分类号 密级

U D C 编号

中南大學

CENTRAL SOUTH UNIVERSITY

硕士学位论文

论 文 题 目 组织微阵列及其图片聚类分析系统的研究和制作

学 科 、 专 业 生物医学工程

研 究 生 姓 名 王飞

指 导 老 师 及

专业技术职务 何继善(教授) 熊平（教授）

摘 要

聚类分析是用数学的方法解决大量有联系的事物的分类问题的学科，已经广泛应用在生物学、医学、气象学、经济学、社会学等需要做分类的很多学科中，特别是事物间的关系较为模糊时更显它独特的优势。被称为生命科学中一大突破性发明的组织微阵列因为其信息量的庞大，挖掘有用信息的难度显得特别大，所以在它诞生近八年的时间还未得到广泛的应用，甚至还在探讨怎么去制作组织微阵列。针对这些情况，本文对组织微阵列做了详细的论述，分析了它的应用范围，并总结出一套详尽的制作步骤，这对它的普及将起着很大的作用。然后将聚类分析引入组织微阵列的研究中，以人们的具体研究目的让它们进行不同的归类，使分析简单化。最后，又用软件的形式实现了组织微阵列图片聚类的各个步骤，使对它的分析最大程度的简单化，也使得从海量数据中提取信息成变得容易。从而，可以将癌症等疑难病实现各种统计学上的分类，找出它致病的各种因素所占的权重，最后实现对它的治疗及预防。

关键词 组织微阵列，组织芯片，生物微阵列，聚类分析

ABSTRACT

Clustering Analysis is a course to class a great deal of related things under many math methods. In order to get every class of our aims, it is widely used in many fields: biology, iatrology, meteorology, economics, sociology and so on. And it is mostly useful when criterions are faintness. Tissue microarray (TMA) is entitled as a pivotal invention in life sciences. It was born 8 years ago, but its area for using still very small, because of its huge data capacity. How to dig out so many data becomes a big problem. In nowadays, many papers' viscera are how to make TMA in Chinese magazine. To solve these, in this dissertation, I give out a particular explain for TMA, and sum up a process to make TMA. Then, a great many words are given to clustering analysis to show its' big functions in TMA researches. It makes the researches become very easy. At last, some useful soft wares about TMA pictures clustering analysis are compiled. Accordingly, TMA clustering analysis becomes easer and easer. So, dig data from a grate many TMA comes true, and one day, we can get every factor's proportion that lead to cancer or some other difficult-solving diseases after a statistics classifying. In this way, to cure and to prevent the disease is also an easy thing.

KEY WORDS tissue microarray, TMA, bio-microarray, clustering analysis

目 录

第一章 绪 论

1.1 引言

“21 世纪是生命科学的世纪”，随着科学技术的发展，现代生命科学几乎可以和所有的学科进行交叉组成新的学科，出现了许多新的名词——生物物理学、生物化学、数学生物学、生物信息学、生物医学等等，然后再次交叉，如生物信息物理学和生物医学影像学。但从总的研究范围看，它是同时向着宏观和微观两边发展，而本文要论述的组织微阵列则是其中一座搭在宏观生物学和微观生物学之间的桥梁。对它进行深入的研究便显得意义重大，怎样进行研究却又是摆在我们面前的一个难题。

1.2 组织微阵列

组织微阵列是一个新鲜名词，最早是由 Wan，Fortuna 和 Furmanski 于 1987 年在 Journal of Immunological Methods 做了描述[9]，直到 1998 年[7]美国国家人类基因实验室 Kononen 教授和他的同事们将它实现，并进行了全面报道。

组织微阵列，顾名思义，就是将很小的组织排成阵列，“微”字更让我们想象到它小的程度，目前，已经有商业化的组织微阵列产品出现，最常见的是把它制成玻片标本，就是我们生物学实验中用的那种玻片，可是就是这么小小的一张玻片却容纳着 60 多个组织标本，每个组织直径仅约为 1 毫米，意为着只将它放在显微镜下便可以对 60 多个组织样本进行观察和对比分析研究，容纳的信息量令人惊讶，而更让人惊讶的是，60 只是一个很小的数目，根据需要，几百甚至上千也是可以做到的。同时，因为制作过程中，每个玻片上的阵列是同条件下生成的，克服了传统病理学切片因切片厚度和染色等造成的差异。而且在制作时组织的取样是有针对性的，能够做到所取组织定位准确，大大减少了无效组织数量。应用前景广阔。

组织微阵列一出现，很快成为众多相关学者研究的焦点。到了 2001 年，国内相继有人开始对它进行研究，但是不管是国外还是国内，对它的应用仍然很有局限，受传统病理学切片的影响，它一般是被认为高通量化了的病理切片，所以，大量的文章都是与癌症[25] [53]（如在某种癌症的某个病例上的应用）有关，很少人能想着去利用好它的高信息量这个优点去主动地挖掘其他信息。究其原因，我认为是组织微阵列包含信息太多，选不出合适的方法去把握更多信息。就拿上边

的一张 60 点阵的玻片来说，每个点阵又包含着许多信息，把它放在显微镜下一个一个的点阵进行观察，还没看到一半，最先看的点阵的信息早就忘记了。如果看一个记录一个，最后进行整理分析，效率又太低了。其实我们可以根据需要把它们分类，让计算机代替我们做一些简单却又繁重的工作，分类后再进行对比研究就简单的多了。这个按需要先进行分类再做分析研究的过程就是聚类分析。

1.3 聚类分析

聚类，简单的说就是归类，是专门解决庞大信息分类问题的一种数学方法。已经广泛应用在生物学、医学、气象学、经济学、社会学等需要做分类的很多学科中。

聚类方法特别多，主要分成以下几大类：聚合法、分裂法、调优法、加入法、最优分段法、图论法、预报法、变量筛选法等。但是不管什么方法，我们都是要先选择一个对比的标准来表示各个样本或类之间的相似程度进而对它们进行归类，表征这个标准的量经常是“距离”。这里的距离是广义的距离，计算的方法也是多种多样，根据样本性质，适当的选取“距离”直接关系到聚类结果好坏。以聚合法中的一种方法系统聚类法为例，在聚类过程中因距离计算方法的不同又常分成八种，即最短距离法、最长距离法、中间距离法、重心法、类平均法、可变类平均法、可变法、离差平方和法，在每种方法中又可以选取不同的距离表达，如明氏距离、马氏距离和兰氏距离等。

1.4 论文背景及主要工作

1.4.1 论文背景及研究意义

作为生命科学中起着“桥梁”作用的组织微阵列自其诞生已经有近 8 个年头了，从各类相关文献看，对它的研究还可以说是刚刚起步。文章主要涉及到三个方面：概念介绍、制作、应用。直到 2004 年在很多核心期刊还有不少专门对它进行概念介绍的文章[6][26][27]，名字一般类似为一种新的生物芯片、芯片家族的新成员、新兴细胞生物学技术或直接就写成组织微阵列等。对于制作，这是现在文章的主要组成，大量文献显示，我们仍在探讨怎么去制作组织微阵列[40−52]，而且这些文章中关于制作要么就是笼统的介绍一下，要么就是在某个工艺上谈作者的经验。应用研究上，始终不能发挥它高信息量的优势，更为直接的说，就是对它研究时，由于工作量大，往往对微阵列的点阵数量需求太少，这就是为什么商业化的微阵列一般只有 60 点阵左右，致使研究产生了很大的局限性[28-35]。没有足

够量的点阵信息怎么能保证研究成果的普遍性呢，为了让它真正实现高信息量的优势，必然需要进行合适的归类，即后边介绍的聚类分析，经过文献论证，国内尚无对组织微阵列以聚类分析的方法进行过系统研究。如果组织微阵列能够按照我们的要求容易地实现分类，那将对组织微阵列以后的统计学研究起着巨大的推动作用。

1.4.2 论文的主要工作

因为国内组织微阵列的相关文章和报道总的来说还是比较少，所以本文选题为组织微阵列及其聚类分析系统的研究和制作。文中将对组织微阵列以综述的形式，从概念到应用，到怎样制作，作了详细的论述。然后针对它数据量大、分析难的问题，引入聚类分析的方法，主要用一个例子的实现来讲述聚类过程，通过走访了许多病理学和血液方面的医生，最后以一个医生提出的分类依据，以十五张乳腺癌的微阵列照片为例进行聚类。最后证明分类效果良好。论文最后则介绍了怎样让计算机代替我们从事繁重的聚类计算，即用软件实现了各个模块。在以后的研究中，不断获取更多的可行性分类依据后，逐步将他们全部用计算机软件实现，最后整合成一个自动化程度较高，操作简单的拥有庞大分析功能的组织微阵列的聚类分析系统软件。

论文各章的内容简述如下：

第二章主要是通过大量文献总结了组织微阵列的应用，结合自己的实际操作和文献翻阅设计了一套组织微阵列制作的步骤，并指出了各步需要注意的相关问题。

第三章主要对实际例子进行了完整的聚类，证明了从组织形态上进行聚类的可行性。

第四章主要讲述了怎样用计算机实现聚类中的四个步骤，重点实现了提取数据和距离阵的计算这两个涉及到庞大计算量工作的模块。

第五章为全文的一个总结，谈了对组织微阵列图片进行聚类分析的可能性的突破点，并指出了以后重点研究的目标。

1.5 论文其他说明

由于条件有限，本论文中所有微阵列图均是我用同一相机（每图为 3145728 像素点）、同一显微镜下拍摄的不同人的经过 HE 染色的乳腺癌微阵列照片，由于缺乏更广泛的其它组织微阵列图片，在考虑误差方面定会有不周之处，欢迎给出意见和建议。

