

# 4 METHODS

The inherent uncertainty in the clustering process is well documented. However, little work has been done to resolve this issue. Here, I briefly discuss prior work that has been done towards that goal, and then present a new methodology to interpret the often-conflicting partitions produced by different clustering configurations.

## 4.1 Motivation & Prior Work

Having presented cluster analysis in Section 4, it might be useful to once again pose the following question: *What does one hope to achieve by clustering microarray data?* Normally, the final goal is knowledge discovery, i.e. that cluster analysis will uncover previously unknown information about biological processes. With cluster analysis, biologically-relevant information is typically discovered in one of two ways: from *individual clusters* or from the *partition structure*. Examples of both types are given in Figure 4.1a.

### *Different Clustering Configurations Produce Different Clusters*

Still, experiments of both types typically rely on a single algorithm; they rarely show results to be robust under different clustering configurations. For instance, a seminal experiment by Bittner *et al.* [15] demonstrates that unsupervised clustering successfully identify phenotypes of cutaneous malignant melanoma.

However, a subsequent study by Goldstein *et al.*[39] showed the results of Bittner *et al.* to break down under different configurations of the clustering algorithm “...noting that these issues are not unique to this particular data set.”

### ***Confronting the Uncertainty in Clustering***

The sources of such indeterminacy have been demonstrated here (in Appendix A) and in the literature (notably [57, 74, 78]). To account for the variability, the standard recommendation is to try different combinations of algorithms and parameters and then conduct a post-clustering evaluation [79].

Halkidi, for instance, has written:

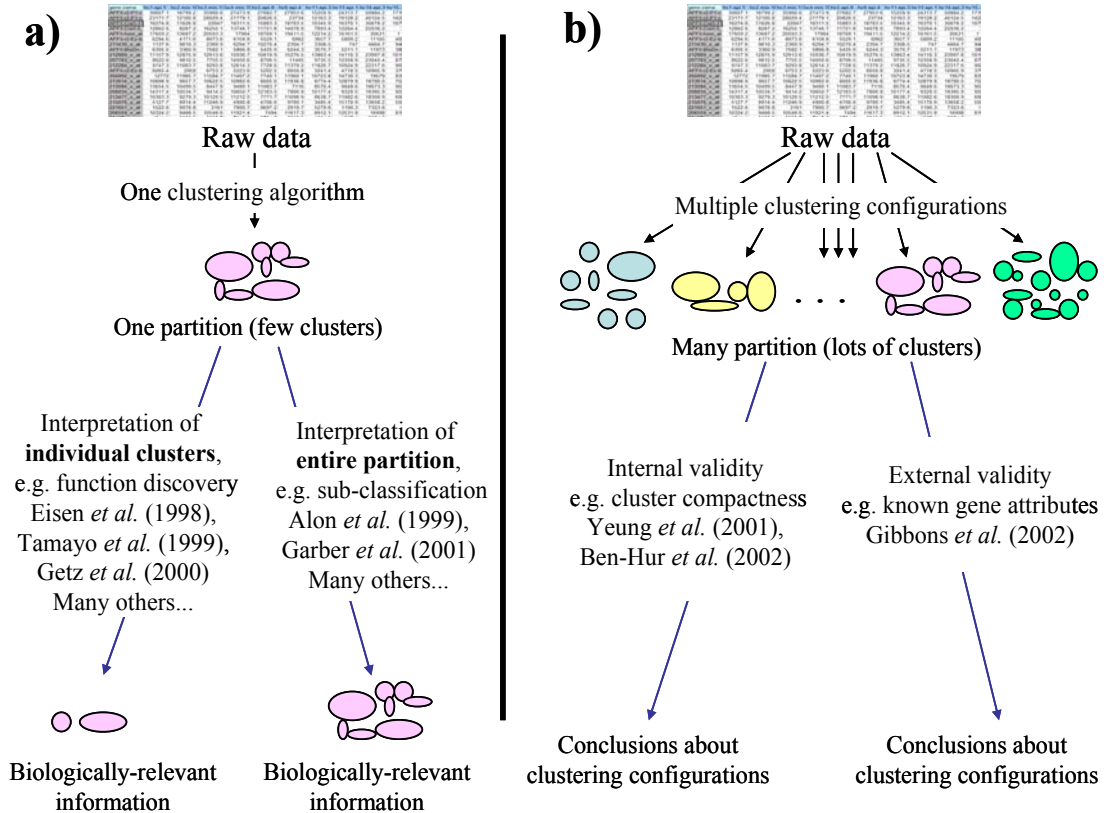
It is important then to be able to choose the optimal partitioning of a data set as a result of applying different algorithms with different input parameter values...it is obvious that the final partition of a data set requires some sort of evaluation in most applications. [44]

This evaluation is typically referred to as *cluster validity analysis*. Cluster validity provides means for assessing the quality of both individual clusters (e.g. Dunn’s indices [30], silhouette method[81]) and complete partitions (e.g. cophenetic correlation coefficient, inconsistency coefficient[1]), thereby providing an ‘objective’ criteria with which to compare clustering configurations.<sup>2</sup> Cluster validity is assessed based on either *internal* or *external* criteria. Though the two have been shown to be somewhat correlated[90], they are fundamentally different approaches. Whereas internal validity is an unsupervised procedure applicable to cluster analysis

---

<sup>2</sup> There is, of course, a certain amount of subjectivity in how one defines validity, i.e. in the choice of validity metric. See, for instance, [56].

in general, external validity is a supervised concept that is application-specific (Figure 4.1b). All of the techniques developed in this thesis measure internal validity.



