Sparse Estimation of Module Gaussian Graphical Models with Applications to Cancer Systems Biology

1. Introduction

Gaussian graphical models (GGMs) provide a powerful framework to represent rich statistical dependencies among random variables. Each edge in a GGM represents a conditional dependency between the two nodes connected. Biologists are increasingly interested in understanding how thousands of genes interact with each other based on gene expression data, which has stimulated considerable research into structure estimation of high-dimensional GGM. It is well-known that the non-zero pattern of Σ^{-1} (inverse covariance matrix) corresponds to the network structure of the GGM. Thus, many authors proposed to obtain a *sparse* estimate of Σ^{-1} in order to learn the network structure of a high-dimensional GGM [1, 2], a method called the *graphical lasso* that independently penalizes each off-diagonal element of Σ^{-1} with an L_1 norm.

While the graphical lasso is a popular approach, it suffers from poor scalability and interpretability. This is because the independence assumption on edge-wise sparsity is unrealistic for many real-world networks that are *structured*, where edges are not mutually independent. In a *gene regulatory network*, genes involved in a similar functional *module* are more likely to interact with each other. There are also high-level interactions between functional modules, which can be difficult to identify in a standard GGM representation (see Figure 1(a)). Importantly, how genes are organized into functional modules and how these modules interact with each other may be scientifically relevant.

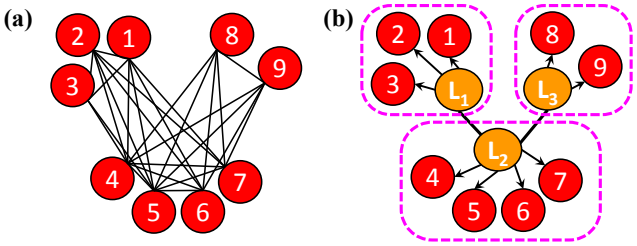


Figure 1. (a): GGM representation of $\mathbf{X} = \{X_1, \ldots, X_9\}$; (b) MGL representation of $\mathbf{X}$ assigned to three modules.

In this paper, we propose a general framework to accommodate the modular nature of many real-world networks. Our approach, called *module graphical lasso* (MGL), is characterized by the incorporation of latent variables into the GGM (see Figure 1(b)). The actual variables are organized into tightly coupled modules and a graph structure is estimated to determine the conditional independencies among modules. Figure 1(b) illustrates a toy example where three latent variables L_1, L_2 and L_3 have mutual dependencies in addition to connections to observed variables by directed edges. Each of L_1, L_2 and L_3 represents aggregate activity level of specific functional modules as defined by a core of tightly coupled genes. The edges among latent variables determine the dependencies among these modules. As shown in Figure 1, MGL provides a more compact representation of the conditional independence relationships compared to the equivalent GGM. Moreover, by modeling the conditional independence relationships among k latent variables, instead of p variables ($k \ll p$), we showed that MGL scales better than standard graphical lasso [2], which enables to learn a GGM with tens of thousands of variables.

2. Related Work

Many authors attempted to incorporate latent variables into GGMs [3, 4, 5, 6, 7, 8, 9, 10]. Toh et al. (2002) proposed to first cluster variables and then learn the dependency structure among the cluster centroids [3]; while MGL uses a coordinate ascent procedure to *jointly* learn the assignment of variables into modules and the network among modules. Chandrasekaran et al. (2012) assume that Σ^{-1} of observed variables decomposes into a sparse matrix and a low-rank matrix, and the low-rank matrix represents the effect of unobserved latent variables [4]. They proposed a convex optimization algorithm that utilizes both L_1 and nuclear norm as penalty terms. The SIMoNe [5] uses an Expectation-Maximization approach [11] for variational estimation of the latent structure while inferring the network among the entire variables. In contrast, MGL performs a more aggressive dimensionality reduction by learning a network of k latent variables instead of p observed variables ($k << p$). Guo et al. (2010) proposed a three-step algorithm: 1) apply the graphical lasso to compute an adjacency matrix of the variables; 2) partition variables into disjoint clusters using the normalized cut algorithm [12]; and 3) estimate a sparse Σ^{-1} with a modified penalty term such that within-cluster edges are less strongly penalized [6]. Given the module assignment of variables, Duchi et al. (2008) and Schmidt et al. (2009) proposed to penalize the L_1-norm and L_∞-norm of the inverse covariance matrix block corresponding to each module in the network of the variables [7, 8]. Marlin et al. (2009) make use of these methods [7, 8], after first identifying the groups of the variables when the modular structure is unknown [9, 10].

