes and Erlebach 2005).

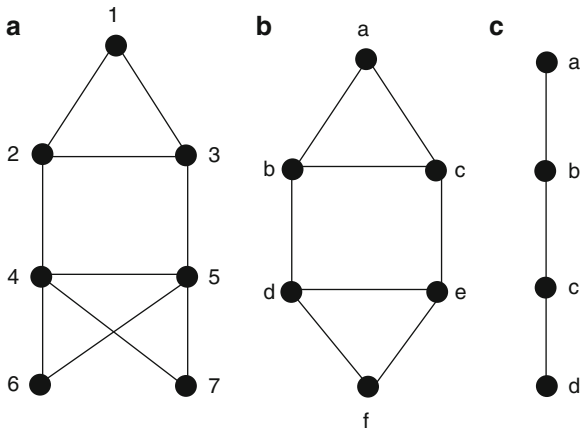
Given a structural equivalence relation, if a vertex u in block X is connected to a vertex w in block Y, then every vertex in block X is connected to every vertex in block Y. Then, by defining an edge (x, y) between the blocks X and Y, we can construct a *reduced graph* or *blockmodel* of the network.

Intuitively, blockmodeling represents the relationships between social positions. For example, Fig. 11b

shows a blockmodel of the graph in Fig. 11a, defined according to structural equivalence. Block a consists of vertex 1, block b consists of vertex 2, block e consists of vertex 5, and block f consists of the vertex set {6, 7}.

There are two relaxations of structural equivalence: *automorphic equivalence* and *regular equivalence*. Informally, automorphically equivalent vertices have the same position in a network in a more abstract sense than structurally equivalent vertices: They are not connected to the exactly same vertices, but to vertices that play analogous roles in the network.

To make this idea more formal, we need the concept of an automorphism. A permutation α of V is an *automorphism* of the graph $G = (V, E)$ if, for each edge (u, x) in G, $(\alpha(u), \alpha(x))$ is an edge in G, and vice versa.



Graph Algorithms in Network Analysis, Fig. 11 Examples of blockmodels of a graph in (**a**), based on (**b**) structural equivalence, and (**c**) automorphic equivalence

Vertices u and v are automorphically equivalent if $v = \alpha(u)$ for some automorphism α. For example, Fig. 11c shows a blockmodel of the graph in Fig. 11a as defined by automorphic equivalence. Block a consists of vertex 1, block b consists of the vertex set {2, 3}, block c consists of the vertex set {4, 5}, and block d consists of the vertex set {6, 7}.

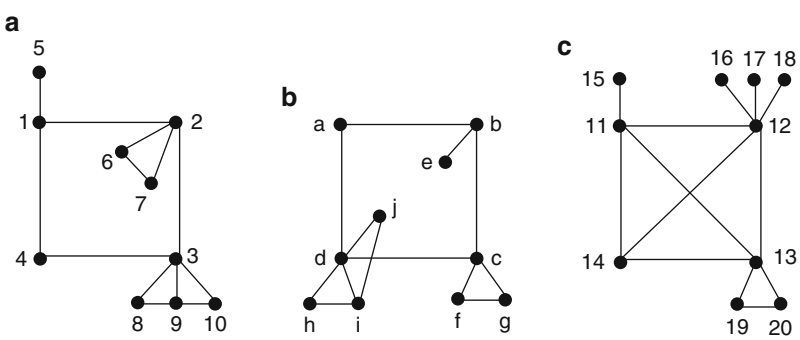
Computing the automorphisms of a given graph is a computationally hard problem known as *isomorphism complete*. However, for special classes of graphs such as trees and planar graphs, an automorphism can be computed efficiently in linear time. For details, see (Brandes and Erlebach 2005).

Network-level Analysis

Graph Isomorphism An *isomorphism* between two graphs is a mapping between the vertex sets of the two graphs, which preserves adjacency. An isomorphism of graphs G and H is a bijection f: $V(G) \rightarrow V(H)$ between the vertex sets of G and H, such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H.

For example, Fig. 12a and b are isomorphic graphs, since we can define a mapping as follows: {(4, a), (1, b), (5, e), (2, c), (6, f), (7, g), (3, d), (8, h), (9, i), (10, j)}. However, Fig. 12a and c are not isomorphic graphs, since there is no mapping between the two vertex sets that preserves adjacency.

Testing whether two graphs are isomorphic is a computationally hard problem. The time complexity of the problem is not known for general graphs. However, the problem can be solved efficiently in linear time, for special classes of graphs such as trees and planar graphs. For details, see (Brandes and Erlebach 2005).



Graph Algorithms in Network Analysis, Fig. 12 Example of isomorphic graphs: the graphs in (**a**) and (**b**) are isomorphic graphs, (**c**) nonisomorphic graph

References

Bondy JA, Murty USR, Bondy JA, Murty USR (1976) Graph theory with applications. North Holland, New York
Brandes U, Erlebach T (2005) Network analysis: methodological foundations. Springer, Berlin
Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to algorithms, 2nd edn. MIT Press/McGraw-Hill, Cambridge, MA/New York
Goodrich MT, Tamassia R (2001) Algorithm design. Wiley, Chichester
Junker BH, Schreiber F (2008) Analysis of biological networks. Wiley, Hoboken
Wasserman S, Faust K (1994) Social network analysis: methods and applications. Cambridge University Press, New York
West D (2001) Introduction to graph theory. Prentice Hall, Upper Saddle River

Graph Alignment

▶ Scoring Function, Graph Alignment

Graph Alignment, Protein Interaction Networks

Michal Kolář
Institute of Molecular Genetics, Academy of Sciences of the Czech Republic, Prague, Czech Republic

Synonyms

Alignment, protein interaction networks; Network alignment, protein interaction networks

Definition

Graph alignment of protein–protein interaction networks (▶ Protein-Protein Interaction Networks) is a method of comparison of protein interaction data between two or more species and is one of the methods of ▶ comparative analysis of molecular networks. The method represents the interaction data in a form of a graph (▶ Graph, ▶ Protein-Protein Interaction Networks, ▶ Interactome) and constructs a mapping between the nodes of the graph (proteins) and the corresponding links (protein–protein interactions). Similarly to sequence alignment, it provides means for functional and phylogenetic comparison of the proteins.