**Figure 4.1:** Different methodologies to the clustering of microarray data. **a)** 'Standard' approaches to clustering of microarray data. Biologically-relevant information is derived from either the entire partition or individual clusters. **Left branch of a):** Eisen *et al.*[31] use hierarchical clustering to group genes of known and unknown biological functions. If an individual cluster is primarily composed of genes with the same known function, they hypothesize that uncharacterized genes in that cluster also have the dominant function. Similarly, Tamayo *et al.*[84] use SOM clustering to characterize the role of unknown genes in leukemia-related cellular processes, and Getz *et al.*[37] use the same technique to “zero in” on clusters of colon cancer genes. **Right branch of a)** Alon *et al.*[6] produce a partition that reflects the difference between normal and cancerous tissues. Garber *et al.*[36] use the entire partition structure to reveal and identify subgroups of different types of lung cancer. The partition corresponds to prognostic indicators, and “...thus promises to extend and refine standard pathologic analysis.” **b)** Validity-based approaches to clustering of microarray data. Validity is determined using internal criteria (left branch), or external criteria (right branch). **Left branch of b)** Ben-Hur *et al.*[11] and Yeung *et al.*[100] use internal criteria to evaluate clustering configurations. Both make this judgement by slightly permuting the data set, and observing the effects on the clusters. Yeung *et al.*, for instance, present a “Figure of Merit (FOM)” metric that measures the empirical stability of each algorithm under small perturbations of the input data. They suggest that the FOM is a ‘quality metric’ that can be used to assess a given clustering configuration’s stability and predictive power. **Right branch of b)** Gibbons and Roth[38] use external criteria to evaluate different algorithms’ performance under different distance metrics. For each clustering configuration, they assess the extent to which the produced clusters accurately mirror known gene classifications (as defined in a public database). Thus, Gibbons and Roth can judge clustering configurations in a practical setting.

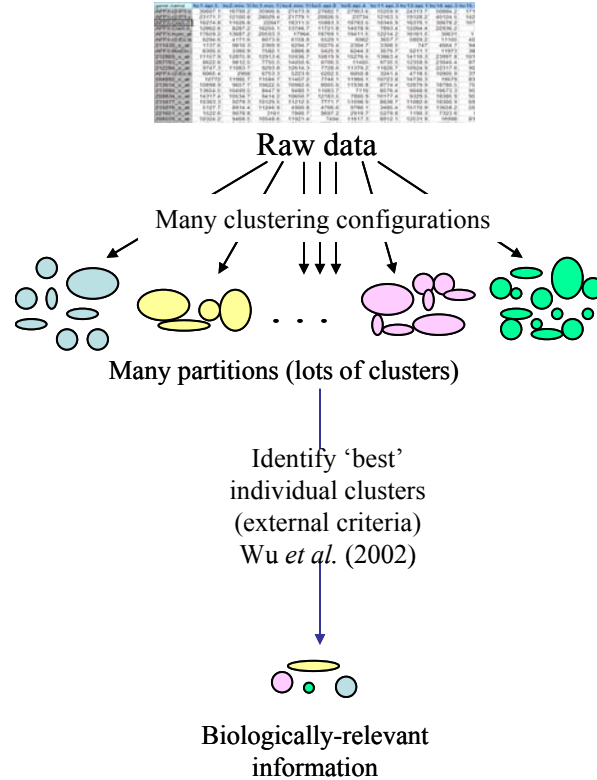
### ***Moving Beyond the Uncertainty***

Cluster validity analysis can potentially provide valuable advice in empirically differentiating between different clustering configurations. However, the advice is inconsistent, and there will always be choices to make in clustering. As Jain and Dubes remark, “the choice of a suitable hierarchical clustering method is an important matter in applications, but theory provides few guidelines for optimizing the choice.”[56] Lacking a firm theoretical foundation, it is common practice to use different clustering algorithms in a very ad-hoc manner. Azuaje and Bolshakova, for instance, give the following advice:

Several algorithms indirectly assume that the cluster structure of the data under consideration exhibits particular characteristics...Unfortunately, this type of knowledge may not always be available in an expression data study. In this situation a solution may be to test a number of different techniques... In general, the application of two or more clustering techniques may provide the basis for the synthesis of accurate and reliable results. A scientist may be more confident about the clustering experiments if very similar results are obtained by using different techniques[7].