3. Module Graphical Lasso

3.1. Preliminaries

We consider learning the GGM with p variables based on n observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is a $p \times p$ covariance matrix. The non-zero pattern of $\mathbf{\Sigma}^{-1}$ corresponds to the network structure [13, 14]. Specifically,

$(\mathbf{\Sigma}^{-1})_{jj'} = 0$ for $j \neq j'$ if and only if X_j and $X_{j'}$ are not connected by an edge. In order to obtain a sparse estimate for $\mathbf{\Sigma}^{-1}$, many authors [1, 2] considered maximizing the penalized log likelihood (*graphical lasso*):

$$\max_{\mathbf{\Theta} \succ 0} \left\{ \log\det \mathbf{\Theta} - \mathrm{tr}(\mathbf{S\Theta}) - \lambda \sum_{j \neq j'} |\Theta_{jj'}| \right\}, \quad (1)$$

where λ is a positive tuning parameter and the constraint $\mathbf{\Theta} \succ 0$ restricts the solution to the space of positive definite matrices. The probabilistic interpretation is that we optimize the joint log-likelihood: $\log P(\mathbf{S}, \mathbf{\Theta}) = \log P(\mathbf{S}|\mathbf{\Theta}) + \log P(\mathbf{\Theta})$, with a prior distribution $P(\Theta_{j,j'}) = \lambda/2 \cdot \exp(-\lambda|\Theta_{j,j'}|)$.The hyperparameter λ adjusts the sparsity of the optimization variable $\mathbf{\Theta}$.

3.2. Module Graphical Lasso Formulation

Let $\mathbf{L} = \{L_1, \ldots, L_k\} \sim N(\mathbf{0}, \mathbf{\Sigma_L})$ be a set of *latent variables*, where $\mathbf{\Sigma_L}$ is a $k \times k$ covariance matrix. Let $\mathbf{X} = \{X_1, \ldots, X_p\}$ be a set of *observed variables*: $X_i|L_{Z_i} \sim N(L_{Z_i}, \sigma^2)$, where Z_i refers to the index of the latent variable which X_i is associated with. Here, we refer the observed variables that correspond to the same latent variable as a *module* $\mathcal{M}$. $\mathbf{Z} = \{Z_1, \ldots, Z_p\}$ defines the module assignment of p variables into k modules. Then, the joint distribution $P(\mathbf{X}, \mathbf{L}, \mathbf{Z}, \mathbf{\Sigma_L})$ can be written as:

$$\begin{aligned} &\prod_{i=1}^{p} P(X_i|L_{z_i}) P(\mathbf{L}|\mathbf{\Sigma_L}) P(\mathbf{\Sigma_L}^{-1}) P(\mathbf{Z}) \quad (2) \\ &= \prod_{i=1}^{p} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(X_i - L_{z_i})^2}{2\sigma^2} \right\} \frac{1}{\sqrt{(2\pi)^k |\mathbf{\Sigma_L}|}} \\ &\quad \exp\left\{ -\frac{1}{2} \mathbf{L}^\intercal \mathbf{\Sigma_L}^{-1} \mathbf{L} \right\} \prod_{j \neq j'} \frac{\lambda}{2} \exp\left\{ -\lambda |\mathbf{\Sigma_L}^{-1}_{jj'}| \right\}. \end{aligned}$$

Given the n observations $\mathbf{x}[1], \ldots, \mathbf{x}[n] \in \mathbb{R}^p$ on $\mathbf{X}$, MGL estimates the latent variables $\mathbf{L}$, the module assignment variables $\mathbf{Z}$, and $\mathbf{\Sigma_L^{-1}}$ of the latent variables. In order to estimate the inverse covariance matrix over $\mathbf{X}$, we can use the relationship between $\mathbf{\Sigma_L^{-1}}$ and $\mathbf{\Sigma_X^{-1}}$, described in *Lemma* 1.