Characteristics

Graph Alignment Structure

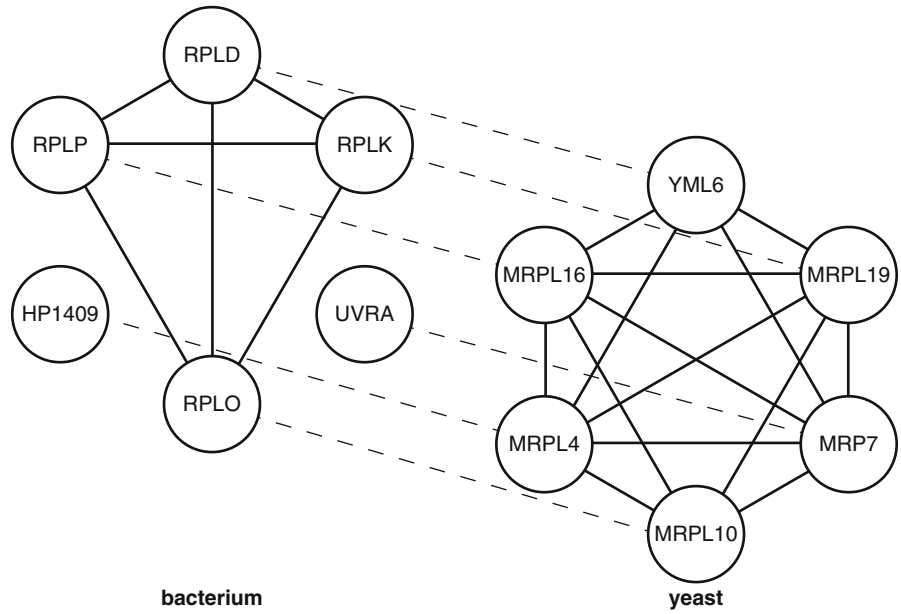
A graph alignment (▶ Network Alignment) compares protein–protein interaction networks (Protein–Protein Interaction Network) (PIN) and groups their proteins into equivalence (analogy) classes. In the case of a pair-wise comparison, the equivalence is conveniently represented by a pairing of the analogous proteins thus creating a map between the nodes (proteins) of the two networks. The alignment of the links (protein–protein interaction) results from the aligned nodes.

Figure 1 represents a pair-wise graph alignment between two PINs representing a part of the ribosomal complex in two distinct species. Proteins are represented as the nodes of the network and the protein–protein interactions as the links. The alignment is shown by dashed lines, which interconnect the proteins in the same equivalence class. The alignment of nodes forces the alignment of links. Several modalities exist for the aligned links. The links may be either present in both species resulting from matching protein–protein interactions or they may be absent in one or both species.

In practice, the alignment of the two networks is represented as in Fig. 2. The proteins belonging to the same equivalence class are overlaid. The line type indicates presence or absence of the protein–protein interactions in individual PINs and thus the modality of the link. An alignment of multiple networks is represented similarly, for example, see Fig. 3. There, the nodes represent proteins of the same equivalence class (e.g., MRPL19, RPLK, and RLX1). Some equivalence classes may be absent in one or more of the networks (e.g., MRPL4 and HP1409 do not have an analogous protein in fish).

A pair-wise graph alignment of networks $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ is formally defined as a mapping A from the vertex (node) set V_1 to the vertex set V_2; $A: i \in V_1 \rightarrow i' \in V_2$. An edge (link) $(i, j) \in E_1$ is told to be aligned to an edge $(i', j') \in E_2$ if and only if $A(i) = i'$ and $A(j) = j'$.

The type of the mapping A distinguishes whether the alignment is considered global or local. The global graph alignment (▶ Global Network Alignment) is defined by a (total) injective mapping A. Thus, each node in the smaller network is aligned to some node in



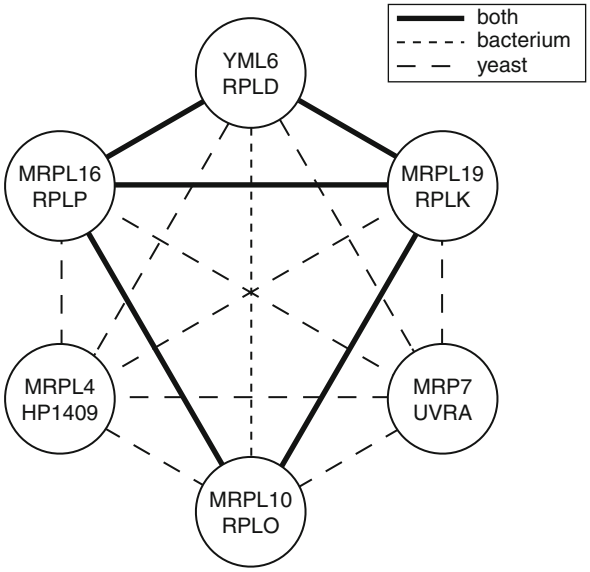
Graph Alignment, Protein Interaction Networks, Fig. 1 A small part of protein–protein interaction networks (Protein–Protein Interaction Network) of a bacterium (*Helicobacter pylori*) and yeast (*Saccharomyces cerevisiae*) representing protein components of the ribosome. Each node represents a protein and is labeled by its symbol. The links stand for protein–protein interactions. The bacterial network is sparse with only several interactions described in the databases. The yeast network on the other side forms an almost complete clique. The graph alignment, which is denoted by dashed lines, may be used to predict physical interactions among the bacterial genes. The figure is based on Sharan et al. (2005) and updated using STRING db version 8.3 (http://string-db.org)

the larger network. No two nodes from the smaller network may be aligned to the same node in the larger network. The local graph alignment (▶ Local Network Alignment) is defined by a partial injective mapping A. Thus, only elements of a subset of the node set V_1 are aligned to some nodes in V_2. Still, no two nodes from one network may be aligned to the same node in the other network. In some definitions of the graph alignment, the injective property of the mapping is lifted, allowing for alignment of two or more nodes of one network to the same node in the other network. This option allows for the representation of, for example, protein duplications.