本论文中附录 2 中的亮度分布图是我用 Photoshop7.0 得到的，各分段数据是手工提取后一一读取的。距离的计算则是将数据生成 Excel 表格，然后进行函数编辑计算出来的。人工做了聚类的矩阵计算，并在 Word 中画图。以上工作费了很长的时间和精力，为的是尽量找出各步应该注意到的情况，以使第四章中编写的各软件模块更合理，从而也得到了较为标准的数据，可以作为验证计算机实现的各模块效果的标准。

第二章 组织微阵列综述

2.1 概念

2.1.1 生物芯片技术

组织微阵列也许对大家还是个陌生的概念，所以我在这里先介绍一个生物医学界内炒得很热的东西，那就是生物芯片，该技术 20 世纪 80 年代兴起的，它是物理学、微电子学和分子生物学综合交叉形成的高新技术[1][2]。芯片这个称呼源于计算机芯片的概念，在计算机芯片上排列的是密集的电路，类比下，生物芯片上则排列的是密集的探针阵列（Array）。所以在外文文献中这类芯片实际名字为微阵列（Microarray），也就有了后来的组织微阵列。

生物芯片，或者叫生物微阵列，是指在面积不大的基片表面(玻璃、硅片、聚丙烯酰胺凝胶、尼龙膜等)上有序地排列上可寻址的识别分子，使成千上万个与生命有关的信息集中在一块芯片上，在特定条件下与目的分子进行结合或反应，其反应结果用同位素法、化学荧光法、化学发光法或酶标法显示，然后用精密扫描仪记录，最后通过计算机计算软件分析，综合成可读的ＩＣ信息[3]，达到对生物分子、细胞、组织的高通量检测分析[4]。

目前最常见的生物芯片是基因芯片（Gene chip）和蛋白质芯片（Protein chip）。其中以基因芯片发展最快，已经成功应用于杂交测序、基因表达分析、突变检测、ＤＮＡ多态性分析、基因分型、药物筛选、微生物鉴定与检测、疾病诊断、毒理学研究等方面[4][5] ，已经有了很多商业化产品。对于蛋白质芯片，出现的比较晚，但也取得了一些重大进展，如活性保持等。继它之后，又出现了组织芯片、细胞芯片、抗体芯片等，这其中，组织芯片，即组织微阵列，它可以一次对 1000 份以上[30]的组织样品同时进行分析研究，从而克服了传统分析方法的诸多缺陷，因为其无比的应用前景，一下成为人们研究的焦点。

2.1.2 组织微阵列概念

组织微阵列(Tissue microarray，TMA），因为它是生物芯片的一种，在中国很多人又叫它组织芯片，组织微阵列一般是指将数十至上千个小组织整齐地排放在一张载玻片上而制成的组织切片[6][7]。

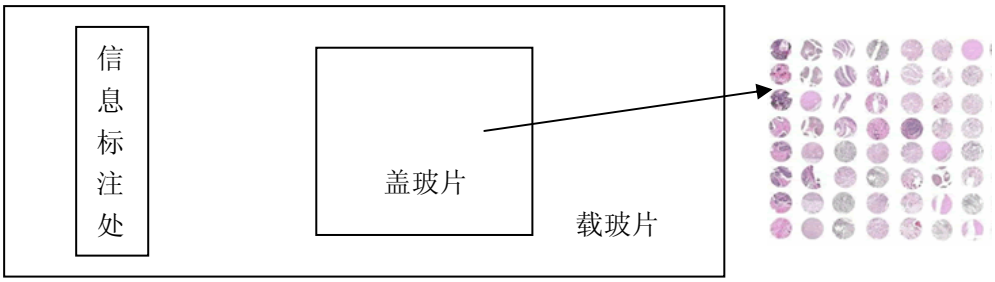


图 2.1 玻片式组织微阵列示意图

同一套组织微阵列便可对上百种生物分子标记（如抗原，DNA 和 RNA）进行分析、检测，因而倍受组织病理学家的重视[8]。作为芯片技术的新成员，组织微阵列具有经济、简便快捷的特点，特别是将分子生物学和组织形态学结合的优势，满足基础研究和临床研究工作者的需要，具有广泛的应用前景。与传统组织病理技术比较，具有信息量大、体积小的特点，是传统技术的革新[6]。已经在生物学和医学中有了应用，而且取得了很大成绩。在临床教学上更显示了它独特的优势。

2.2 国内外研究概述

组织微阵列，最早是由 Wan，Fortuna 和 Furmanski 于 1987 年在 Journal of Immunological Methods 做过描述[9]，而真正实现并且普及则是美国国家人类基因实验室 Kononen 教授和他的同事们在 1998 年[7]完成的，并进行了全面报道。

因为国内较早的文章称它为“组织芯片”，所以后来的研究者在发表文章时仍把“组织芯片”作为关键词，本节也将暂时用这一名字进行文献检索。从维普中文科技期刊全文数据库中键入“组织芯片”进行搜索，最早的文章是在 2001 年发表的，其中《华西医学》2 篇，《中国科学基金》和《癌症》各 1 篇，共 4 篇，在其中一篇中有文字显示该文作者于 1999 年便开始这一技术的研究而且设计了一种制作组织微阵列的器具并获国家专利[2]。

总的来说，在中国，组织微阵列的研究仅算的上刚刚起步。本人在维普中文科技期刊全文数据库上的检索情况：2000 年（含）以前文章 0（0）篇，2001 年文章 4（3）篇， 2002 年文章 30（25）篇，2003 年文章 46（38）篇，2004 年文章 71（51）篇，2005 年文章 116（73）篇。如图 2.2 所示。以上数据中带括号的是直接介绍组织微阵列或其应用的文章数，括号外的数据是总的有关组织微阵列的文章数。

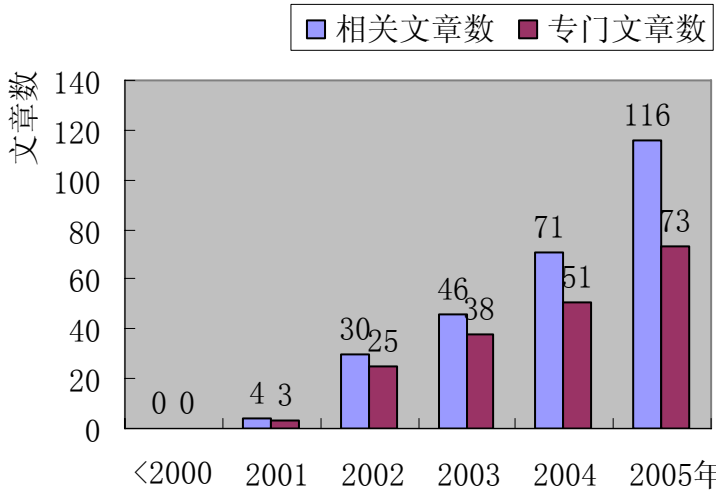


图 2.2 国内主要期刊有关组织微阵列文章的统计

从数据中可以看出，自 2001 年首次有人发表该类文章后，每年的发表数量增长迅速。也就是说，在中国越来越多的人开始关注和研究组织微阵列。作为应用于临床的新技术，目前国内在组织微阵列研究方面的文章大致分三个方面：对概念的介绍、制作方法上某个环节的控制、应用。

从 1998 年它的诞生到现在快 8 个年头了，我国国内介绍概念的文章还是有不少。而应用方面则显得较窄，几乎集中在肿瘤方面的诊断探讨，把它当作集成了的大量病理切片用，其他一些则主要是它在新地方上的尝试，微阵列的量上也需求不大。据一篇文章显示，作者在中国医院知识仓库检索中对 2001 年－2004 年的 101 篇有关组织微阵列的文章进行分析得出，与肿瘤学相关的有 79 篇，占 78.2%[53]。至于它的制作，几乎每篇相关文章中均有一点介绍，专门的文章也是特别多，但是都不够详细，一般是对某个工艺的小改进。针对这些，本章以下内容将对组织微阵列作详细介绍。

国外的情况，从 elsevier 中搜索最新（2006 年）的文章看，应用的文章居多，如和 DNA、mRNA 结合的研究[54][58]，癌症诊断[55]中的应用以及其他病理学[56][57]中的应用等。

2.3 组织微阵列的特点及应用范围

2.3.1 特点

1. 容纳信息量大，可在短时间内高通量（high-throughput）[38]、快速、平行检测

大批量组织样本中多个实验指标，大大提高实验效率；

2. 一套组织微阵列可同时进行多个样本多个指标比较，减少一系列复杂因素所致的组内和组间差异，使多样本比较分析的准确性大为提高；

3. 能够做到所取组织定位准确，无效组织数量减少，同时对原始组织样本的蜡块破坏小；

4. 制作中多个过程实现自动化 ，可人为控制各样本组织排列位置，便于设置各种对照；

5. 需在低温、阴暗处保存，不能与有机溶剂混放。

2.3.2 应用范围

1. 在免疫组化中的运用，可用于常规临床病理的免疫组化检测，获得更多信息，提高阳性检测率；

2. 在 FISH 中应用；

3. 和基因微阵列技术结合运用于肿瘤生物学研究；

4. 测试生物试剂；

5. 教学工作中的应用：正常组织微阵列形成微组织学图谱可用于组织胚胎学教学，各种类型肿瘤微阵列形成病理图谱可提高医生病理诊断水平，还可用于研究人和动物的发育、分化、各种疾病等；