### ***The Brute-Force Approach***

In the first study of its kind, Wu *et al.*[98] developed a methodology to eliminate some of the uncertainty in the clustering process. They used an aggregate of ten of the most common clustering algorithms to generate a large database of 14,000 clusters of genes, then, in a manner similar to Eisen *et al.*[31], assigned function to unknown genes based on the contents of individual clusters (Figure 4.2). Though other studies have tested and compared different clustering configurations, Wu *et al.* were apparently the first (and only?) to take advantage of the variability in clustering.



**Figure 4.2** *Wu et al.'s approach* Using existing biological knowledge, each cluster is scored based on the presence of genes with known function. Clusters with high scores contain a large percentage of genes of similar biological function. Genes of unknown function in clusters with high scores are presumed (and verified) to have the same biological function as the known genes in the cluster.

## 4.2 Overview of Methods

The brute-force approach used by *Wu et al.* is unique because it does not assume there is a 'best' clustering configuration. Instead, they treat each configuration as a potential source of information, and look for clusters that maximize their external criteria. A similar approach is taken in this thesis. Like *Wu et al.*, I assume that each clustering configuration offers potential insight into the structure of the underlying expression data. However, the techniques presented in 4.3-4.6 rely on

internal criteria, and represent a more general approach to the mining of clusters for knowledge.

**(4.3) Producing A Large Collection of Clusters: ‘Shotgun’ Clustering:**

The first step is to produce a large quantity of clusters that is robust and reflects a wide range of structure in the original set of samples. In a manner similar to Wu *et al.*, many different combinations of algorithms and parameters are used. In addition, the set of genes used as the basis for the  $n_{\text{used}}$ -dimensional vectors is varied, both in size and composition.

**(4.4) Measuring Global Consistency of the Collection:  $\lambda$ :**

In this section I describe a measure for assessing the overall consistency of the collection of clusters. This can be used to compare and characterize structural aspects of the aggregate of clustering configurations.

**(4.5) Identifying Prevalent Clusters:  $p_{\mathcal{D}}$ :**

Wu *et al.* selected ‘relevant’ clusters based on external information; here a technique is described that uses internal information to identify ‘prevalent’ clusters. These clusters are discovered by many different clustering configurations, and are representative of the structure found in cluster analysis.

**(4.6) Clustering the Clusters: ‘Condensation’ Clustering:**

Lastly, a methodology is presented to ‘cluster the clusters.’ It is proposed that, just as standard cluster analysis can reveal patterns in objects, so can ‘condensation’ cluster analysis reveal patterns in clusters. In this way,

the results from the shotgun stage are pooled and condensed into a cover of the original sample space.

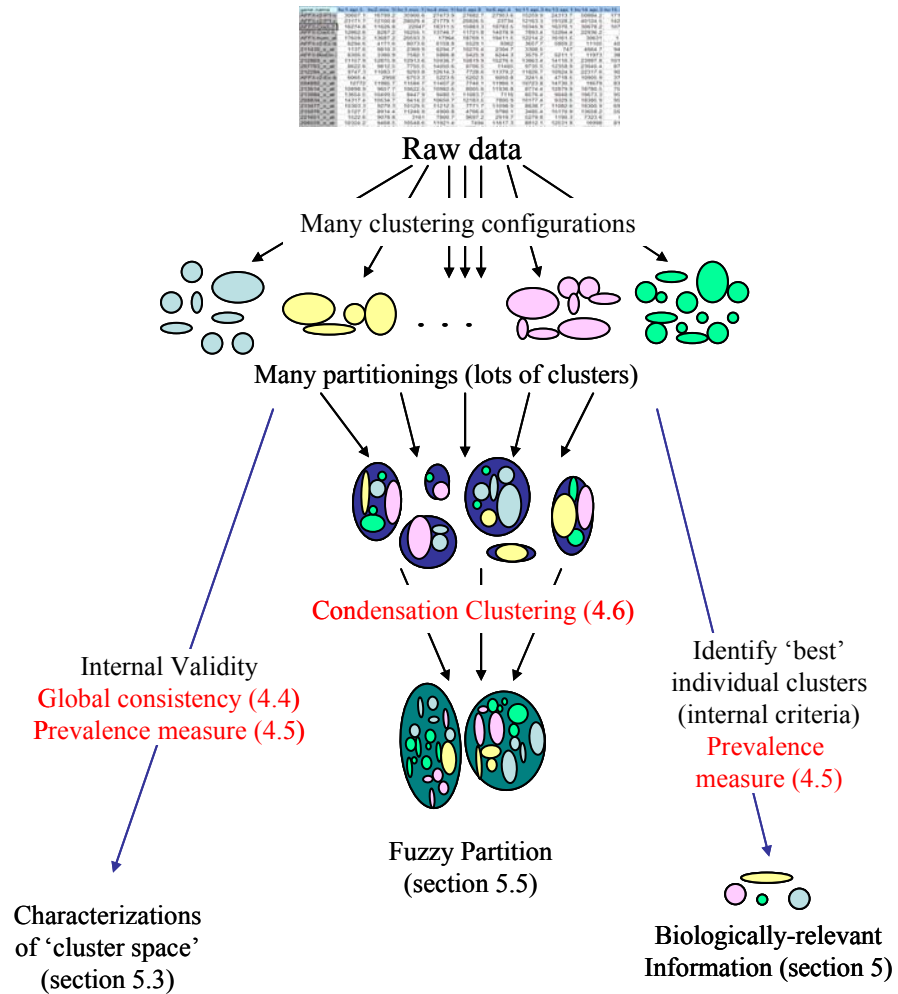


Figure 4.3: Methodology of this thesis

### *A Crude Metaphor*

Metaphorically, shotgun clustering can be seen as a 'voting' phase where many clustering configurations cast an equal vote on what the proper clustering