Lemma 1. *The relationship between* $\mathbf{\Sigma_X}$ *and* $\mathbf{\Sigma_L}$ *is as:*

$$\mathbf{\Sigma_X} = \left\{ (1/\sigma^2)\mathbb{I} - \mathbf{C}^\intercal \mathbf{B}^{-1} \mathbf{C} \right\}^{-1}, \quad (3)$$

where $\mathbf{B}$ *and* $\mathbf{C}$ *are defined as follows:*

$$\mathbf{B} = \mathbf{\Sigma_L}^{-1} + \begin{bmatrix} |\mathcal{M}_1|/\sigma^2 & \ldots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \ldots & |\mathcal{M}_k|/\sigma^2 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} -(1/\sigma^2) & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & -(1/\sigma^2) & \ldots & 0 \end{bmatrix}$$

where $|\mathcal{M}_k|$ *means the number of* X *variables in the module* k*, and* $\mathbf{C}$ *is a* $k \times p$ *matrix whose element* $C_{ij} = -(1/\sigma^2)$ *if* $X_j \in \mathcal{M}_i$ *and 0 otherwise.*

4. Learning Algorithm

4.1. Overview

Our learning algorithm optimizes the joint distribution described in (2). Given $X (\in \mathbb{R}^{p \times n})$ containing n observations $\mathbf{x}[1], \ldots, \mathbf{x}[n] \in \mathbb{R}^p$ on $\mathbf{X}$, MGL aims to learn the values in the following: $L (\in \mathbb{R}^{k \times n})$ that contains the values on $\mathbf{L}$ in the n observations $\mathbf{l}[1], \ldots, \mathbf{l}[n] \in \mathbb{R}^k$; $Z (\in \{1, \ldots, k\}^p)$ specifying the module assignment of $X_1, \ldots, X_p$ into k modules; and $\Theta_L (\in \mathbb{R}^{k \times k})$ denoting the estimate of $\mathbf{\Sigma_L^{-1}}$. Using *Lemma* 1, we can obtain $\Theta_X (\in \mathbb{R}^{p \times p})$ as well.

We address our learning problem by finding the joint *maximum a posteriori (MAP)* assignment to the optimization variables – L, Z and Θ_L. This means that we optimize the following objective function $\log P(X, L, Z, \Theta_L; \lambda)$ with respect to L, Z and Θ_L ($\succ 0$ meaning positive definiteness):

$$\begin{aligned} &\log P(\Theta_L) + \log P(L|\Theta_L) + \log P(X|L, Z) \quad (4) \\ &= \frac{n}{2} \left(\log\det \Theta_L - \mathrm{tr}(S_L \Theta_L) \right) \\ &\quad - \lambda \sum_{j \neq j'} |(\Theta_L)_{jj'}| - \sum_{i=1}^{p} \frac{\| X_i - L_{Z_i} \|_2^2}{\sigma^2}, \end{aligned}$$

where S_L denotes the empirical covariance matrix of L, X_i denotes the ith row of the matrix X, L_i denotes the ith row of the matrix L, and λ is a positive tuning parameter that adjusts the sparsity of Θ_L. We assume a uniform prior distribution over $\mathbf{Z}$. We use a *coordinate ascent procedure* over the three sets of optimization variables – L, Z and Θ_L. We iteratively estimate each of the optimization variables until the convergence.

4.2. Iterative estimation of L, Z and $\mathbf{\Theta}_L$

Estimation of L: To estimate L given Z and Θ_L, from (4), we solve the following problem:

$$\max_{L_1, \ldots, L_k} \left\{ -\frac{n}{n-1} \mathrm{tr}\left(L L^\intercal \Theta_L \right) - \sum_{i=1}^{p} \frac{\| X_i - L_{Z_i} \|_2^2}{\sigma^2} \right\}. \quad (5)$$

Setting the derivative of the objective function in Eq (5) to zero with respect to L_m leads to:

$$L_m = \frac{\sum_{X_i \in \mathcal{M}_m} X_i - \frac{n\sigma^2}{n-1} \sum_{i \neq m} (\Theta_L)_{im} L_i}{|\mathcal{M}_m| + \frac{n\sigma^2}{n-1} (\Theta_L)_{mm}}, \quad (6)$$

where $\mathcal{M}_m = \{X_i | Z_i = m\}$, and $|\mathcal{M}_m|$ means the number of variables that belong to $\mathcal{M}_m$. We update L_m for each m ($1 \leq m \leq k$), based on the current values of the other latent variables. If all elements in Θ_L equal to zero, L_m would be set to be the centroid of the mth module. This leads to a nice interpretation of the MGL learning algorithm with respect to the *k-means* clustering. *The k-means clustering algorithm is the special case of the MGL when the latent variables are assumed to be independent* ($(\Theta_L)_{ij} = 0 \ \forall i \neq j$). *More specifically, the MGL is a generalization of k-means with the distance metric determined by the sparse estimate of the latent structure* (Θ_L).