6. 有利于形态学的比较研究：可将各种不同的组织器官集中在一个组织芯片上，医学类学生可以用它进行比较学习；

7. 用于各种免疫组织化学染色、原位杂交、原位 PCR、荧光原位杂交、原位 RT-PCR 和寡核苷酸启动的 DNA 合成（PRINS）等等；

8. 用于临床和基础的研究，分子诊断、预后指标筛选、治疗靶点定位、抗体和药物筛选、基因和表达分析等。

两个应用组织微阵列的典型例子：

Bubendorf 等人[7][36]应用 FISH 技术检测了 371 例前列腺组织样本（32 例良性增生、223 例原发癌、54 例切除组织、62 例转移癌组织）中 AR、 MYC 、Cyclin-D_1、 $ERBB_2$ 和 NMYC 等五种基因扩增水平。结果发现在早期前列腺癌中上述 5 种基因出现高水平扩增者较少（<2%）。而在激素不敏感者转移的癌组织中 AR 出现高水平扩增者为 22%，MYC 出现高水平扩增者为 11%，Cyclin-D_1 出现高水平扩增者为 5%，而 $ERBB_2$ 和 NMYC 在进展期前列腺癌中则未扩增出。这种组织微阵列的 FISH 法对检测基因扩增是一个高产出量的分析，同时，组织微阵列法使 Bubendorf 检测了前列腺癌的发生、发展过程中激素受体的扩增情况，即将增生——良性肿瘤——恶性转换——转移、复发等不同时期病变阵列在一个

组织块上进行 FISH 检测基因扩增，有助于全面分析肿瘤发展过程中癌基因的遗传学改变。

Mucct（本例为湘潭华鉴科技有限公司提供）从 12 例前列腺尸体解剖的肿瘤组织标本中选出 100 张常规切片，制成两张组织微阵列，比较了低分化前列腺癌中神经内分泌因子的 CGA 和 SYG 的表达。结果发现：常规切片仅发现 13%的病例看到局灶性表达。这一方法和结果证明了组织芯片的优势，一张芯片可获得更多的信息，提高阳性检测率，更能反映真实情况。

由于笔者对医学知识不够熟悉，以上成果只能是总结[22-37]了很多相关工作者的应用，当然了，这远非它的全部，正如前边所讲，组织微阵列信息量庞大，有着惊人的应用前景。

2.4 组织微阵列制作

1. 组织样本收集：

以收集恶性肿瘤标本为例，一般是医院病理科取材后剩下的全部标本，要求保持新鲜，以免产生蛋白质分解变性，导致细胞自溶以及细菌滋生，不能反映组织活体时的形态结构，并有病理切片申请单的资料和医院的病理报告结果，标本必须带有癌旁组织和周围的正常组织，以便后期在组织微阵列的制作中加以区分。

2. 组织样本的处理：

（1）固定 用适当的化学药液——固定液浸渍切成小块的新鲜组织样本，迅速凝固或沉淀细胞和组织中的物质成分、终止细胞的一切代谢过程、防止细胞自溶或组织变化，尽可能保持其活体时的结构。固定能使组织硬化，有利于后期切片的进行，而且也有媒浸作用，有利于组织着色。固定液的种类很多，其对组织的硬化收缩程度以及组织内蛋白质、脂肪、糖类等物质的作用各不相同。例如纯酒精可固定肝糖而能溶解脂肪，甲醛能固定一般组织，但溶解肝糖和色素。固定液可分为单一固定液及混合固定液。前者有甲醛（蚁醛、福尔马林）、酒精、醋酸或冰醋酸、升汞、锇酸（四氧化锇）、重铬酸钾及苦味酸等，单一固定液不能固定细胞中的所有成分；混合固定液可以互补不足，常用的混合固定液有 Bouin 氏液、Zenker 氏液、FAA 液、Carnoy 氏液、SuSa 液。因此，应根据所要显示的内容来选择适宜的固定液。10％福尔马林（4％甲醛）或 10％磷酸缓冲福尔马林是病理切片常规使用的固定液，不仅适用于常规 HE（苏木精-伊红）染色，还可以用于组织学有关的其他技术的切片染色。固定液的用量通常为材料块的 20 倍左右，固定时间则根据材料块的大小及松密程度以及固定液的穿透速度而定，可

以从 1 小时至数天，通常为数小时至 24 小时。

（2）洗涤与脱水 固定后的组织材料需除去留在组织内的固定液及其结晶沉淀，否则会影响以后的染色效果。多数情况下是采用流水冲洗，使用含有苦味酸的固定液固定的则需用酒精多次浸洗，如果组织经酒精或酒精混合液固定，则不必洗涤，可直接进行脱水。固定后或洗涤后的组织内充满水分，如不除去水分就无法进行以后的透明、浸蜡与包埋，因为透明剂多数是苯类，苯类和石蜡均不能与水相融合，水分不脱尽，苯类不能浸入。酒精为常用脱水剂，它既能与水相混合，又能与透明剂相混，为了减少组织材料的急剧收缩，应使用从低浓度到高浓度递增的顺序进行，通常从 30％或 50％酒精开始，经 70％、85％、95％直至纯酒精（无水乙醇），每次时间为 1～数小时，如不能及时进行各级脱水，材料可以放在 70％酒精中保存，因高浓度酒精易使组织收缩硬化，不宜处理过久。正丁醇、叔丁醇、丙酮及二氧陆环等也可做脱水剂。

（3）透明 纯酒精不能与石蜡相溶，还需用能与酒精和石蜡相溶的媒浸液，替换出组织内的酒精。材料块在这类媒浸液中浸渍，出现透明状态，此液即为透明剂，透明剂浸渍过程称透明。常用的透明剂有二甲苯、苯、氯仿、正丁醇等，各种透明剂均是石蜡的溶剂。通常组织样本先经纯酒精和透明剂各半的混合液浸渍 1～2 小时，再转入纯透明剂中浸渍。透明剂的浸渍时间则要根据组织材料块大小及属于囊腔抑或实质器官而定。如果透明时间过短，则透明不彻底，石蜡难于浸入组织；透明时间过长，则组织硬化变脆，就不易切出完整切片，最长为数小时。

（4）浸蜡 用石蜡取代透明剂，使石蜡浸入组织而起支持作用。通常先把组织材料块放在熔化的石蜡和二甲苯的等量混合液浸渍 1～2 小时，再先后移入 2 个熔化的石蜡液中浸渍 3 小时左右，浸蜡应在高于石蜡熔点 3℃左右的恒温箱中进行，以利石蜡浸入组织内。

3. 组织样本的包埋

浸蜡后的组织材料块放在装有蜡液的容器中（摆好在蜡中的位置），待蜡液表层凝固即迅速放入冷水中冷却，即做成含有组织块的蜡块。容器可用光亮且厚的纸折叠成纸盒或用金属包埋框盒。如果包埋的组织块数量多，应进行编号，以免差错。石蜡熔化后应在蜡箱内过滤后使用，以免因含杂质而影响切片质量，且可能损伤切片刀，造成切片不完整。通常石蜡采用熔点为 56～58℃或 60～62℃两种，可根据季节及操作环境温度来选用。

为了达到好的效果，也有人经常采用一种比较麻烦的包埋法[47]，即在包埋容器里灌入少许石蜡，待石蜡冷凝后，用烧热的镊子将包埋位置的石蜡熔化，再把组织埋入，如此反复操作，等组织块全部包埋完后，再将容器放在温度高的界面

上，等石蜡渐熔时，将石蜡加满。用此方法包埋的蜡块，石蜡与组织充分融合，各组织块基本在一个平面上，切片时不会出现组织不全或组织块崩出现象，而且可以在一个包埋容器里埋入较多的组织块。

4. 蜡块的切片

包埋好的蜡块用刀片修成规整的方形或长方形，以少许热蜡液将其底部迅速贴附于小木块或塑料等材料制成的托体上，夹在轮转式切片机的蜡块钳内，使蜡块切面与切片刀刃平行，旋紧。切片刀的锐利与否、蜡块硬度都直接影响切片质量，可用热水或冷水等方法适当改变蜡块硬度，因为组织差异大，对刀片要求特别高，所以要经常更换刀片（每个刀片切约 50 张）[50][52]。通常切片厚度为 4～7 微米，切出一片片的蜡片，将完整性好的用毛笔轻托轻放在纸上。

5. 贴片与烤片

用粘附剂（医院用的粘附剂是蛋白甘油）将展平的蜡片牢附于载玻片上，以免在以后的脱蜡、水化及染色等步骤中造成滑脱。首先在洁净的载玻片上涂抹薄层蛋白甘油，再将单个蜡片轻轻放入温水（40℃～45℃左右[40][41][42]）中，利用水的张力将它展平，再捞至玻片上铺正，简单起见，可以直接滴两滴蒸馏水于载玻片上，再把蜡片放于水滴上，略加温，使蜡片铺展，最后用滤纸吸除多余水分，将载玻片放入 45℃温箱中干燥，也可在 37℃温箱中干燥，但需适当延长时间。

6. 切片脱蜡及水化

干燥后的切片需脱蜡及水化才能在水溶性染液中进行染色。一般是用二甲苯脱蜡，再逐级经过纯酒精、各梯度酒精直至蒸馏水。实际操作中，就是把一些有盖大广口瓶装上各溶液，按顺序放好，让玻片逐个经它们浸泡。如果染料配制于酒精中，则将切片移至与该酒精近似浓度时，不必再继续移，即可进行染色。

7. 染色

染色的目的是使细胞组织内的不同结构呈现不同的颜色以便于观察。未经染色的细胞组织的折光率相似，不易辨认。经染色，可显示细胞内不同的细胞器、内含物以及不同类型的细胞组织。染色剂种类繁多，应根据观察要求及研究内容采用不同的染色剂及染色方法，还要注意选用适宜的固定剂才能取得满意的结果。经典的苏木精（Hematoxylin）和伊红（曙红，Eosin）染色法是组织学标本及病理切片标本的常规染色，简称 HE 染色。经 HE 染色后，细胞核被苏木精染成紫蓝色，多数细胞质及非细胞成分被伊红染成粉红色。由于苏木精是带阳离子的染料，染液呈碱性，核内染色质及胞质内核糖体等嗜碱性物质对这种染料有亲和性，而带阴离子的染料伊红配制的染液呈酸性，可以对嗜酸性物质进行染色。有时不同的组织结构还需要用特殊的染料及染色方法加以显示，称特殊染色。有些细胞组织经硝酸银浸润后，可使溶液中银离子还原成金属银或银粒附着在细胞