structure is. The  $\lambda$  and  $p_{\mathcal{D}}$  statistics are used to measure general and specific patterns in the voting. In condensation clustering, the ballots are tallied and a consensus partition is chosen.

### 4.3 Producing A Large Collection of Clusters: ‘Shotgun’ Clustering

In the ‘shotgun’ stage a large number of clustering structures are produced in parallel (see Figure 4.3).<sup>\*code B3.1</sup> Table 4.1 contains a compendium of the algorithms, parameters, and gene subsets used.

Algorithm type	Version Used
Partitioning	<ul style="list-style-type: none"> <li>• k-means</li> </ul>
Hierarchical	<ul style="list-style-type: none"> <li>• Agglomerative hierarchical</li> </ul>
Parameter type <sup>3</sup>	Versions Used
Distance measure	<ul style="list-style-type: none"> <li>• Manhattan (<math>L_1</math>)</li> <li>• Euclidean (<math>L_2</math>)</li> <li>• Minkowski (<math>L_5</math> only)</li> <li>• Pearson correlation</li> </ul>
Linkage function	<ul style="list-style-type: none"> <li>• Single Linkage</li> <li>• Complete Linkage</li> <li>• Ward Linkage</li> </ul>
K-value	<ul style="list-style-type: none"> <li>• dependent on number of samples in dataset</li> </ul>
Seed placement <sup>4</sup>	<ul style="list-style-type: none"> <li>• placement minimizing <math>d_{tot}</math> (see notes)</li> </ul>

<sup>3</sup> Certain parameter/version combinations are only used in one type of clustering (i.e. *k*-means or hierarchical). For instance, linkage metrics aren’t used in *k*-means, and seed placement does not apply to hierarchical clustering.

<sup>4</sup> For each clustering configuration of the *k*-means algorithm, clustering is conducted 10 times with different random seed placement. The 10 replicates are compared, and only the best replicate is kept. The best replicate is the one achieving the minimum value of  $d_{tot}$ .

Gene subset type <sup>5</sup>	Subsets Used
All Genes	<ul style="list-style-type: none"> <li>• Every gene on the microarray</li> </ul>
Canonical subsets	<ul style="list-style-type: none"> <li>• High standard deviation ([31, 84])</li> <li>• High average ([15, 76])</li> <li>• All values in range <math>a \leq x_{i,j} \leq b</math> ([6, 94])</li> </ul>

**Table 4.1:** Parameters used in ‘shotgun’ stage of clustering

Each clustering configuration  $A_\alpha$  represents a unique combination of the above options.  $A_\alpha$  produces a potentially unique partition  $\Pi(A_\alpha)$ . The fact that  $\Pi(A_\alpha)$  is a partition of  $X$  can be inferred from the structure of the algorithms. I will let  $I_1$  denote the collection of all clusters generated by shotgun clustering (for reasons that will become evident later).

Notation	Description
$A_\alpha$	$\alpha^{\text{th}}$ clustering configuration, $1 \leq \alpha \leq cc_{\max}$ , e.g. $A_\alpha = \{algorithm_\alpha, params_\alpha, genes_\alpha\}$
$\Pi(A_\alpha)$	The partition of $X$ produced by $A_\alpha$ (see Definition 3.2)
$I_1$	The collection of all clusters produced in the shotgun stage.

**Table 4.2:** Notation used in this thesis

Using this notation, shotgun clustering produces  $cc_{\max}$  different partitions,

which contain a total of  $\sum_{\alpha=1}^{cc_{\max}} |\Pi(A_\alpha)|$  clusters.<sup>6</sup>

<sup>5</sup> This is, in the author’s opinion, one of the most under-explored issue in microarray clustering. As there have been no studies to determine the most appropriate genes, subsets were chosen based on what is commonly found in the literature. For each listed filtering criterion 3 sets were formed: one with 10 genes, one with 50, and one with 500. <sup>\*code B3.2</sup>

#### 4.4 Measuring Global Consistency of the Collection: $\lambda$

The task now is to make sense of these different partitions and clusters. A first question to ask of the collection of clusters is: *how consistent are the results from the different clustering configurations?* Intuitively, if there were a single dominant structure in the data, all clustering configurations would produce roughly the same partition, i.e. the same set of clusters.<sup>7</sup> If, on the other hand, there were no structure in the data, each clustering configuration would produce a different partition.