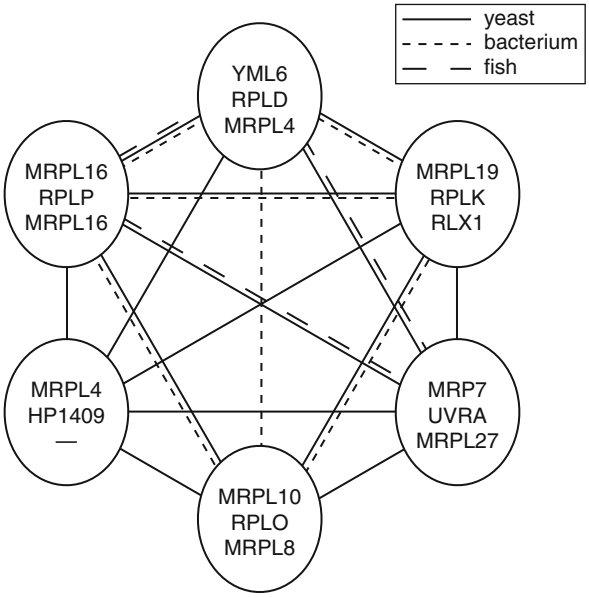
A multiple graph alignment (▶ Multiple Network Alignment) of N graphs $G_i(V_i,E_i)$, $i = 1, \ldots, N$ is an equivalence relation A over the nodes $V = V_1 \cup \ldots \cup V_N$. An equivalence relation is transitive and partitions V into a set of disjoint equivalence classes. A local alignment is a relation over a subset of the nodes in V; a global alignment is a relation over all nodes in V. Figure 3 shows an example of an alignment of three protein–protein interaction networks. Nodes (proteins) in the same equivalence class are considered functionally analogous.

Score of a Graph Alignment

The alignment is scored by interaction similarity and protein similarity. Each equivalence class of aligned proteins (or a pair of aligned proteins in case of a pair-wise graph alignment) contributes to a node score (▶ Node Score, Graph Alignment) S_n, which rewards similarity of the aligned proteins (e.g., level of their homology, say a BLAST bit score) and penalizes similarity among proteins not respected by the graph alignment and its equivalence classes. Aligned protein pairs contribute a positive link score (▶ Link Score, Graph Alignment) S_l if the interaction between the proteins is conserved in all or some networks. Thus, the interaction between equivalence classes (MRPL16, RPLP, MRPL16) and (YML6, RPLD, MRPL4) in Fig. 3 would contribute a positive link score (Link Score, Graph Alignment), as the protein–protein



Graph Alignment, Protein Interaction Networks, Fig. 2 A more concise representation of the alignment in Fig. 1 is created by overlaying the aligned nodes and coding the modality of the links by line types. Here, *full strong* lines stand for the interactions present in both species, *dashed lines* for the interactions present in yeast only, and *dotted lines* for the interactions present in the bacterium only



Graph Alignment, Protein Interaction Networks, Fig. 3 Illustration of an alignment of multiple networks. In addition to the bacterium and yeast, the protein–protein interaction network of fish (*Oryzias latipes*) also has been aligned. The nodes correspond to the equivalence classes and the protein symbols in the three species are given (from the *top*: yeast, bacterium, and fish). The protein–protein interactions present in each of the networks are given in different line type. The equivalence class consisting of MRPLl4 in yeast and HP1409 in the bacterium is not represented in fish

interaction is present in all three networks. On the other side, the interactions present in a single network only would be penalized by the link score (Link Score, Graph Alignment) (e.g., interaction between the classes (MRPL10, RPLO, MRPL8) and (MRP7, UVRA, MRPL27) in Fig. 3). The node score (Node Score, Graph Alignment) and the link score (Link Score, Graph Alignment) form the full scoring function (▶ Scoring Function, Graph Alignment) of the graph alignment $S = S_n + S_l$.

Construction of a Graph Alignment

The graph alignment problem stands in finding the highest scoring graph alignment among all possible alignments of the protein–protein interaction networks. As opposed to the sequence alignment, finding an optimal graph alignment is a computationally hard problem. The underlying subgraph isomorphism problem, which asks if one ▶ graph exists as an exact subgraph of the other graph, is ▶ NP-hard. This means that no exact and efficient algorithm can be found. All current methods use some heuristic algorithms (▶ Heuristic Optimization) to find the best graph alignment with a notable exception of Natalie (https://www.mi.fu-berlin.de/w/LiSA/Natalie), which employs Lagrangian relaxation.

Graph alignment heuristics have been proposed based on three main ideas: The first kind of algorithm forces alignment of orthologous proteins (▶ Orthologs) in the PINs. Thus, these aligners use only the information on the nodes of the network. This approach allows to identify ancestral networks, network parts enriched in conserved edges, or to decide between paralogous genes.

The second approach utilizes only the information on the interaction patterns of the proteins. It allows to discover common regulatory motives in PINs and to study phylogeny. Similarity of the aligned proteins is not required in this approach as common topological structures are searched for.

The third strategy relies both on aligning homologous proteins and on aligning topologically analogous subnetworks. This is the most complete approach, which allows direct evolutionary comparison of the graphs. Several algorithms have been proposed. For excellent reviews of particular algorithms and their

applications, see references Sharan and Ideker (2006), Chen et al. (2009), Stumpf and Wiuf (2009), Cannataro et al. (2010), and Pržulj (2011).