组织上，呈棕黑色，这种性质称亲银性，而有些细胞组织本身不能使硝酸银的银离子还原成金属银，还需加还原剂才能将银离子还原，称嗜银性。不同的研究目的下，我们可以根据组织的特性选用不同的方法将它们进行染色。注意，要根据染色方法、组织类别及切片厚度，正确选择染色时间，这样才能达到较好的染色效果。

8. 切片脱水、透明和封片

染色后的切片尚不能在显微镜下观察，需要再进行脱水、透明和封片处理，脱水和透明过程与组织样本的处理几乎一样。这里，切片也需经梯度酒精脱水，在 95%及纯酒精中的时间可适当加长以保证脱水彻底，但是如果染液为酒精配制，则应缩短在酒精中的时间，以免脱色。经二甲苯透明后，迅速擦去材料周围多余液体，滴加适量（1～2 滴）中性树胶，再将洁净盖玻片倾斜放下（防止出现气泡），即封片。封片后即制成永久性玻片标本，可以在光镜下可长期反复观察，此玻片在简单条件下便可实现长期保存。

9. 制成组织微阵列

病理专家在显微镜下观察已经做好的玻片标本，以此在样本蜡块上圈定制作微阵列所用的区域。制作人员用空心管在圈定区域进行取材，取出的柱状标本编号备用。用步骤3中提到的包埋容器制成空白蜡块，然后用组织微阵列制作仪钻出直径为1毫米（目前取样直径有2.0mm、1.5mm、1.0mm、0.6mm四种[43]）间距约2毫米（或2倍孔直径）[46]的孔。因为该仪器价格特别昂贵，尚有很多人沿用直径1毫米的钻头打孔[45]，缺点是容易打偏甚至打“豁”。打孔后在空白蜡块上形成一个“孔阵列”。把在不同标本上取下的很多个柱状标本按预先确定的位置有序地插入对应的孔中，将缝隙灌蜡，加热，以使它们充分融合。这样就制成了微阵列样本蜡块。最后进行以上4～8的步骤，就制成了玻片式组织微阵列。

在制作组织微阵列中，除了上边介绍的石蜡切片法，还有冷冻切片法，但制成的效果不是很好，只是时间快些而已，随着制片技术的发展，现在用冷冻切片法的人慢慢减少。

近些年来，在病理常规制片过程中采用了微波技术，从而大大缩短了制片过程，而且对形态结构并没有影响。组织经微波（波长为 1 米～1 毫米，频率为 300 兆赫～300 千兆赫）辐射后加速组织内部分子的高速运动，以使液体的运输加快，增加弥散、渗透和交换效率，从而加速组织的固定、脱水、透明、包埋和染色各个环节。例如常规福尔马林固定需数小时～1 天，而且能造成组织收缩及对某些抗原成分不同程度的破坏，微波固定仅需 1～2 分钟，且可减少抗原的丢失和损害。选择适当的功率、辐射时间和温度是极为重要的。目前微波技术在制片上的应用在国内尚处于起步阶段，许多技术应用环节尚需进一步摸索。

2.5 本章小结

本章从生物芯片谈起，谈到组织微阵列的概念，并且总结了大量文献后找出文献反映出来的问题加以解决，即以概述的形式让人一目了然地看到组织微阵列到底是什么，能够做些什么事情，后边则详细地介绍了它的制作过程及制作时每步要注意的东西。

第三章 聚类分析研究

组织微阵列信息量庞大，单纯靠人为一个点阵一个点阵的去看，去计算，去记录，去分析，那将是多大的工作量。随着计算机和数学等学科的发展，我们完全可以利用这些工具替我们分担很多繁重甚至枯燥的工作。通过对很多研究组织微阵列的人进行函调或其他形式（如直接对话和分析他们发表的文章等）的考察后，我得到个结论——信息量庞大，严重影响应用范围。何不把这庞大的信息按我们的具体需要分类，再做分析呢？

其实，上世纪 40 年代就有人研究怎样将庞大的信息分类这一数学问题，即后来的聚类分析。我国在 70 年代出现这一概念，最早是中科院地质研究所为了解决矿物标本分类引入的，到 70 年代末得到了推广，逐渐涉足生物学、医学、气象学、经济学、社会学等需要做分类的很多学科中。可见它的优势所在了。对于生物微阵列的研究，聚类分析已经被引入基因微阵列的分类中，把普通人难以想象的分析简单化，大大提高了研究效率。然而，对于刚刚起步的组织微阵列来说，目前国内还没有文章显示引入聚类分析，但是根据它的应用前景，当前对组织微阵列进行聚类分析研究将显得尤为重要，要尽快填补这一空缺。

3.1 聚类分析概念

“人以群分，物以类聚”，聚类分析就是研究如何把物进行分类的学科。因为它用到的是数学方法，而且往往是解决大量有模糊差异的事物的分类问题。起源于分类学，又称群分析，它是研究（样品或指标）分类问题的一种多元统计方法。所以可以将其定义为：运用数学的方法，根据指标间的相关性或者是样品间的相似性，研究类的划分以及类与类之间亲疏程度的工具。在本文，它将作为按组织微阵列各种反映出来的信息为指标将微阵列进行归类的工具。

表示聚类结果的图示方法有很多，如下图 3.1，是一种很直观而且看似简单的表示。

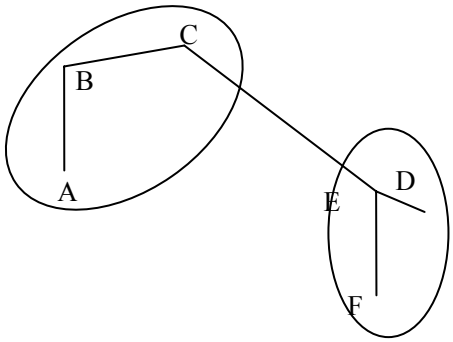


图 3.1

很明显，如果把上图 3.1 中 A～F 这 6 个样本分成两类，假设点与点之间的距离就是它们接近程度，一眼就看出 A、B、C 一类，D、E、F 一类。但是遇到样本比较多，而且按多个相关系数或距离进行分类时，就不那么直观了。在本论文中将一般采用如下图 3.2 所示的表示方式。可以很明显的看到，如果我们想知道以某一距离/相关度将一些样品进行聚类，直接在对应坐标上画一垂直与坐标轴的直线，这样，被切下来的左半边就直观的显示出其聚类来了。

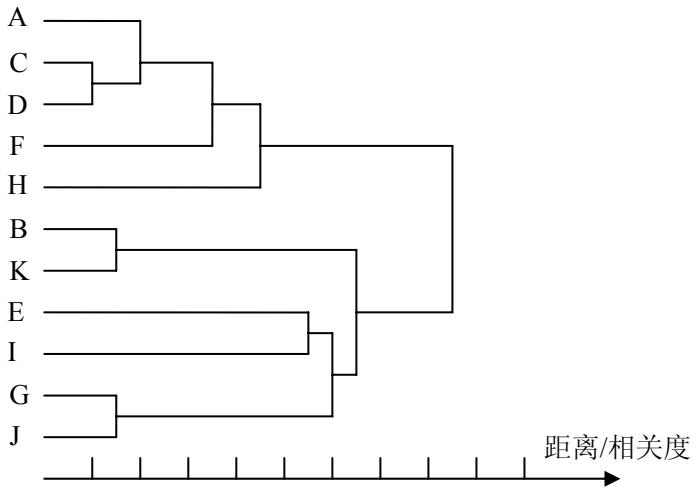


图 3.2

3.2 聚类分析的方法

从前边的知识我们已经知道聚类分析就是要把很多的变量（样本）按照它们性质上的亲疏远近的程度进行分类。到底怎样来描述它们的亲疏程度，通常有两个途径：其一是把每个样品看成 m 维空间的一个点，这样点和点之间就存在某种

“距离”，我们想法求出这个距离就得到了点与点间的亲疏程度；其二，定义某种相似系数来表示样品间的亲疏程度。有了这个基础，就可以进行聚类了。聚类的方法有很多，大致[10]归类为以下一些：

1. 聚合法（joining）：首先把每个样品各做一类，然后将距离最近的类合并，使类逐渐减少，直至变成一类。
2. 分裂法（splitting）：与聚合法相反，它是先把全体样品看成一类，逐渐按一规则进行分裂，达到分类目的。
3. 调优法(switching)：这种方法是将样本大致分类，然后根据分类函数尽可能小原则对分类进行调整。
4. 加入法（adding）：将新样本添加到已有分类中，确定其最合适的位置，形成新类，然后再挨个添加，完成分类。
5. 最优分段法：有的时候样品次序是不允许打乱的，如特定情况下组织微阵列的分类，上边的点阵是按特定顺序分布的，做定向研究用的。这种分类不常用，相对来说也简单，可以将其他方法加以条件限制即可。

除此以外，还有一些方法在聚类分析中也有时见到，如图论法、预报法、变量筛选法等。这里不一一介绍。

根据不同的用途，聚类分析逐渐出现了一些成熟的算法：系统聚类分析、逐步聚类分析、自组织图分析、Bayesian 聚类分析、二向聚类分析、主要成分分析、多维度分析等[14-21]。但归根到底，其实就是区分到底用哪种方法求得样本间的“距离”。在生物学中，因为研究的信息量很大，很模糊，系统聚类分析用的比较多而且有效。本论文也是重点用这种方法进行组织微阵列的聚类分析研究的。