Estimation of Z: Given L and Θ_L, we solve the following:

$$\max_{Z_1,\ldots,Z_p} \left\{ -\sum_{i=1}^{p} \| X_i - L_{Z_i} \|_2^2 \right\}, \quad (7)$$

which finds the module for X_i that minimizes the Euclidean distance between X_i and the latent variable.

Estimation of Θ_L: Given L and Z, we solve the following:

$$\max_{\Theta_L \succ 0} \left\{ \log\det \Theta_L - \mathrm{tr}\,(S_L \Theta_L) - \lambda \sum_{j \neq j'} |(\Theta_L)_{jj'}| \right\}, \quad (8)$$

where S_L means the empirical covariance matrix of L. Since L is given, the optimization problem (8) can be solved by the standard graphical lasso algorithm applied to L.

5. Experiments

Ovarian cancer is the 5th leading cause of cancer death among US women and has a 5-year survival rate of 30% [15]. Learning the gene regulatory network from expression data is an effective strategy to identify novel disease mechanisms [16, 17]. Thus, we applied MGL to three expression datasets containing 10404 gene expression levels in a total of 909 patients with ovarian serous carcinoma [17, 18, 19]. Given the data, MGL estimates Z, L and Θ_L (see (4)), which describe a gene module network characterized by the assignments of genes to modules and the latent structure among the modules (see Figure 1(b)). We evaluated MGL based on: 1) how well the learned model fits unseen data; 2) how significantly the inferred modules are coherent in terms of gene functions; and 3) how well the inferred latent variables are predictive of clinical outcomes (survival time).

Cross-data test log-likelihood results: We created four train-test dataset pairs by using the three expression datasets described above. For each pair, we computed the log-likelihood of the GGM on the test data, using the parameters learned in the training data. Since this application requires learning a network with >10,000 variables, among the methods discussed in Section 2, only Toh et al. [3] allowed comparison with MGL. Table 1 shows that for varying numbers of modules (K) and tunning parameters (λ), MGL achieves better test log likelihood compared to the method of Toh et al. [3].

		1	2	3	4
I	MGL	**-8478.8**	**-8823.1**	**-8160.7**	**-8822.4**
	Toh et al.	-8616.3	-8964.8	-8272.9	-8967.1
II	MGL	**-8594.4**	**-8937.9**	**-8237.3**	**-8909.3**
	Toh et al.	-8702.7	-9051	-8348.4	-9050.7
III	MGL	**-8557.6**	**-8945.6**	**-8283.7**	**-8975.3**
	Toh et al.	-8700.4	-9096	-8445.6	-9174.1

Table 1. For varying (K,λ) values, I. (250, 0.01), II. (250, 0.05), and III. (500, 0.05), we compared MGL with Toh et al. [3] based on the test log likelihood, in four train-test dataset pairs: 1) [17]-[18], 2) [17]-[19], 3) [18]-[17], and 4) [18]-[19].

Functional enrichment of modules: A set of genes assigned to the same module are likely to share similar functions. We applied k-means clustering and used the resulting clusters as a starting point for MGL, and compared between k-means clusters and MGL modules in terms of functional coherence. For each of the 4722 GeneSets from the Molecular Signatures Database [20], we computed the significance of the overlap between the GeneSet and modules (clusters). We applied Bonferroni correction to the hypergeometric p-values and only considered the GeneSets with $pval < 0.05$ in either MGL or k-means. As can be seen in Figure 2(a), for varying numbers of modules (K) and tunning parameters (λ), there are almost always more GeneSets that are more significantly overlapped with MGL modules than with k-means clusters. Additionally, we observed that about 2/3 of the probability mass of the distribution of $-\log(pval_{MGL}/pval_{k-means})$ lies on the right side of the y-axis, which can be seen in Figure 2(b). These results indicate that the MGL improves the module assignment of k-means, resulting in more coherent gene groups.