Purposes of a Graph Alignment

Similarly to sequence alignment the graph alignment provides a broad applicability. The local graph alignment has been used for detection of conserved network motives (▶ Network Motif) or pathways (▶ Pathway) among different species, for detection of paralogous pathways in a single species or for a database search in which a small query pathway is searched within a large protein–protein interaction network of a target organism.

Graph alignment may further help in decision on protein orthology in cases where sequence homology is not conclusive enough (e.g., when proteins have a weak sequence homology only or when several ▶ paralogs exist). The global alignment immediately suggests functional orthologs across species. In addition, the method of graph alignment allows detection of a functional replacement of a protein by an unrelated nonhomologous protein (non-orthologous gene displacement). The alignment may also be used to predict the function of a protein based on the functional annotation of other proteins in the same equivalence class.

Global graph alignment may be used to recover species phylogeny. Topology-only-based alignments have the potential to provide a completely new, independent source of phylogenetic information.

While comparison of the PINs has been given the major focus in the recent years, mainly because of good quality and availability of the data, many applications emerged also in comparison of other Biological Network Models, including protein contact maps (▶ Protein Contact Maps), ▶ metabolic and signaling networks, gene ▶ co-expression networks, and ▶ gene regulatory networks (▶ Comparative Analysis of Molecular Networks).

Strengths and Weaknesses

As the graph alignment employs for detection of analogy of proteins two pieces of biological information (▶ Information, Biological) – similarity of the aligned proteins and similarity of the topology of their interactions – it is much stronger than sequence alignment and it may detect orthologous proteins also in case of a weak sequence similarity. However, the complexity of the data leads also to its weakness: the computational cost of the algorithms for finding of the optimal graph alignment is large.

Algorithms and Tools

There is a large variety of graph alignment algorithms and tools; tools with academic licensing include:

- GRAAL (http://bio-nets.doc.ic.ac.uk/GRAAL_suppl_inf)
- Græmlin (http://graemlin.stanford.edu)
- GraphAlignment (http://bioconductor.org/packages/bioc/html/GraphAlignment.html)
- IsoRank (http://groups.csail.mit.edu/cb/mna)
- Natalie (https://www.mi.fu-berlin.de/w/LiSA/Natalie)
- PathBlast (http://pathblast.org)

Cross-References

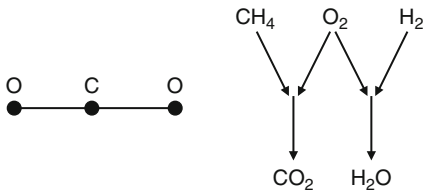
▶ Co-expression
▶ Comparative Analysis of Molecular Networks
▶ Gene Regulatory Networks
▶ Global Network Alignment
▶ Graph
▶ Heuristic Optimization
▶ Information, Biological
▶ Interactome
▶ Link Score, Graph Alignment
▶ Local Network Alignment
▶ Metabolic and Signaling Networks
▶ Multiple Network Alignment
▶ Network Alignment
▶ Network Motif
▶ Node Score, Graph Alignment
▶ Orthologs
▶ Paralogs
▶ Pathway
▶ Protein Contact Maps
▶ Protein-Protein Interaction Networks
▶ Scoring Function, Graph Alignment

References

Cannataro M, Guzzi PH, Veltri P (2010) Protein-to-protein interactions: technologies, databases, and algorithms. ACM Comput Surv 43(1):1–36

Chen L, Wang RS, Zhang XS (2009) Biomolecular networks: methods and applications in systems biology. Wiley, New Jersey

Pržulj N (2011) Protein-protein interactions: making sense of networks via graph-theoretic modeling. Bioessays 33(2):115–123
Sharan R, Ideker T (2006) Modeling cellular machinery through biological network comparison. Nat Biotech 24(4):427–433
Sharan R, Ideker T, Kelley B, Shamir R, Karp RM (2005) Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. J Comp Biol 12(6):835–846
Stumpf MPH, Wiuf C (2009) Statistical and evolutionary analysis of biological networks. Imperial College Press, London



Graph Mining, Fig. 1 A molecule (carbondioxide) represented as a graph (*left*) and a chemical interaction network depicting the oxidation of methane and hydrogen (*right*)

Graph Clustering

► Modules in Networks, Algorithms and Methods
► Network Clustering

Graph Mining

Jan Ramon
Declarative Languages and Artificial Intelligence Group, Katholieke Universiteit Leuven, Leuven, Belgium

Synonyms

Learning from graph structured data; Network analysis

Definition

Graph mining is the study of how to perform ► data mining and machine learning (► Identification of Gene Regulatory Networks, Machine Learning) on data represented with graphs. One can distinguish between, on the one hand, transactional graph mining, where a database of separate, independent graphs is considered (such as databases of molecules and databases of images), and, on the other hand, large network analysis, where a single large network is considered (such as chemical interaction networks and concept networks).

Characteristics

Graph-Structured Data

In many applications, it is natural to represent data with ► graphs. One can distinguish two main settings. First, in the transactional graph mining setting, databases of separate, independent graphs are considered. For example, in a molecule database, molecules are commonly represented using one vertex for every atom and one edge for every bond between two atoms. Large, publicly available databases of chemical compounds include the NCI dataset (http://cactus.nci.nih.gov/) and the ZINC dataset (http://zinc.docking.org/).

Second, in the single (large) network setting, all data is represented in one large, connected network. Examples of such networks include the Internet, social networks, citation networks, concept networks, computer networks, chemical interaction networks, gene regulatory networks, socioeconomic networks, and encyclopedias. Sample datasets are publicly available at, among others, http://snap.stanford.edu/data/. In a chemical interaction network, molecules are represented by vertices connected by chemical reactions. The level of detail and the exact representation may be different among datasets. For example, chemical reactions may be represented as separate nodes in the network with arcs from/to the participating compounds, or they may be implicit, in which case compounds which are involved in the same chemical reaction are just connected with an undirected edge. Next to networks of chemical compounds, it is also common to consider higher-level networks such as protein interaction networks and gene regulatory networks. For example, in gene regulatory networks nodes represent genes and arcs between nodes indicate that one gene codes for a transcription factor regulating the other gene. In comparison to the transactional setting, an important challenge in the single network setting is that one's beliefs on all data may be dependent on one another. Most traditional machine learning techniques assume that examples are drawn identically and independently (i.i.d.) (Fig. 1).