A useful approximation of consistency between clustering configurations can be attained by looking for redundancy in the collection of clusters. This approximation is based on the intuition that if two clustering configurations  $A_\alpha$  and  $A_\beta$  both generate the same cluster  $C_p$ , they are both recognizing similar structure in the underlying data  $X$ . Letting  $\Omega$  denote an arbitrary collection of clusters, equation (4.1) measures the level of redundancy in  $\Omega$ :<sup>\*code B4.1</sup>

$$\lambda(\Omega) = 1 - \frac{|\mathbf{U}(\Omega)|}{|\Omega|} \quad (4.1)$$

The collection  $\Omega$  is not a proper set, since it can contain multiple instances of the same cluster. In proper terminology,  $\Omega$  is a **multiset** of clusters – a set of clusters that permits multiple membership. However  $\mathbf{U}(\Omega)$  is a proper set – namely,  $\mathbf{U}(\Omega)$  is

the set of all *distinct* clusters in  $\Omega$ , i.e.  $\bigcup_{\alpha=1}^{cc_{\max}} \Pi(A_\alpha)$ . It follows that  $|\mathbf{U}(\Omega)| \leq |\Omega|$  and

$0 \leq \lambda(\Omega) < 1$ . High values of  $\lambda(\Omega)$  indicate high consistency among clustering

---

<sup>6</sup>  $|\Pi(A_\alpha)|$ , the number of clusters in  $\Pi(A_\alpha)$ , is determined by the value of  $k$  used in the clustering configuration  $A_\alpha$ .

<sup>7</sup> This is not entirely accurate, as there must be at least one distinct partition for each value of  $k$  used.

configurations. The value  $\lambda(\Omega)$  measures internal validity across partitions, much as Yeung *et al.*'s "Figure of Merit" index[100] measures the internal validity of a single partition (cf. Figure 4.1b, left branch).

#### 4.5 Identifying Prevalent Clusters: $p_{\mathcal{D}}$

The last section gives a technique for measuring the extent to which different clustering configurations 'agree;' here I describe a technique for finding the clusters which are 'agreed upon.' The approach is similar to the one used by Wu *et al.*, but whereas Wu *et al.*'s approach was supervised, this approach is unsupervised and relies only on internal criteria. As will be demonstrated in section 5, these prevalent clusters correspond quite closely to known biological subsets of the original dataset  $X$ .

##### *What is a 'Prevalent' Cluster?*

Standard approaches to clustering produce a single partition, such that each sample appears in exactly one cluster. By contrast, after the 'shotgun' stage each sample occurs exactly  $cc_{\max}$  times; clusters can and must overlap. I define a prevalent cluster to be one that is produced by many different clustering configurations. What is the likelihood of finding the same cluster multiple times? Given  $m$  samples, there

can be no more than  $\max\{|\mathbf{U}(I_1)|\} = \sum_{j=1}^m \binom{m}{j} = 2^m - 1$  unique clusters. If the

distribution of clusters were random (which it is not), the probability of observing  $\tau$  or more occurrences of any arbitrary cluster  $C_p$  is approximately<sup>8</sup>

$$P(m, cc_{\max}, \tau) = (2^m - 1) \cdot \sum_{i=\tau}^{cc_{\max}} \binom{cc_{\max}}{i} \left( \frac{1}{2^m - 1} \right)^i \left( 1 - \frac{1}{2^m - 1} \right)^{cc_{\max} - i} \quad (4.2)$$

To give the reader a sense of the magnitude of  $P(m, cc_{\max}, \tau)$ : for 30 samples clustered to produce 10,000 clusters, the probability of any cluster appearing three or more times is  $P(30, 10000, 3) = 2.36 \times 10^{-54}$ . There are many approximations being made here, but  $P(m, cc_{\max}, \tau)$  as a heuristic indicates that the expected redundancy is quite small. Actual values are experimentally determined in section 5.

### ***Making ‘Prevalent’ Less Stringent***

Given the nearly nonexistent likelihood of finding the same exact cluster many times, it is natural to instead look for ‘similar’ clusters that appear many times. Assuming a metric  $\mathcal{D}$  exists to assess dissimilarity of two sets  $C_1$  and  $C_2$ , we define  $C_1$  to be in the  $\varepsilon$ -neighborhood of  $C_2$  iff  $\mathcal{D}(C_1, C_2) \leq \varepsilon$ . Thus, instead of looking

---

<sup>8</sup> This is actually the upper bound on  $P$ .  $P$  was derived by first reinterpreting the situation in the following way: Given  $n$  independent identically distributed trials with  $m$  outcomes, what is the probability that there exists an outcome  $E_\alpha$  that is observed  $\geq \tau$  times? The probability that fixed  $E_1$

is observed  $\geq \tau$  times is  $P(E_1) = \sum_{i=\tau}^n \binom{n}{i} \left( \frac{1}{m} \right)^i \left( 1 - \frac{1}{m} \right)^{n-i}$ . For arbitrary  $E_\alpha$ , the probability is

exactly is  $P\left(\bigcup_{\alpha=1}^m E_\alpha\right)$ . Equation (4.2) gives an expression for  $\sum_{\alpha=1}^m P(E_\alpha)$ , where