3.3 距离

聚类分析中的距离是个广义的表示样本间相似（接近）程度的量。假设有 n 个样品，每个样品测得 p 项指标（变量），则有资料阵为

$$X=\begin{array}{c} \\ X_1 \\ X_2 \\ \vdots \\ X_n \end{array}\begin{array}{c} \begin{array}{cccc} x_1 & x_2 & \cdots & x_p \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \end{array}$$

其中 $x_{ij}(i=1,\cdots,n;j=1,\cdots,p)$ 为第 i 个样品的第 j 个指标的观测数据。第 i 个样品 X_i 为矩阵 X 的第 i 行所描述，这样任何两个样品 X_k 与 X_l 之间的相似程度便可以通过对比矩阵 X 中的第 K 行与第 L 行的的资料来进行描述。如果把 x_{ij} 以数

的形式体现出来，我们便可以假设这 n 个样本为 p 维坐标中的点，这样样本间也就可以算出距离了。设 p 维空间的两个点 x_i 和 x_j，定义它们的距离为 d_{ij}，则它应该满足以下四个条件：

（1） $d_{ij}=0$ 等价于 x_i 与 x_j 相等（即两点重合）；

（2） $d_{ij}\geqslant 0$ 对所有点成立（即 $\forall i, j$）；

（3） $d_{ij}=d_{ji}$ 对一切 i，j 都成立；

（4） 符合三角不等式，即三角形两边和大于第三边：$d_{ij}\leq d_{ik}+d_{kj}$。

但是在实际聚类分析中，为了增强某些对比，有时候不利用条件（4），取而代之的是：（4）’：$d_{ij}\leqslant \max\{d_{ik}, d_{kj}\}$，其实 $d_{ij}\leqslant \max\{d_{ik}, d_{kj}\}\leqslant d_{ik}+d_{kj}$，仍是满足条件（4）的，只是条件性更强些，这个条件下的距离叫做极端距离。

3.3.1 明氏距离

满足条件（1）—（4），最常用的有距离公式：

$$d_{ij}(q)=\left[\sum_{k=1}^{m}\left|x_{ik}-x_{jk}\right|^{q}\right]^{1/q} \tag{3.1}$$

此公式计算出来的距离叫明考斯基（Minkowski）距离，简称明氏距离。

当 q＝1 时，称为绝对值距离或曼哈坦（Manhattan）距离：

$$d_{ij}(1)=\sum_{k=1}^{m}\left|x_{ik}-x_{jk}\right| \tag{3.2}$$

当 q＝2 时，称为欧几里德距离，简称欧氏距离：

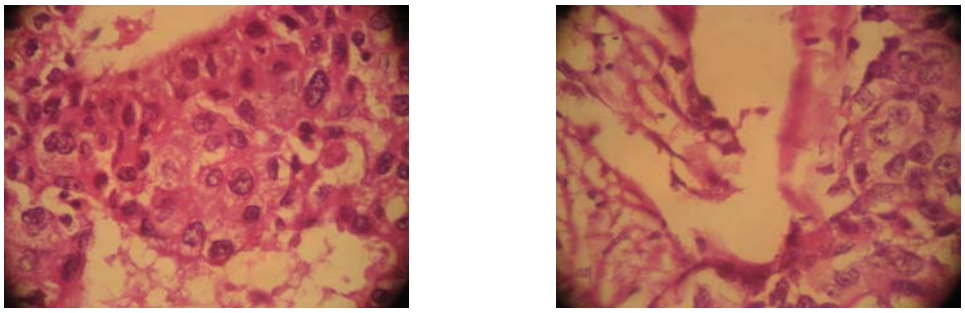
$$d_{ij}(2)=\left[\sum_{k=1}^{m}\left|x_{ik}-x_{jk}\right|^{2}\right]^{1/2} \tag{3.3}$$

当 q 趋于无穷（∞）时，称为切比雪夫距离：

$$d_{ij}(\infty)=\max_{1\leq k\leq m}\left|x_{ik}-x_{jk}\right| \tag{3.4}$$

以上三种距离表达是明氏距离最常用的，特别是欧氏距离，算出来的距离就是我们习惯上所说的距离，实际效果也是最好的，所以使用最多，针对图像处理

和识辨，也是被很多人推崇。本论文中后边的例子使用到的距离也就是欧氏距离。到底怎么样算出组织微阵列间的距离呢？假设我们以图像亮度的直方图相似程度来反映微阵列图片间的相似程度做聚类分析。



（1）　　（2）

图 3.3 乳腺癌某两阵列 HE 染色图（1）和（2）（放大倍数 40*10）

（1’）　　（2’）

图 3.4 两阵列图对应的亮度直方图（1’）（2’）

为了表示上图中两亮度分布图的相似程度，即求出它们间的距离，可以在这个直方图上的每个色阶统计像素数量，将两图假设为 256 维坐标上的两个点，各像素数量即为对应维上的坐标值，这样它们间就存在距离了。根据一位组织微阵列研究人员的要求，他只想知道某一范围内所占比重的情况，这样一来，问题又可以简单化了。由于没有更详细的要求，就先根据图像的大体分布情况将其化为 13 段，即假设一个 13 维坐标系。列表如下：

表 3.1 亮度图分段统计像素个数表

分段 / 图像	0～50	51～70	71～90	91～110	111～120	121～130	131～140	141～150	151～160	161～170	171～180	181～190	191～225	总计

1	85747	168932	401844	729254	372197	312830	248849	214828	222716	299575	87745	1202	9	3145728
2	62662	113455	376520	625138	346483	353398	307723	293881	422728	231612	12040	88	0	3145728

从表中可以看到，数字很大，不便于直接比较，所以也可以转化成其他形式，如所占百分比：

表 3.2 亮度图分段像素量百分比表（单位％）

分段 图像	0～50	51～70	71～90	91～110	111～120	121～130	131～140	141～150	151～160	161～170	171～180	181～190	191～225	总计
1	2.73	5.37	12.77	23.18	11.83	9.94	7.91	6.83	7.08	9.52	2.79	0.04	0	99.99
2	1.99	3.61	11.97	19.87	11.01	11.23	9.78	9.34	13.44	7.06	0.38	0	0	99.68

这样表示，数据直观了很多，但是，可以从它的总计中看出，由于在计算中涉及到小数位数保留问题，致使总计总是不可避免的与 100％有一些误差。

利用公式 3.3 计算 1、2 两图的欧氏距离：表 3.1 中为：278336（小数部分 4 舍 5 入忽略）；表 3.2 中为：8.9267（小数部分 4 舍 5 入精确 4 位）。

3.3.2 马氏距离

从前边的介绍中可以看出，明氏距离计算上直观简单，但是，我们做聚类分析的时候，选取的变量往往有很大的相关性，这在明氏距离公式中却体现不出来。前边所举例是把亮度图分布做的变量，如果按别的要求，可能每个变量又同时反映着多个特征，各个特征又占着不同的比重，那用明氏距离就不好解决了。

针对这个缺点又产生了马氏（P.C.Mahalanobis）距离，马氏距离是由印度统计学家马哈拉诺比斯于 1936 年引入的，故称为马氏距离。这一距离在多元统计分析中起着十分重要的作用，同样满足前边的（1）－（4）条件。设Σ表示指标的协差阵即：

$$\Sigma=(\sigma_{ij})_{p\times p}$$

其中 $\sigma_{ij}=\frac{1}{n-1}\sum_{a=1}^{n}(x_{ai}-\bar{x}_i)(x_{aj}-\bar{x}_j)$　i, j = 1, ⋯, p

$$\bar{x}_i=\frac{1}{n}\sum_{a=1}^{n}x_{ai}\qquad \bar{x}_j=\frac{1}{n}\sum_{a=1}^{n}x_{aj}$$

如果 Σ^{-1} 存在，则两个样品之间的马氏距离为

$$d_{ij}^2(M)=(X_i-X_j)'\Sigma^{-1}(X_i-X_j) \tag{3.5}$$

这里 X_i 为样品 X_i 的 p 个指标组成的向量，即原始资料阵的第 i 行向量。样品 X_j 类似。

样品 X 到总体 G 的马氏距离定义为

$$d^2(X,G)=(X-\mu)'\Sigma^{-1}(X-\mu)$$

其中 μ 为总体的均值向量，Σ为协方差阵。

马氏距离既排除了各指标之间相关性的干扰，而且还不受各指标量纲的影响。除此之外，它还有一些优点，如可以证明，将原数据作一线性交换后，马氏距离仍不变等等。

但是，在聚类分析中，在未形成类以前，如果用全部数据计算的均值和协差阵来求马氏距离，效果却不是很好。“比较”合理的办法是用各个类的样品来计算各自的协差阵，同类样品的马氏距离应当用这一类的协差阵来计算，但类的形成要依赖于样品间的距离，而样品间的合理马氏距离又依赖于类，这样一来就形成了一个恶性循环[10]。一些分析表明，把数据处理后做欧氏距离计算得出的结果会好些[11]。可见，马氏距离的应用还是需要条件的，可以作为明氏距离的辅助手段解决一些明氏距离不好解决的问题，让它们发挥各自的特长才能更有效的进行聚类分析研究。

3.3.3 兰氏距离

该距离是由 Lance 和 Williams 最早提出的，故称兰氏距离。

$$d_{ij}(L)=\frac{1}{p}\sum_{a=1}^{p}\frac{|x_{ia}-x_{ja}|}{x_{ia}+x_{ja}}\qquad i,j=1,\cdots,n \tag{3.6}$$

此距离仅适用于 $x_{ij}>0$ 的情况，这个距离有助于克服各指标之间量纲的影响，但没有考虑指标之间的相关性。

计算任何两个样品 X_i 与 X_j 之间的距离 d_{ij}，其值越小表示两个样品接近程度越大，d_{ij} 值越大表示两个样品接近程度越小。如果把任何两个样品的距离都算出来后，可排成距离阵 D：

$$D=\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & & & \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}$$