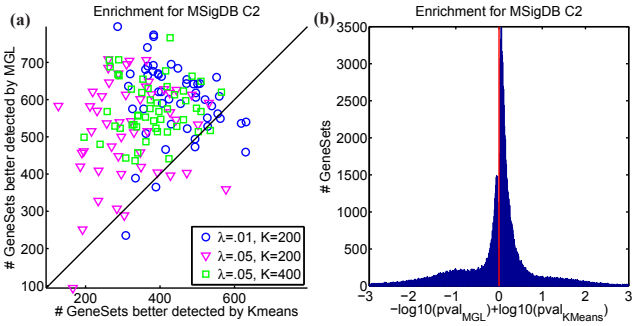


Figure 2. Comparison between MGL and k-means in terms of functional coherence. (a) Each dot represents a k-means run. For each run, the number of GeneSets more significantly overlapped with MGL modules (y-axis) is compared to that with k-means clusters (x-axis) for K={200, 400} and λ={.01,.05}. (b): Probability mass of the distribution of $-\log(pval_{MGL}/pval_{k-means})$. x-axis is limited to the interval [-3,3] for better visualization.

Survival prediction using latent variables: The inferred latent variables could represent activity levels of pathways relevant to the disease process and the clinical outcome. We evaluated how well the inferred latent variables are predictive of survival time of ovarian cancer patients. We trained the penalized Cox regression model using the inferred latent variables as features in a training dataset; then tested the model on a separate test dataset and calculated the concordance index (c-Index) for varying sparsity levels. For two of the sparsity levels selected from the x-axis of the sparsity level vs. c-Index curve, where the trained model estimates 25 and 100 nonzero coefficients respectively, Figure 3 shows the maximum c-Index achieved for varying sparsity levels and the area under the c-Index curve up to the selected sparsity level. We can conclude from Figure 3 that a small number of latent variables learned by MGL can predict the survival time better than the same number of genes or the latent variables inferred by k-means. In this

experiment, k-means model was trained with K=250, and MGL was trained with (K,λ)=(250, .01). The c-Index values were averaged over 50 runs for both methods due to non-deterministic initialization of the cluster assignments. The results were consistent for varying values of K and λ, and for a wide range of nonzero coefficients selected by Cox regression model.

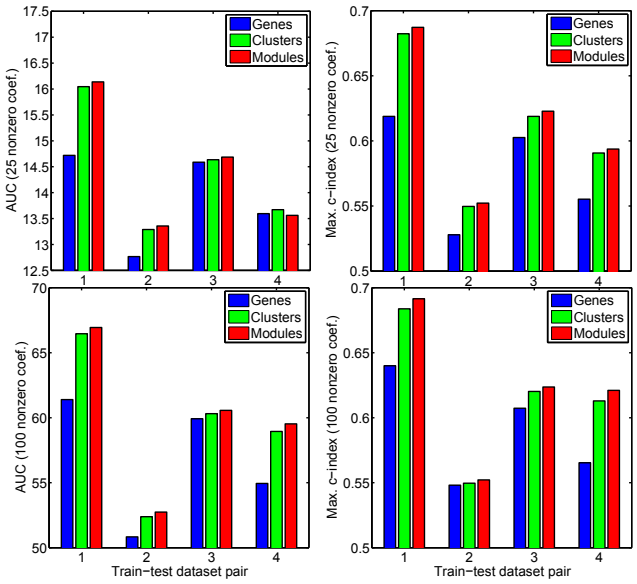
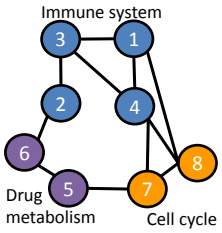


Figure 3. Comparison between genes, MGL modules, and k-means clusters in terms of survival prediction performance. Rows correspond to 25 and 100 nonzero coefficients selected by penalized Cox regression model. Columns correspond to maximum c-Index achieved and area under sparsity vs. c-Index curve up to the selected sparsity level respectively. Each bar group in each plot correspond to a different training-test dataset pair: 1) [17]-[18], 2) [17]-[19], 3) [18]-[17], and 4)[18]-[19].