Other abstractions are sometimes preferred to graphs in order to represent similar data, such as relational databases and logic. The domains focusing on data mining using these representations are called relational data mining and inductive logic programming, respectively. Representing data with graphs has several advantages. First, the representation language is simple and therefore, allows for the fast development of algorithms. Second, the representation language is expressive and adequate for the majority of applications. Finally, there is a vast literature on efficient graph algorithms. A potential disadvantage, especially in order to use algorithms implemented only for simpler graph representation, is that it may be necessary to transform the data into a simpler (but equally expressive) graph format in a preprocessing step.

Transactional Graph Mining Methods

Graph mining methods cover the whole range of methods from data mining and machine learning. We only list here a few examples of methods which received significant attention in the literature.

Graph Pattern Mining

Graph pattern mining methods perform ▶ pattern mining on graph-structured data, i.e., they list all patterns which satisfy some interestingness criterium such as being frequent. A frequent pattern is a pattern which is a subgraph of at least a certain fraction of the transaction graphs in the database. Well-known graph mining systems are gSpan (Yan and Han 2002) and Gaston (Nijssen and Kok 2004).

A popular strategy for the application of these systems and related ones to quantitative structure–property relationship (QSPR) modeling (i.e., the modeling of the relationship between the structure of molecules and their chemical properties) is to first generate frequent molecular fragments, then to generate one boolean feature per pattern (with value 1 for molecules having the pattern as substructure and with value 0 for other molecules), and then to apply some suitable ▶ classification algorithm (such as a support vector machine (▶ Biomedical Decision Support Systems)) on these features.

Comparing Graphs

In order to compare small graphs, such as molecular graphs, one can use graph kernels, graph metrics, and maximum common subgraph operators. Kernels on molecular graphs such as presented in (De Grave and Costa 2010) can be used with any kernel-based learning (▶ Learning, Kernel-based) method such as support vector machines and Gaussian processes. Metrics and maximum common subgraph operators can be used in instance-based learning approaches, or as features for a wide range of classification algorithms (Schietgat et al. 2010).

Methods for Analyzing Large Networks

Analyzing Overall Network Regularity

An important starting point for many methods for analyzing large networks is the observation that large real-world networks, independently of the domain, satisfy a number of statistical regularities. For example, many networks satisfy the small world model, which informally corresponds to the fact that the number of highly connected nodes is much smaller than the number of low degree nodes. Also, many networks can be clustered in modules of nodes which are much better connected to each other than to nodes in other modules. As a consequence, much inspiration has come from random graph theory (Bollobás 2001; Durrett 2007) and spectral graph theory (Chung 1997), which study the statistical properties of such graphs. Alon (2007) discusses motifs in biological networks and the surprising deviation of frequencies of certain motifs from what one would expect if the given network were completely random.

Predicting Node Properties

Often however, in addition to network-level regularities, also a more detailed node-by-node analysis of a network is necessary. Several approaches aim at modeling properties of nodes in a network. First, in the field of statistical relational learning (Getoor and Taskar 2007), probabilistic models are being studied which allow to reason about beliefs of the properties of individual nodes and their connections in a Bayesian network manner. Second, semi-supervised learning (Zhu and Goldberg 2009) aims at learning predictive models exploiting not only the information about the training examples but also the information about the unlabeled examples. This is especially useful in networks where nodes and their connections are known, but not the value of some target attribute.

Cross-References

- ▶ Biomedical Decision Support Systems
- ▶ Classification
- ▶ Data Mining
- ▶ Learning, Kernel-Based
- ▶ Learning, Relational
- ▶ Learning, Supervised
- ▶ Learning, Unsupervised
- ▶ Pattern Mining

References

Alon U (2007) An introduction to systems biology. Chapman & Hall/CRC, Boca Raton
Bollobás B (2001) Random graphs. Cambridge University Press, Cambridge
Chung F (1997) Spectral graph theory. AMS, Providence
De Grave K, Costa F (2010) Molecular graph augmentation with rings and functional groups. J Chem Inf Model 50(9):1660–1668
Durrett R (2007) Random graph dynamics. Cambridge University Press, Cambridge
Getoor L, Taskar B (2007) An introduction to statistical relational learning. MIT Press, Cambridge, MA
Nijssen S, Kok J (2004) A quickstart in frequent structure mining can make a difference. In: Proceedings of the 10th ACM SIGKDD international conference, pp 647–652
Schietgat L, Costa F, Ramon J, De Raedt L (2010) Effective feature construction by maximum common subgraph sampling. Machine Learning 83(2):137–161
Yan X, Han J (2002) gSpan: graph-based substructure pattern mining. In: Proceedings of the 2002 international conference on data mining (ICDM'02), pp 721–724
Zhu X, Goldberg AB (2009) Introduction to semi-supervised learning. Synth Lect Artif Intell Machine Learn 3:1–130

Graph Partitioning

- ▶ Modules, Identification Methods and Biological Function
- ▶ Network Clustering

Graph Study of Metabolic Networks

- ▶ Topology of Metabolic Reaction Networks

Graphical Gaussian Model

Zhong-Yuan Zhang
School of Statistics, Central University of Finance and Economics, Beijing, China

Synonyms

Concentration graph model; Covariance selection model

Definition

Graphical Gaussian model (CGM) (Crzegorxczyk et al. 2008; Hache et al. 2009; Werhli et al. 2006) is an undirected graph whose nodes are genes and two genes are linked by an edge if there is an interaction between them. The interactions are measured by the partial correlation coefficients conditioned on all the other genes. After the correlation coefficients are determined, a statistical significance test is employed, and the two genes whose score is significantly large are considered to have an interaction. Otherwise they are considered to be conditional independence and there is no edge between them. Under the assumption that the data are distributed according to a multivariate Gaussian distribution, the partial correlation coefficient of gene x and gene y is given as:

$$score(x, y) = -\frac{C^{-1}_{xy}}{\sqrt{C^{-1}_{xx} C^{-1}_{yy}}}$$

where C^{-1}_{xy} is the element of C^{-1}, inverse of the covariance matrix C of the data.