其中 $d_{11}=d_{22}=\cdots=d_{nn}=0$。D 是一个实对称阵，所以只须计算上三角形部分或下三角形部分即可。根据 D 可对 n 个点进行分类，距离近的点归为一类，距离远的点归为不同的类。

以上三种距离的定义是适用于间隔尺度变量的，其实，明氏距离在大多情况下会简单而且有效，有的时候，我们可以对数据进行标准化，以克服量纲等的影响。标准化的方法有两个：

（1） 标准差标准化

$$x'_{ij}=\frac{x_{ij}-\bar{x}_j}{s_j} \qquad i=1,2,\cdots,n;\quad j=1,2,\cdots,m \qquad (3.7)$$

其中 $s_j=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2}$

这样一来，每个变量的子样标准差都化为 1 了，标准化的数据{x'ij}就与变量的量纲没关系了。

（2） 极差标准化

$$x''_{ij}=\frac{x_{ij}-\bar{x}_j}{R_j} \qquad i=1,2,\cdots,n;\quad j=1,2,\cdots,m \qquad (3.8)$$

其中 $R_j=\max_{1\le i\le n}\{x_{ij}\}-\min_{1\le i\le n}\{x_{ij}\}$

同样，这里的每个变量的子样极差都化为 1 了，{x''ij}也就与变量的量纲无

关了。

3.4 相似系数

相似系数——就是描述变量间相似程度的量。设有变量 X_i 和 X_j，用 c_{ij} 表示它们的相似系数，则有以下要求：

（1）$c_{ij}=\pm 1$ （=） $X_i=aX_j$，a 为常数，且 $a\neq 0$；

（2）$|c_{ij}|\leqslant 1$ 一切 i，j；

（3）$c_{ij}=c_{ji}$ 一切 i，j。

$|c_{ij}|$ 越接近 1，说明它们相互关系越密切；如果接近 0，说明它们关系很疏远。

常用的相关系数[10]有：

1. 夹角余弦 忽略变量各个绝对长度，单从形状上判断相似性。如两个正方形，虽然大小不一，但仍认为一样。定义为向量（x_{1i}，x_{2i}，…，x_{ni}）和（x_{1j}，x_{2j}，…，x_{nj}）之间的夹角余弦：

$$c_{ij}=\frac{\sum_{k=1}^{n}x_{ki}x_{kj}}{\sqrt{(\sum_{k=1}^{n}x_{ki}^2)(\sum_{k=1}^{n}x_{kj}^2)}} \tag{3.9}$$

2. 相关系数 是标准化了的夹角余弦。

$$c_{ij}=\frac{\sum_{k=1}^{n}(x_{ki}-\bar{x}_i)(x_{kj}-\bar{x}_j)}{\sqrt{\sum_{k=1}^{n}(x_{ki}-\bar{x}_i)^2\sum_{k=1}^{n}(x_{kj}-\bar{x}_j)^2}} \tag{3.10}$$

3. 指数相似系数

$$c_{ij}=\frac{1}{m}\sum_{k=1}^{m}e^{-[\frac{3(x_{ik}-x_{jk})^2}{4\,s_k^2}]} \tag{3.11}$$

显然 $1\geqslant c_{ij}\geqslant 0$，只有当 $x_{ik}=x_{jk}$（k=1，2，…，m）时，即相等时，它才为 1。

4. 非参数方法 令 $x'_{ij} = x_{ij} - \bar{x}_j$，（i＝1，2，…，n； j＝1，2，…，m）

$\delta(x) = \begin{cases} 1 & \text{若} x > 0 \\ 0 & \text{若} x \le 0 \end{cases}$，记

$$n_+ = \sum_{k=1}^{m} \delta(x'_{ik}\ x'_{jk})$$

$$n_- = \sum_{k=1}^{m} \delta(-x'_{ik}\ x'_{jk})$$

即 n_+ 为 x'_{ik} 与 x'_{jk} 符号相同的个数，n_- 为 x'_{ik} 与 x'_{jk} 符号相异的个数。则有相似系数：

$$c_{ij} = \frac{n_+ - n_-}{n_+ + n_-} \tag{3.12}$$

显然 $|c_{ij}| \leqslant 1$。若 $|c_{ij}| = 1$，必有 $n_+ = 0$ 或 $n_- = 0$，这时说明两个样本的变化趋势完全一致或完全相反；若 $|c_{ij}| = 0$，必有 $n_+ = n_-$，表明两者的变化没有必然的联系。

5. 当$\{x_{ij}\}$非负时常用以下相似系数

$$c_{ij} = \frac{\sum_{k=1}^{m} \min(x_{ik}, x_{jk})}{\sum_{k=1}^{m} \max(x_{ik}, x_{jk})} \tag{3.13}$$

$$c_{ij} = \frac{\sum_{k=1}^{m} \min(x_{ik}, x_{jk})}{\frac{1}{2}\sum_{k=1}^{m} (x_{ik} + x_{jk})} \tag{3.14}$$

$$c_{ij}=\frac{\sum_{k=1}^{m}\min(x_{ik},x_{jk})}{\sum_{k=1}^{m}\sqrt{x_{ik}x_{jk}}} \tag{3.15}$$

除此之外还有很多类型的相似系数，如联列系数、连关系数、点相关系数、四分相关系数、改进的夹角余弦等，这里暂不详细介绍。在实际问题中，去有针对性地选择最有效，最简单的即可。

3.5 组织微阵列图片的系统聚类

系统聚类（Hierachical Clustering），是将很多样本一层一层归类，直到最后归成一个类。假设有 n 张组织微阵列图片，现在要做系统聚类，步骤如下：

1. 开始聚类前，每张图片自成一类，即有 n 个类；
2. 根据实际要求，选出合适的量，算出这 n 个类两两间的距离，找出距离最小（即最相似）的两个类，把它们合并，组成一个新类，这时就剩下 n-1 个类了；
3. 由上步得到的新类按照某个（后边详细介绍）规律确定对应的量，确定它与其余各类的距离，重复步骤 2，再次找出距离最小的类合并，以此循环，直到把它们最后归成一类为止。

从以上步骤中可以看出，系统聚类无非是一个类与类的距离计算问题。类与类之间用不同的方法定义距离，就产生了不同的系统聚类方法。常用的系统聚类方法有八种，即最短距离法、最长距离法、中间距离法、重心法、类平均法、可变类平均法、可变法、离差平方和法。下边将以不同距离的计算方法讨论组织微阵列的系统聚类。

3.5.1 最短距离法

本章第三节的距离介绍中，以微阵列亮度图像各段情况为例，算出了两种表示下的欧氏距离，也看出了直接用像素量计算出来的欧氏距离太大，在后边的成图中将会很不便。本节中的举例将继续以表 3.2 的模式，将各样本假设为 13 维空间中的点，以对应段的像素百分比值为各维数值，算出它们间的欧氏距离（为了表格简洁，小数均保留两位），再做系统聚类。这里不用考虑量纲，数值上也不太大，根据前边的分析，可以用欧氏距离。

本次再任意取出 13 张微阵列图片，连同前边的两张共 15 张，进行聚类。原

图及对应亮度分布图详见附录 1 和附录 2。

表 3.3 亮度分段百分比例表

分段 / 图像	0～50	51～70	71～90	91～110	111～120	121～130	131～140	141～150	151～160	161～170	171～180	181～190	191～225	总计
1	2.73	5.37	12.77	23.18	11.83	9.94	7.91	6.83	7.08	9.52	2.79	0.04	0	99.99
2	1.99	3.61	11.97	19.87	11.01	11.23	9.78	9.34	13.44	7.06	0.38	0	0	99.68
3	0.98	1.65	9.77	33.60	17.16	11.49	7.09	5.65	5.14	5.06	2.38	0.04	0	100.01
4	1.17	3.26	15.18	26.06	12.71	11.41	10.84	9.96	6.26	2.98	0.17	0	0	100
5	1.26	1.79	6.26	20.08	17.04	18.72	14.97	10.16	6.84	2.70	0.17	0	0	99.99
6	0.85	2.43	12.81	27.13	15.10	14.33	11.43	8.93	5.58	1.37	0.04	0	0	100
7	1.47	2.09	7.21	25.20	19.81	19.40	12.01	7.00	4.81	0.98	0.02	0	0	100
8	2.53	4.16	16.70	30.72	15.43	11.64	7.28	4.39	2.80	2.09	1.53	0.68	0.05	100
9	0.90	1.95	4.70	11.33	11.15	18.58	25.15	20.62	5.36	0.25	0	0	0	99.99
10	0	0	0.07	1.12	1.39	2.29	3.51	5.57	8.93	12.13	14.34	14.76	35.91	100.02
11	0	0	0	0.11	0.14	0.16	0.27	0.68	1.71	6.33	27.62	41.55	21.43	100
12	0	0	0.56	6.63	5.62	5.41	5.02	4.63	4.50	4.52	4.94	5.76	52.40	100
13	0	0	0.64	3.65	3.41	4.38	5.38	6.06	6.21	6.63	7.50	7.84	48.29	100

14	0	0	0.07	0.54	0.58	0.94	1.31	1.84	2.95	6.77	18.32	30.37	36.31	100
15	0	0.01	0.40	2.76	3.45	5.78	7.32	8.17	8.46	9.61	10.66	9.20	34.18	100

完成图像的数据提取以后，计算它们间的距离（以欧氏距离为例）。总共是15个样本，每两个样本间都有一个距离，这样共有 $s=\frac{n(n-1)}{2}=\frac{15\times14}{2}=105$ 个，不含自己和自己的距离（为0）。计算结果见表3.4。