Interesting Findings: A handful of modules identified by MGL are enriched for processes relevant to tumor biology, drug metabolism, and response to drug therapy. First, there are four modules that are enriched for genes associated with inflammation and immune response. Specifically, module 1 is enriched for cytokine-cytokine receptor interactions (p-value: 9.3×10^{-11}) and the chemokine signaling pathway (p-value: 4.0×10^{-8}), and module 2 is enriched for inflammatory response (p-value: 1.7×10^{-9}). Inflammation, cytokines, and chemokines are directly implicated in regulation of tumor growth, regulation of angiogenesis, invasion, migration, and metastasis [21]. Module 3 is enriched for the natural killer cell pathway (p-value: 6.70×10^{-8}) and module 4 is enriched for signaling in the immune system (p-value 3.11×10^{-16}). Both the natural killer cell pathway [22] and immune signaling [23] play important roles in tumor biology [21]. In Figure 4, we see multiple edges between modules 1 through 4, indicating conditional dependencies among these four modules. Also, we see two modules (7 and 8) are enriched for processes involved in mitosis and the cell cycle. There are suggestive edges between module 4 and modules 7 and 8, because innate immune response can stimulate cell division in neoplastic cells [21]. Module 6 is enriched for drug metabolism of cytochrome P450 (p-value: 7.36×10^{-9}). Cytochromes P450 are very important enzymes in terms of cancer formation as well as activation and inactivation of anticancer drugs [24]. Finally, module 5 is enriched for PDGF for signaling (p-value: 7.52×10^{-9}). PDGF receptor agonists, such as the popular drug Gleevec, have succeeded in treating chronic myelogenous leukemia patients [25].



Mod	Detected Pathway	Mod	Detected Pathway
1	Cytokine receptor interaction/chemokine signaling	5	Signaling by PDGF
2	Inflammatory response	6	Drug metabolism
3	Natural killer cell	7	Mitotic cell cycle
4	Immune system signaling	8	Oocyte meiosis and cell cycle

Figure 4. The pathway structure identified by MGL on the ovarian cancer data with (K, λ) = (200, 0.01). Only the modules discussed in the text are shown.

6. Discussion

We proposed the *module graphical lasso*, a novel high-dimensional GGM representation of tightly coupled sets of variables (*modules*). The MGL algorithm is a novel high-dimensional clustering algorithm that is a generalization of *k-means* clustering, with Mahalanobis distances between variables. The full log-likelihood function (2) defines a non-Euclidean distance metric between the latent variables $\mathbf{L}$ based on Θ_L. There are several possible extensions to MGL. First, MGL could be extended to other graphical models, such as MRFs, with novel distance metrics and clustering properties. Second, the strict assumptions about relationships between latent and observed variables could be relaxed. We could apply soft assignments of variables to modules, use different σ values for different modules, and learn sub-networks within modules. Finally, we plan to apply MGL to gene expression data across multiple healthy and cancerous tissues to identify conserved and differential latent molecular networks driving tumor biology.

[1] N. Meinshausen et al. *Ann. of Stat.*, 34(3):1436–1462, 2006.
[2] J. Friedman et al. *Biostatistics*, 9:432–441, 2007.
[3] H. Toh et al. *Bioinformatics*, 18(2):287–297, 2002.
[4] V. Chandrasekaran et al. *Ann. of Stat.*, 40:1935–1967, 2012.
[5] C. Ambroise et al. *Elect. J. Stat.*, 3:205–238, 2009.
[6] J. Guo et al. *submitted to CSDA*, 2010.
[7] J. Duchi et al. *UAI*, 2008.
[8] M. Schmidt et al. *AISTATS*, 2009.
[9] B.M. Marlin et al. *UAI*, 2009.
[10] B.M. Marlin et al. *ICML*, 2009.
[11] A.P. Dempster et al. *JRSS, Series B*, 39(1):1–38, 1977.
[12] S.X. Yu et al. *ICCV*, 2003.
[13] K.V. Mardia et al. Academic Press, 1979.
[14] S.L. Lauritzen. Oxford Science Publications, 1996.
[15] R.C. Bast et al. *Nature Reviews Cancer*, 9(6):415–428, 2009.
[16] U.D. Akavia et al. *Cell*, 143(6):1005–17, 2010.
[17] Cancer Genome Atlas Research Network. *Nature*, 474(7353):609–15, 2012.
[18] R.W. Tothill et al. *Clin. Cancer Res.*, 14(16):5198–208, 2008.
[19] C. Denkert et al. *J. Pathol.*, 218(2):273–80, 2009.
[20] A. Liberzon et al. *Bioinformatics*, 27(12):1739–1740, 2011.
[21] L.M. Coussens et al. *Nature*, 420(6917):860–867, 2002.
[22] L. Zamai et al. *The J. of Immun.*, 178(7):4011–4016, 2007.
[23] K.E. de Visser et al. *Nature Reviews Cancer*, 6(1):24–37, 2006.
[24] C. Rodriguez-Antona et al. *Oncogene*, 25(11):1679–1691, 2006.
[25] K. Pietras et al. *Cancer Cell*, 3:439–444, 2003.