$\sum_{\alpha=1}^m P(E_\alpha) \geq P\left(\bigcup_{\alpha=1}^m E_\alpha\right)$ . In this instance, the approximation is sufficient since the upper bound

$\sum_{\alpha=1}^m P(E_\alpha)$  is already extremely small.

for a cluster that appears frequently, we look for a cluster with many clusters in its  $\varepsilon$ -neighborhood. The set of clusters in the  $\varepsilon$ -neighborhood of  $C_p$  is defined to be  $S_{\mathcal{D}}(C_p, \varepsilon) = \{C_q \mid \mathcal{D}(C_p, C_q) \leq \varepsilon\}$ . By appropriately tuning  $\varepsilon$ , the following fuzzy relation will denote the *prevalence* of  $C_p$ :<sup>9</sup>

$$p_{\mathcal{D}}(C_p, \varepsilon) = |S_{\mathcal{D}}(C_p, \varepsilon)| \quad (4.3)$$

## 4.6 Clustering the Clusters: ‘Condensation’ Clustering

Shotgun clustering creates a large collection of clusters,  $\lambda$  measures the overall consistency and the value  $p_{\mathcal{D}}$  identifies prevalent clusters. The next technique, ‘condensation’ clustering as it will be referred to, is to cluster the clusters. Condensation clustering is a natural step in the exploratory mining of clusters for knowledge, and is a direct extension of the  $\varepsilon$ -neighborhood approach presented in the last section. The value of  $p_{\mathcal{D}}(C_p, \varepsilon)$  represents the density around  $C_p$ ; condensation clustering essentially seeks out pockets of high density and forces these dense areas to merge. Fundamental concepts are presented in sections 4.6.1-4.6.4, after which the actual algorithm is described in 4.6.5. The interpretation of the methods is deferred until section 5, where we test real data.

---

<sup>9</sup> Here I use  $p_{\mathcal{D}}(C_p, \varepsilon)$  rather than the typical probability  $\frac{p_{\mathcal{D}}(C_p, \varepsilon)}{|\Omega|}$  because the cluster space is extremely sparse, and  $p_{\mathcal{D}}$  will only be used to measure relative prevalence within  $\Omega$ . Dividing by  $|\Omega|$  will only scale all results by a constant factor.

The easiest way to describe condensation clustering is with an example. Suppose there are  $m = 50$  samples,  $cc_{max}=300$  clustering configurations and  $|I_1|=1000$  clusters generated by shotgun clustering. The idea behind condensation clustering is to iteratively merge similar clustering in  $I_1$ , thereby condensing the 1000 clusters into a manageable number that still reflect the original structure of  $I_1$ .

Let the ‘condensation factor’ be  $\tau=2$ . Since 300 configurations were used, after shotgun clustering each sample is guaranteed to occur a total of 300 times, though never more than once in a given set. Now, we use condensation clustering on the 1000 clusters. With a condensation factor of 2, similar clusters are pooled until there are only  $\frac{1000}{2} = 500$  clusters left. No samples are dropped; these 500 clusters still contain 300 instances of each sample, though the average cluster size has increased (also by a factor of 2). The process is now repeated. The 500 clusters are pooled based on similarity until only 250 are left. This process continues until there are only two (very large) clusters remaining. In this example, condensation halts after 9 iterations. See Figure 4.3(center) for a diagram of the process.

#### **4.6.1 Clustering Clusters**

The first task is to find groups of clusters in  $I_1$  that can be merged. Cluster analysis finds groups in data, so it is natural to use cluster analysis to find groups of clusters. The approach taken by the author is to convert each cluster into a vector in  $m$ -dimensional Boolean space, where the  $i^{\text{th}}$  component of cluster  $C_p$  is defined as

$$C_{p,i} = \begin{cases} 1, & x_i \in C_p \\ 0, & x_i \notin C_p \end{cases} \quad (4.4)$$

Again, assuming a metric  $\mathcal{D}$  exists to measure the distance between two sets, it is now relatively straightforward to cluster these clusters. For instance  $k$ -means, when modified to run on clusters, would look like the following:

- (1) Choose  $k$  seed clusters from  $I_1$ . These initially serve as the  $k$  centroids.
- (2) Assign each cluster  $C_p$  to the nearest centroid.
- (3) Recalculate all cluster centroids.
- (4) Iterate until convergence.

The ‘nearest’ in step (2) is defined by  $\mathcal{D}$ , and the centroids can be calculated using the expression

$$\bar{C}_p = \frac{\langle C_{p,i} \rangle \cdot X}{|C_p|} \quad (4.5)$$

where  $\langle C_{p,i} \rangle$  is the  $1 \times m$  vector form of  $C_p$  defined in equation (4.4) and  $X$  is

the original  $m \times n$  expression matrix.