表3.4 欧氏距离表

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0														
2	8.93	0													
3	13.83	18.27	0												
4	9.62	11.05	12.32	0											
5	16.90	14.80	18.53	14.39	0										
6	12.28	13.40	10.63	5.13	11.56	0									
7	17.41	17.88	13.98	14.01	7.83	9.40	0								
8	13.10	18.19	9.22	9.64	19.84	9.48	15.57	0							
9	29.66	25.67	34.46	26.63	18.31	25.99	25.17	34.49	0						
10	50.22	49.84	56.68	54.27	53.58	55.60	56.55	57.12	55.25	0					
11	62.43	63.42	67.40	66.10	66.01	67.11	67.89	67.05	68.03	35.09	0				

12	57.85	57.59	61.46	59.66	58.87	60.17	60.52	61.57	61.08	24.09	53.89	0			
13	55.66	55.13	60.41	58.03	57.18	58.85	59.47	60.53	58.61	17.45	48.78	7.22	0		
14	59.08	59.56	64.23	62.45	62.21	63.48	64.26	63.94	64.12	18.59	20.93	33.99	29.04	0	
15	44.85	43.91	51.21	48.15	46.55	49.22	49.87	51.70	47.50	9.72	41.33	21.35	15.36	25.74	0

为方便后边的计算和标注，首先定义几个概念：设样本 x_p 和 x_q 间的距离为 d_{pq}，设类 G_p 与类 G_q 间的距离为 D_{pq}。当

$$D_{pq} = \min_{x_i \in G_p, x_j \in G_q} d_{ij} \tag{3.16}$$

时，也就是等于类 G_p 和 G_q 中距离最小的两个样品的距离，用这个距离进行系统聚类，就是最短距离法。

按照本节开始时提供的系统聚类的一般步骤对此 15 个样本用最短距离法进行聚类如下：

1. 将样品各成一类，即有 15 个类，样本间的距离见表 3.4，此时 $D_{pq} = d_{pq}$；

2. 将表 3.4 看做一个矩阵，设为 $D_{(0)}$，找出类与类（非对角元素）间的最小距离，即 G_4 和 G_6 的距离 $D_{46}=d_{46}=5.13$，将它们合并构成第 16 个类 G_{16}，有 $G_{16}\{G_4, G_6\}$，记为 G_r；

3. 带入公式 3.16 求新类 G_{16} 与其他类的距离：

$$\begin{aligned} D_{rk} &= \min_{x_i \in G_r, x_j \in G_k} d_{ij} = \min\left\{ \min_{x_i \in G_p, x_j \in G_k} d_{ij}, \min_{x_i \in G_q, x_j \in G_k} d_{ij} \right\} \\ &= \min\{D_{pk}, D_{qk}\} \\ &= \min\{D_{4k}, D_{6k}\}, \text{（设 p<q）} \end{aligned} \tag{3.17}$$

在这里，G_4 和 G_6 已经合并，即都不存在了，在原来 G_4 的位置上填上了它们合并后的新类 G_{16}，G_{16} 和其他类的距离根据公式 3.17 结果可知，是取原来 G_4 和 G_6 与其它类间距离中的小的那个值。生成一个新的矩阵 $D_{(1)}$，以表格形式表示见表 3.5；

表 3.5 第一次并类后的距离阵

	1	2	3	16	5	7	8	9	10	11	12	13	14	15
1	0													
2	8.93	0												
3	13.83	18.27	0											
16	9.62	11.05	10.63	0										
5	16.90	14.80	18.53	14.39	0									
7	17.41	17.88	13.98	9.40	7.83	0								
8	13.10	18.19	9.22	9.48	19.84	15.57	0							
9	29.66	25.67	34.46	25.99	18.31	25.17	34.49	0						
10	50.22	49.84	56.68	54.27	53.58	56.55	57.12	55.25	0					
11	62.43	63.42	67.40	66.10	66.01	67.89	67.05	68.03	35.09	0				
12	57.85	57.59	61.46	59.66	58.87	60.52	61.57	61.08	24.09	53.89	0			
13	55.66	55.13	60.41	58.03	57.18	59.47	60.53	58.61	17.45	48.78	7.22	0		
14	59.08	59.56	64.23	62.45	62.21	64.26	63.94	64.12	18.59	20.93	33.99	29.04	0	
15	44.85	43.91	51.21	48.15	46.55	49.87	51.70	47.50	9.72	41.33	21.35	15.36	25.74	0

4. 对 $D_{(1)}$ 重复 $D_{(0)}$ 刚才的步骤，即第 2 步，再次找出非对角元素的最小距离值，

即 G_{12} 和 G_{13} 的距离 7.22，…，生成 $D_{(2)}$，以此继续并类，直到它们最后成为一个类为止。

起始为 15 个类，每次循环，并类一次，即需要并类 14 次后变成一个类。每次并类结果见附录 3。最后结果，即利用附录 3 中每步的结果画出直观的聚类图，见图 3.5。

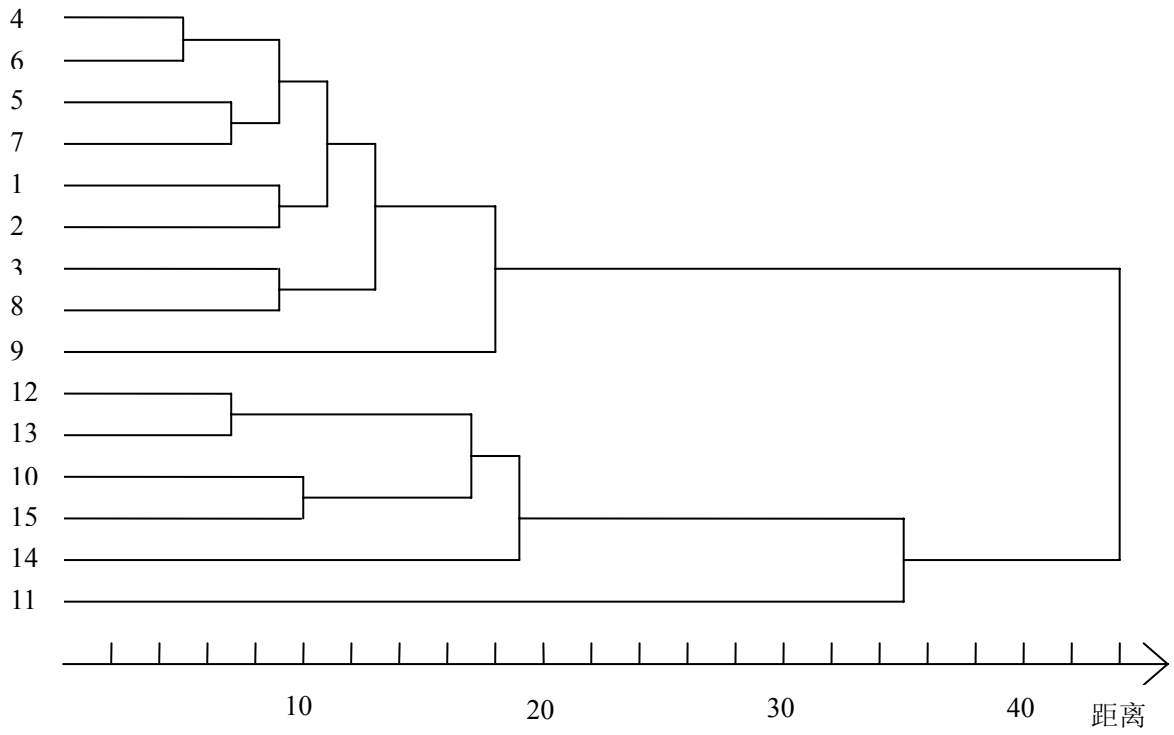


图 3.5 微阵列样本聚类图

从聚类图 3.5 上可以很明显的看出，样本 4、6 间最接近，而它俩与样本 12、13、10、15、14、11 间明显差异特别大。现将样本 4、6、11 单独取出看一下聚类效果，见图 3.6，可以看出得到的结果比较满意。