Cross-References

- ▶ Identification of Gene Regulatory Networks, Machine Learning

References

Crzegorxczyk M, Husmeier D, Werhli AV (2008) Reverse engineering gene regulatory networks with various machine learning methods. In: Emmert-Streib F, Dehmer M (eds) Analysis of microarray data: a network-based approach. Wiley GmbH, Weinheim, KGaA

Hache H, Lehrach H, Herwig R (2009) Reverse engineering of gene regulatory networks: a comparative study. EURASIP J Bioinform Syst Biol 2009:617281

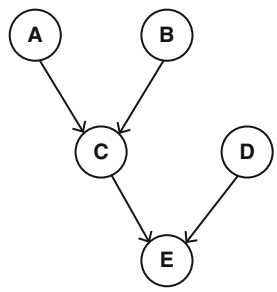
Werhli AV, Grzegorczyk M, Husmeier D (2006) Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks. Bioinformatics 22(20):2523–2531

Graphical Model

Xiujun Zhang
Institute of Systems Biology, Shanghai University, Shanghai, China

Definition

A graphical model is a way of representing probabilistic relationships between random variables. In a graphical model, variables are represented by nodes, conditional independencies (independencies) are represented by (missing) edges (Koller and Friedman 2009). Figure 1 gives a simple graphical model in which the nodes 'A–E' indicate variables and the arrows indicate dependencies between variables. In this graphical model, 'C' depends on 'A' and 'B', 'E' depends on 'C' and 'D'. The two most common forms of graphical models are directed graphical models and undirected graphical models. One of the most popular directed graphical model is Bayesian network.



Graphical Model, Fig. 1 Simple graphical model

References

Koller D, Friedman N (2009) Probabilistic graphical models. MIT Press, Massachusetts. ISBN 0-262-01319-3

Graphical Representation of Biological Processes

▶ SBGN

Green Systems Biology

▶ Plant Systems Biology

Green's Theorem

▶ Partial Differential Equations, Poisson Equation

Grid Computing, Parallelization Techniques

Giuseppe Agapito
Department of Experimental Medicine and Clinic, University Magna Graecia of Catanzaro, Catanzaro, Italy

Synonyms

Data parallel; Distributed data access; Distributed data management; Integration technologies; Parallel computing

Definition

A Computational Grid is a collection of heterogeneous computers and resources spread across the network making a confederation of multiple administrative domains with the intent to provide users uniform access to these resources to reach a common goal. To provide an

efficient use of resources of a Computational Grid, many protocols to access resources have been developed. Each access protocol has unique security requirements and implications for both the user and the resource provider. Grid Computing was developed to provide scalable access to wide area distributed resources.

Characteristics

Research in many areas of life sciences, such as genomics and proteomics, are particularly computationally intensive. This generated the need for a huge amount of computational resources for running algorithms of ever-increasing complexity over data of ever-increasing size. Grid provides the optimal solution to meeting many of the computational needs of research in life sciences as described in Krishnan (2004). One of the major challenges for the bioinformatics is to provide the tools needed to analyze the sequences provided by whole genome sequencing. The existent data are distributed in different domains (since data experiments are conducted in different laboratories and research centers), reason for which Grids represent the solution to develop applications able to speed up the analysis of the whole genome, in order to predict the function of a new gene, or identify important regions in genomic sequences.

The computational Grids can be viewed such as persistent environments that enable the realization of software able to integrate visualizations, computing and analysis resources belonging to different domains and geographically distributed.

In Grid, the sharing is not limited to data, but it is extended to applications and hardware, and it is possible to share the resources among different organizations spread in the network. The access to resources is regulated by different policies enabling the use to only authorized users. Furthermore, Grid allows the users to access and use resources and services spread across the network in a transparent way, without requiring that they are aware of the physical locations of the resource.

In fact, supposing that a job submitted on a node of the Grid crashes due to some reasons, the Grid automatically resubmits the job on another available node. This is an advantage for the users that only have to submit their service requests at the Grid, which then automatically locates the available computing resources to serve the requests.

Grid Computing is a special category of parallel computing because it relies on a network of heterogeneous computers spread across the network that is in contrast to the traditional notion of a supercomputer, which has many processors connected by local high-speed connectors. The major advantage from the Grid Computing is the easiness and cheapness whereby it is possible to increase the computational power. The increasing of the computational power is done by combining more resources such as traditional computers, clusters, and different kind of resources, thus obtaining a computational power similar to that of a multiprocessor supercomputer but at a lower cost.

Developing Grid applications presents significant challenges, considering the high heterogeneity of resources and the dynamic behavior of Grid environments. Grid applications need to be designed to exploit the heterogeneous capacities of the available resources and overcome the problems related to fluctuations in both performance and availability of these shared resources. In such context, it seems that writing efficient and stable programs for the Grid is more difficult than write applications for traditional parallel machine. But nevertheless, it is possible to use classical programming techniques based on the exchange of messages to develop Grid applications. The most common techniques are developed starting from the Message Passing Interface (MPI). For scientific applications, the MPI specification is the most widely used, and there exist several versions of MPI optimized for the Grid. The efficient and reliable execution of real scientific applications is the reason for which applications for the Grid are developed using MPI (Nascimento et al. 2007). The Message Passing Interface Standard is a message passing library, and its goal is to establish a portable, efficient, and flexible standard for writing message passing programs.

To simplify the development of Grid-based applications, the GridMPI and MPICH-G2 standards were developed.