#### 4.6.2 Merging Clusters to Form Multisets of Samples

Given a group of clusters identified as ‘similar,’ condensation clustering merges all of the clusters in the group. Formally, two clusters  $C_p$  and  $C_q$  are merged using the *multiset sum* operation, which is the natural generalization of the standard union operation to multisets[72, 83]. The resulting cluster  $C_r = C_p \oplus C_q$  is a multiset

of samples that can contain multiple instances of a given sample. Multiset sum is the only multiset operation that will be used in this thesis; all other operations on multisets should be regarded as the ‘standard’ set operations, e.g.  $\mathbb{C}_r \cap \mathbb{C}_s = \mathbf{U}(\mathbb{C}_r \oplus \mathbb{C}_s)$ . A summary of the notation is listed in Table 4.3.

Notation	Description
$\mathbb{C}_r$	A multiset of samples, e.g. $\mathbb{C}_3 = \{x_1, x_2, x_2, x_3, x_3\}$
$ \mathbb{C}_r $	Cardinality of $\mathbb{C}_r$ , e.g. $ \mathbb{C}_3 =5$
$\mathbf{U}(\mathbb{C}_r)$	Unique samples in $\mathbb{C}_r$ , e.g. $\mathbf{U}(\mathbb{C}_3) = \{x_1, x_2, x_3\}$
$\mathbb{C}_r \oplus \mathbb{C}_s$	multiset sum of $\mathbb{C}_r$ and $\mathbb{C}_s$
$\mathcal{D}(\mathbb{C}_r, \mathbb{C}_s)$	The distance between (dissimilarity of) $\mathbb{C}_r$ and $\mathbb{C}_s$
$\mathbb{C}_{r,i} = \text{card}(\mathbb{C}_r, x_i)$	number of occurrences of $x_i$ in $\mathbb{C}_r$
$\langle \mathbb{C}_{r,i} \rangle$	The vector representation of $\mathbb{C}_r$
$\widehat{\mathbb{C}}_r$	The normalized multiset of $\mathbb{C}_r$ : $\widehat{\mathbb{C}}_r = \frac{\langle \mathbb{C}_{r,i} \rangle}{ \mathbb{C}_r }$

**Table 4.3:** Multiset notation used in this thesis<sup>10</sup>

### 4.6.3 Clustering Multisets

Once the clusters from  $I_1$  have been merged to form multisets, cluster analysis is used to pick similar groups of multisets which are merged, and the process is repeated until there are only two large multisets remaining. However, multisets cannot be directly clustered using the technique of 4.6.1, since they are not  $m$ -dimensional Boolean vectors. Just as equation (4.4) defines the  $i^{\text{th}}$  component of

<sup>10</sup> This is not the notation typically found in the literature. The standard notation requires a formal definition of multiset that is far more complicated than is necessary, and which would distract the reader from the essence of this thesis.

$\mathbb{C}_p$  in  $m$ -dimensional Boolean space, so can we define the  $i^{\text{th}}$  component of  $\mathbb{C}_r$  in the  $m$ -dimensional space of non-negative integers:

$$\mathbb{C}_{r,i} = \text{card}(\mathbb{C}_r, x_i) \quad (4.6)$$

where  $\text{card}(\mathbb{C}_r, x_i)$  is the number of occurrences of  $x_i$  in  $\mathbb{C}_r$ . Then we can view

$\langle \mathbb{C}_{r,i} \rangle$  as the  $m$ -dimension vector representation of  $\mathbb{C}_r$ , and generalize equation (4.5)

from  $\bar{\mathbb{C}}_p$  to  $\bar{\mathbb{C}}_r$ :

$$\bar{\mathbb{C}}_r = \frac{\langle \mathbb{C}_{r,i} \rangle \cdot X}{|\mathbb{C}_r|} \quad (4.7)$$

Equation (4.7) represents a transformation from multiset space to the original  $n$ -dimensional expression-space. The value  $\bar{\mathbb{C}}_r$  can be seen as the *weighted* centroid of  $\mathbb{C}_r$ , where each of the  $n$  dimensions is weighted by the number of occurrences of  $x_i$  in  $\mathbb{C}_r$ . Using a metric  $\mathcal{D}$  and the equation for  $\bar{\mathbb{C}}_r$ , ‘ $k$ -means on clusters’ can be generalized to run on multisets as well.