GridMPI is an implementation of MPI designed to obtain high-performance computing in the Grid. GridMPI can establish multiple connections among geographically distributed computers in order to obtain a high computational power in an easy way.

MPICH-G2 is a Grid-enabled implementation of the MPI standard that allows the user to couple multiple machines, potentially of different architectures, to run MPI applications.

Cross-References

- ▶ Cores
- ▶ GridMPI
- ▶ Message Passing Interface (MPI)
- ▶ MPICH-G2
- ▶ Parallel Computing

References

Krishnan A (2004) A survey of life sciences applications on the grid. New Gener Comp 2:111–125

Nascimento AP, Sena AC, Boeres C, Rebello VEF (2007) Distributed and dynamic self-scheduling of parallel MPI grid applications. Concurr Computat Pract Exper 19:1955–1974

Grid Computing, Parameter Estimation for Ordinary Differential Equations

Ivan Merelli, Ettore Mosca and Luciano Milanesi
Institute for Biomedical Technologies – CNR (Consiglio Nazionale delle Ricerche), Segrate, Milan, Italy

Synonyms

Fitting of continuous and deterministic models; High throughput computing

Definition

The parameter estimation for ordinary differential equations with grid computing refers to the exploitation of a combination of computer resources, usually characterized by loose coupling, heterogeneity, and geographical dispersion, to carry out the calculations required to identify a particular set of values for the parameters included in continuous and deterministic models.

Characteristics

Systems biology kinetic models contain parameters that usually describe physical and chemical properties of macromolecules and biological processes represented by the models. Just to mention a few cases, these parameters can be the constants appearing in the Michaelis-Menten kinetics (Nelson and Cox 2005), the association constant for two proteins interacting to produce a protein complex or the diffusion coefficient of a protein between two compartments (e.g., between the cytoplasm and nucleus).

Due to the lack of experimental measurements, experimental errors, and biological variability, the value of many of these parameters is yet unknown or uncertain (Gunawardena 2010). Since variations of the parameter values can dramatically affect the system's trajectory over the phase space, the selection of a "proper" set of parameter values is a crucial task for the usability of a model as a tool for the in silico investigation of the properties of the modeled system.

The Parameter Estimation Problem

The parameter estimation problem of nonlinear dynamical systems is stated as the minimization of a cost function $J(\mathbf{Y}, \mathbf{Y}^*)$ that measures the goodness of the model output $\mathbf{Y}^*$ with respect to a given data set $\mathbf{Y}$ (experimental data), subject to constraints that are the model itself $f(\mathrm{d}\mathbf{x}/\mathrm{d}t, \mathbf{x}, \mathbf{p}, t) = 0$ (where $\mathbf{x}$ are the variables and $\mathbf{p}$ parameters), the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$, other possible algebraic equalities $\mathbf{g}(\mathbf{x}, \mathbf{p}) = 0$ and inequalities $\mathbf{h}(\mathbf{x}, \mathbf{p}) \leq 0$, and the bounds over the parameter values $\mathbf{p}^{\mathrm{L}} \leq \mathbf{p} \leq \mathbf{p}^{\mathrm{U}}$ (Moles et al. 2003). Thus, it is mathematically defined as a nonlinear programming problem with differential-algebraic constraints, shortly a NLP-DAE problem.

Global Optimization

The nonlinearity and the constrained nature of the system dynamics make usually these problems non-convex. This means that the NLP-DAE solution must be searched with a global optimization (GO) method, since it is very likely that a local method would identify a solution of local nature. GO strategies can be classified in two broad classes, deterministic and stochastic: generally speaking, deterministic approaches can assure a higher degree of assurance that the global optimum will be reached, but no algorithm can guarantee the identification of the global optimum with certainty in finite time and, moreover, the computational cost increases very quickly with the problem size; stochastic methods have weak theoretical guarantees of convergence to global optimality, but can

locate the vicinity of it in modest computation time, and, moreover, are easy to implement and do not require a transformation of the original problem (Moles et al. 2003).

Evolution strategies (ES), a sub-class of nature-inspired stochastic optimization methods belonging to the class of evolutionary algorithms (Fogel 2006), are a good candidate to cope with NLP-DAE problems exploiting grid platform. In fact, they have shown good performance when applied to parameter estimation in biochemical pathways (Moles et al. 2003), and it is possible to implement them in a data parallel manner, which is the best solution to obtain very good performance with grid computing. More precisely, it is possible to run different instances of an ES algorithm simultaneously, swapping periodically the best results among the processes. Importantly, this approach speeds up the convergence to the optimal solution since a wider search over the solutions space is performed.

Distributed Implementation

To manage the distribution of each run of the ES algorithm on different computational resources and to enable the asynchronous communication among these runs, a software environment is required. A possible solution, described in Mosca et al. (2008), involves a relational database that stores the results achieved during each evolution process. More precisely, each run is performed on a computational resource, which can be completely independent of the others. The only requirement is the availability of a communication network to establish a connection to the database. Each run starts by contacting the main database to inform the system of its presence, then it randomly initializes the population and runs the optimization. After the accomplishment of each iteration, each process stores its solutions along with the corresponding cost function (that evaluates the quality of the solution). If the other processes are ready to swap their results, the algorithm downloads from the database a subset of the solutions (e.g., sorted by the fitness value) from another randomly selected process and adds them to its current set of results.

The coordination of the swap among the processes is delegated to a script which runs in close association with the database. This script queries the database to check if all processes are ready to exchange their individuals, which means that they have performed a minimum number of iterations from the beginning or after the previous swap. If this is the case, the script flags a specific field in the database that enables the exchange when each process has completed the current step.

Cross-References

- ▶ Convex Programming
- ▶ Dynamical Systems Theory, Asymptotics and Singular Perturbations
- ▶ Evolutionary Algorithms
- ▶ Global Optimum
- ▶ Grid Computing, Parallelization Techniques
- ▶ High-Throughput Computing, Asynchronous Communication
- ▶ Mathematical programming, Constraint
- ▶ Mathematics, Nonlinear Programming
- ▶ Ordinary Differential Equation (ODE)
- ▶ Parallel Computing, Data Parallelism

References

Fogel DB (2006) Evolutionary computation: toward a new philosophy of machine intelligence. IEEE Press, Piscataway

Gunawardena J (2010) Models in systems biology: the parameter problem and the meanings of robustness. In: Lodhi HM, Muggleton SH (eds) Elements of computational systems biology. Wiley, Hoboken, pp 21–47

Moles CG, Mendes P, Banga JR (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res 13(11):2467–2474

Mosca E, Merelli I, Alfieri R, Milanesi L (2008) A distributed approach for parameter estimation in systems biology models. Il Nuovo Cimento 32:165–168

Nelson DL, Cox MM (2005) Lehninger principles of biochemistry. WH Freeman, New York

GridMPI

Giuseppe Agapito
Department of Experimental Medicine and Clinic, University Magna Graecia of Catanzaro, Catanzaro, Italy

Definition

GridMPI is an open source project with the aim to provide an efficient and easy use of the MPI

(Message Passing Interface) standard in the Grid environments. GridMPI unlike from MPI comes with a collection of functions and services specific for the Grid environments that allow the user to overcome in an easy way problems typical of the Grid, related with the development of Grid applications. Typical examples regarding such problems often include a connection between different resources and organizations in order to improve the computation costs associated with scientific applications tipically performed by limiting the latency time, and managing and managing security problems automatically and in a transparent way.

Cross-References

▶ Grid Computing, Parallelization Techniques

Groove (G) Domain

Marie-Paule Lefranc
Laboratoire d'ImmunoGénétique Moléculaire, Institut de Génétique Humaine UPR 1142, Université Montpellier 2, Montpellier, France

Synonyms

G domain; G type domain; Groove domain

Definition

The Groove (G) domain is a type of structural unit (domain) that characterizes a protein chain belonging to the major histocompatibility (MH) superfamily (MhSF) (▶ MH Superfamily (MhSF)). The G domain comprises the G-DOMAIN of the MH and the G-LIKE-DOMAIN of the MhSF proteins other than MH (or related proteins of the immune system (RPI)-MH1Like).

A G domain (G-DOMAIN or G-LIKE-DOMAIN) is usually encoded by one exon of a gene. Two G domains participate to the characteristic groove structure that, in the MH, binds a peptide. Each G domain is very conserved and comprises four antiparallel beta strands (floor of the groove) and a helix (wall of the floor). In the MH class I (MH1) or in the RPI-MH1Like, the two G domains belong to the same chain (MH1-Alpha, or RPI-MH1Like-Alpha), whereas in the MH class II (MH2), the two domains belong to different chains, MH2-Alpha and MH2-Beta (Lefranc et al. 2005). The G-domain description per receptor type and chain type, based on the ▶ IMGT-ONTOLOGY concepts, is shown in Table 1. IMGT® labels are in capital letters.

Analysis of G-domain amino acid sequences can be performed by tools of IMGT®, the international ImMunoGeneTics information system® (http://www.imgt.org) (▶ IMGT® Information System).

IMGT/DomainGapAlign tool (Ehrenmann et al. 2010) aligns the user amino acid sequences with the closest G domains of the IMGT domain reference directory, creates gaps according to the ▶ IMGT unique numbering for G domain (Lefranc et al. 2005), delimits the strands, loops and helix, highlights differences with the closest reference(s), and generates the IMGT Colliers de Perles (▶ IMGT Collier de Perles).

For MH proteins with known three-dimensional (3D) structures, IMGT Colliers de Perles are used for comparison of pMH contact analysis (Kaas et al. 2008). Contact analysis between V-ALPHA and V-BETA (▶ Variable (V) Domain of the T cell receptor TR-Alpha, and

Groove (G) Domain, Table 1 G-domain description per receptor type and chain type

G domain	Receptor type	G-domain description per chain type	
G-DOMAIN	MH class I (MH1)	G-ALPHA1	On the same chain
		G-ALPHA2	
	MH class II (MH2)	G-ALPHA	
		G-BETA	
G-LIKE-DOMAIN	MhSF other than MH (RPI-MH1Like)	G-ALPHA1-LIKE	On the same chain
		G-ALPHA2-LIKE	

TR-Beta chains, respectively) and G-ALPHA1 and G-ALPHA2 (G domains of the MH1-Alpha chain) or G-ALPHA and G-BETA (G domains of the MH2-Alpha and MH2-Beta chains, respectively) are available in the 3D database (IMGT/3Dstructure-DB) of the ► IMGT® information system (Ehrenmann et al. 2010).

Cross-References

► IMGT Collier de Perles
► IMGT Unique Numbering
► IMGT® Information System
► IMGT-ONTOLOGY
► MH Superfamily (MhSF)
► Variable (V) Domain

References

Ehrenmann F, Kaas Q, Lefranc M-P (2010) IMGT/3Dstructure-DB and IMGT/DomainGapAlign: a database and a tool for immunoglobulins or antibodies, T cell receptors, MHC, IgSF and MhcSF. Nucleic Acids Res 38:D301–307

Kaas Q, Duprat E, Tourneur G, Lefranc M-P (2008) IMGT standardization for molecular characterization of the T cell receptor/peptide/MHC complexes. In: Schoenbach C, Ranganathan S, Brusic V (eds) Immunoinformatics, Immunomics reviews, series of springer science and business media LLC. Springer, New York, pp 19–49 (Chap. 2)

Lefranc M-P, Duprat E, Kaas Q, Tranne M, Thiriot A, Lefranc G (2005) IMGT unique numbering for MHC groove G-DOMAIN and MHC superfamily (MhcSF) G-LIKE-DOMAIN. Dev Comp Immunol 29:917–938

Groove Domain

► Groove (G) Domain

Ground Substance

► Extracellular Matrix

Grounding

► Entity Mention Normalization

Grouping

► Abstraction