#### 4.6.4 Distance Between Multisets

The algorithms presented above, as well as the prevalence measure  $p_{\mathcal{D}}$ , depend on a metric  $\mathcal{D}$  to assess distance between clusters and multisets. If the clusters are proper sets, standard linkage functions are adequate (see Appendix A, section A.2.2); however, linkage functions do not account for the multiple membership property of multisets. To the author’s knowledge, the concept of dissimilarity between multisets has not been addressed in the literature, though there

are many direct analogies to the metrics just mentioned. For instance, setting  $\mathcal{D}(\mathbb{C}_r, \mathbb{C}_s) = d(\overline{\mathbb{C}}_r, \overline{\mathbb{C}}_s)$  would be analogous to using centroid linkage, and setting

$\mathcal{D}(\mathbb{C}_r, \mathbb{C}_s) = 1 - \frac{|\mathbb{C}_r \cap \mathbb{C}_s|}{|\mathbb{C}_r \oplus \mathbb{C}_s|}$  would be quite similar to using the Jaccard coefficient. Of

course, some metrics are patently inappropriate. As an example, given only  $m$  samples, single linkage distance between  $\overline{\mathbb{C}}_r$  and  $\overline{\mathbb{C}}_s$  would be expected to converge to 0 as  $|\mathbb{C}_r|, |\mathbb{C}_s| \rightarrow m$  since the likelihood of overlap would quickly grow.

A distance metric appropriate to multisets must account for the relative presence of each sample. To address this concern, I define the following metric  $\mathcal{D}(\mathbb{C}_i, \mathbb{C}_j)$  to measure the dissimilarity of two multisets. \*Code B5.3

$$\mathcal{D}(\mathbb{C}_r, \mathbb{C}_s) = \frac{1}{2} \sum_{i=1}^m |\hat{\mathbb{C}}_{r,i} - \hat{\mathbb{C}}_{s,i}| \text{ for each } r, s \quad (4.8)$$

$$\text{where } \hat{\mathbb{C}}_r = \frac{\langle \mathbb{C}_{r,i} \rangle}{|\mathbb{C}_r|} \quad (4.9)$$

The vector  $\hat{\mathbb{C}}_r$  can be said to represent the *normalized multiset* of  $\mathbb{C}_r$ , insofar as  $\sum_i \hat{\mathbb{C}}_{r,i} = 1$ . Thus,  $\mathcal{D}(\mathbb{C}_r, \mathbb{C}_s)$  measures the element-wise differences between the relative compositions of  $\mathbb{C}_r$  and  $\mathbb{C}_s$ . The constant  $\frac{1}{2}$  ensures  $0 \leq \mathcal{D} \leq 1$ .

Just as there are many potential metric spaces for samples, there are many potential metric spaces for multisets. The effect of the above  $\mathcal{D}$  is that the magnitude of each multiset is ignored; what is counted is the relative presence of each sample within a multiset. Thus,  $\{x_1, x_2, x_3\}$  and  $\{x_1, x_1, x_2, x_2, x_3, x_3\}$  are the ‘same’ multiset

( $\mathcal{D}=0$ );  $\{x_1\}$  and  $\{x_1, x_2, x_2, x_2, x_2, \dots\}$  are quite different multisets. The metric  $\mathcal{D}$  can, of course, be applied to normal clusters; doing so would be very nearly the same as using 1-Jaccard coefficient, albeit a normalized version.

#### 4.6.5 Condensation Algorithm

Using the concepts from the past few sections, it is now possible to formalize the condensation clustering algorithm: <sup>\*code B5.1</sup>

- (1) Run shotgun clustering to generate  $cc_{\max}$  partitions. *Optional*: discard any partitions with poor distribution of patients, e.g.  $m - \sigma$  patients in first cluster and  $\sigma$  patients in second, where  $\sigma$  is a small constant (singleton bias – see section 5.3.2).
- (2) Choose a multiset algorithm (*either* multiset  $k$ -means or multiset hierarchical).<sup>11</sup>
- (3) Choose a condensation factor  $\tau$ . Set  $k = \text{CEILING}\left(\frac{|I_1|}{\tau}\right)$
- (4) Run multiset clustering on  $I_i$  using the chosen algorithm (initially  $i=1$ , and  $I_1$  is the clusters generated by the shotgun stage). The output from multiset clustering will be another collection of multisets. Call this output  $I_{i+1}$ .

---

<sup>11</sup> Multiset hierarchical is a generalized form of the ‘normal’ hierarchical algorithm described in section 3.2.1, much as multiset  $k$ -means generalizes ‘normal’  $k$ -means. In multiset hierarchical,  $\mathcal{D}$  is used to both generate (step 1) and update (step 2) the dissimilarity matrix  $\mathbf{D}$ . Matlab code is included in the appendix for multiset hierarchical.

- (5) Redefine  $k = \text{CEILING}\left(\frac{|I_{i+1}|}{\tau}\right)$
- (6) Repeat steps (4) and (5) until  $|I_r|=2$  or until the desired condensation has been achieved.

The number of multisets decreases by a factor of  $\tau$  with each iteration.

Therefore, after the  $i^{\text{th}}$  iteration there will be roughly  $\frac{|I_1|}{\tau^{i-1}}$  multisets. The maximum number of iterations is roughly  $\log_{\tau}(|I_1|)$ .<sup>12</sup>

---

<sup>12</sup> This condensation clustering technique clusters multisets of samples. Note that a similar technique could not reasonably be used to cluster multisets of clusters. At each iteration, the complexity of the datatype would increase, e.g. at  $I_2$  you would be clustering multisets of multisets. Though this is not bad of its own right, the sparseness of the input space would increase exponentially.