3.5.2 最长距离法及其它

根据公式 3.16

$$D_{pq} = \min_{x_i \in G_p, x_j \in G_q} d_{ij}$$

对最短距离法的定义，我们可以再定义一个最长距离法：

$$D_{pq} = \max_{x_i \in G_p, x_j \in G_q} d_{ij} \qquad (3.18)$$

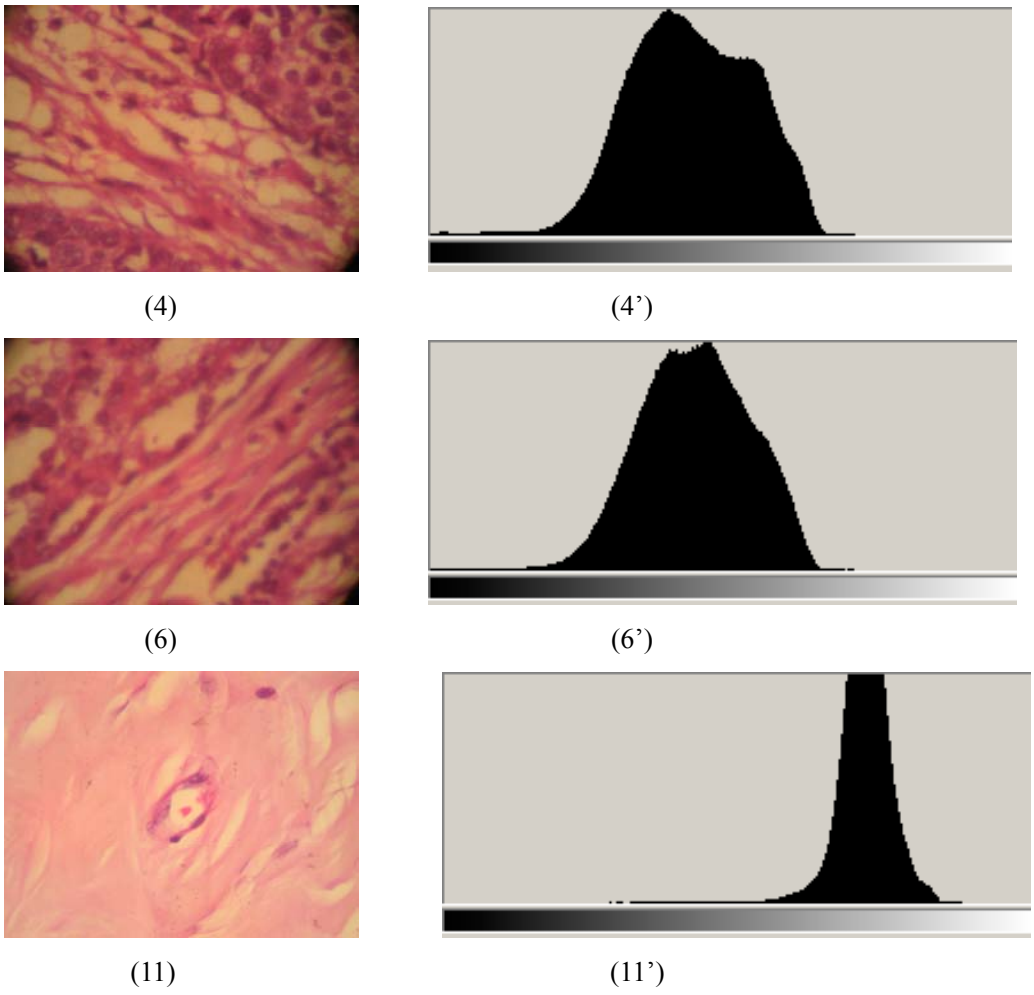


图 3.6 样本 4、6、11 原图及亮度分布图

这样最短距离法中的公式 3.17 在这里就成了

$$D_{rk} = \max\{D_{pk},\ D_{qk}\} \tag{3.19}$$

至于进行聚类，依然是本节最前面提供的“一般步骤”，依然是每步将距离最小的两个类并类，算出新类与其它类的距离，循环，直到变成一类为止。

最长距离法和最短距离法的区别就在于怎么确定新类与其它类的距离，一个是取小，一个是取大。如果既不取小，也不取大，而是取介于两者之间的距离，这就可以定义为中间距离法。如果在某一步将类G_p与类G_q合并为G_r，任一类G_k和G_r的距离公式为：

$$D_{kr}^2 = \frac{1}{2}D_{kp}^2 + \frac{1}{2}D_{kq}^2 + \beta D_{pq}^2 \qquad -\frac{1}{4} \le \beta \le 0 \tag{3.20}$$

当 $\beta=-\frac{1}{4}$ 时，由初等几何知 D_{kr} 就是以 D_{kp} 和 D_{kq} 为两腰的三角形的中线。如果用最短距离法，则 $D_{kr}=D_{kp}$；如果用最长距离法，则 $D_{kr}=D_{kp}$；如果取夹在这两边的中线作为 D_{kr}，则 $D_{kr}=\sqrt{\frac{1}{2}D_{kp}^2+\frac{1}{2}D_{kq}^2-\frac{1}{4}D_{pq}^2}$，由于距离公式中的量都是距离的平方，为了计算机实现的方便，在聚类时，可将距离阵 $D_{(0)}$、$D_{(1)}$、$D_{(2)}$…中的元素，都用相应元素的平方代替而得到平方距离阵 $D_{(0)}^2$、$D_{(1)}^2$、$D_{(2)}^2$…，聚类结果几乎不受影响。

用中间距离法进行聚类也遵循“一般步骤”，聚类结果显然也是介于最长距离法和最短距离法之间。但是它倾向性不明显，而且它是先给出递推公式再计算类与类间的距离，这于前两种方法（先给出类与类间的公式然后推出递推公式）刚好相反，计算出的距离显然受聚类过程的影响。

为了解决中间距离法的弊端，又出现了重心法、类平均法、可变类平均法、可变法、离差平方和法等多种方法，其实这些方法聚类的步骤是完全一样的，所不同的是类与类之间的距离有不同的定义法，依法所给出的新类与任一类的距离公式不同。有人[11][12]为此还将各种距离统一起来，如欧氏距离条件下的统一是1967 年由兰斯（Lance）和威廉姆斯（Williams）实现的，他们将前边提到的八种方法统一成一个递推公式：

$$D_{KR}^2=\alpha_p D_{kp}^2+\alpha_q D_{kq}^2+\beta D_{pq}^2+\gamma|D_{kp}^2-D_{kq}^2| \qquad (3.21)$$

如果不采用欧氏距离时，除重心法、中间距离法、离差平方和法之外，该统一形式的递推公式仍成立。上式中参数 α_p、α_q、β、γ 对不同的方法有不同的取值。表 3.6 列出上述八种方法中参数的取值。将八种方法公式统一，给编制计算机程序提供了很大的方便。

表 3.6 统一公式中的参数取值

方　法	a_p	a_q	β	γ
最短距离法	1/2	1/2	0	-1/2
最长距离法	1/2	1/2	0	1/2
中间距离法	1/2	1/2	$-1/4\geq\beta\geq 0$	0
重　心　法	n_p/n_r	n_p/n_r	$-\alpha_p\alpha_q$	0
类 平 均 法	n_p/n_r	n_p/n_r	0	0
可变类平均法	$(1-\beta)\,n_p/n_r$	$(1-\beta)\,n_p/n_r$	＜1	0

可　变　法	$(1-\beta)/2$	$(1-\beta)/2$	<1	0
离差平方和法	n_i+n_p/n_i+n_r	n_i+n_p/n_i+n_r	$-n_i/n_i+n_r$	0

总之，不同方法聚类的步骤完全一样，仅仅是区别于怎样去定义类与类之间的距离，怎样去计算这个距离而已。一般情况下，用不同的方法聚类的结果是不完全一致的，哪一种方法更好呢？这就需要提出一个标准作为衡量的依据。

3.5.3 聚类结果的比较

不同的聚类方法往往有着不同的结果，怎样确定哪个更贴近些，早在 1968 年 Guttman 就用夹角余弦计算结果距离阵与初始距离阵间的相关性来表述。引用他们的一个例子[10]做简要说明。注意，这里的样本非本论文附录所示的样本。

表 3.7 初始距离阵

样本	1	2	5	7	9	10
1	0					
2	1	0				
5	4	3	0			
7	6	5	2	0		
9	8	7	4	2	0	
10	9	8	5	3	1	0

用此距离阵对样本分别用最短距离法和最长距离法做聚类，得到两个聚类图：

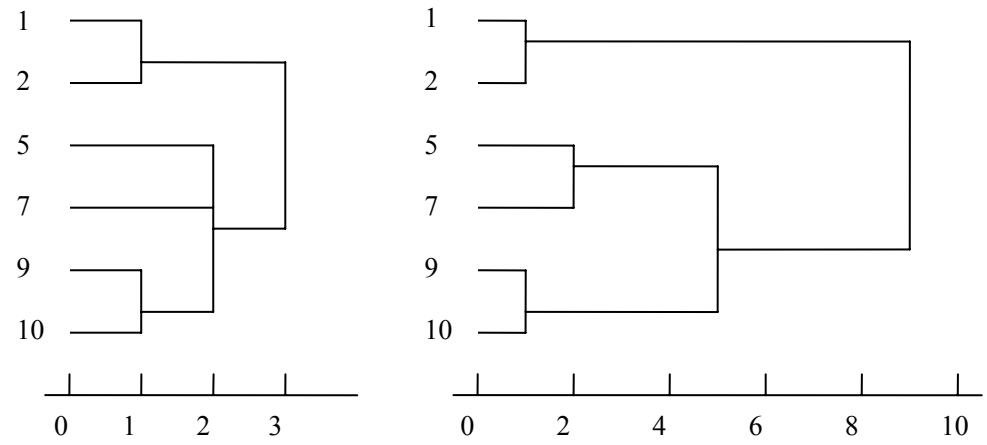


图 3.7 最短距离法（左）和最长距离法（右）聚类图

从聚类图中可以读出初始类间的“距离”，得到两个新的距离阵：

表 3.8 读出的最短距离阵

样本	1	2	5	7	9	10
1	0					
2	1	0				
5	3	3	0			
7	3	3	2	0		
9	3	3	2	2	0	
10	3	3	2	2	1	0

表 3.9 读出的最长距离阵

样本	1	2	5	7	9	10
1	0					
2	1	0				
5	9	9	0			
7	9	9	2	0		
9	9	9	5	5	0	
10	9	9	5	5	1	0

设 d_{ij} 为初始距离阵重的元素，d_{ij}^{*} 为读出来的距离阵重的元素，利用公式 3.9 计算它们的相关系数，看哪个新距离阵更接近于初始距离阵，即对应的聚类图更合理些。

在这里将公式 3.9 表述为

$$c = \frac{\sum_{i=1}^{n}\sum_{j=1}^{i} d_{ij} d^{*}_{ij}}{\sqrt{(\sum_{i=1}^{n}\sum_{j=1}^{i} d_{ij}^{2})(\sum_{i=1}^{n}\sum_{j=1}^{i} d_{ij}^{*2})}} \qquad (3.22)$$

对最短距离法

$$c = \frac{1\times1+(4+6+8+9)\times3+(3+5+7+8)\times3+(2+4+5)\times2+(2+3)\times2+1\times1}{\sqrt{\sum(1^2,4^2,6^2,8^2,9^2,3^2,5^2,7^2,8^2,2^2,4^2,5^2,2^2,3^2,1^2)(1+8\times3^2+5\times2^2+1^2)}}$$

$$= \frac{184}{404\times94} = 0.944$$

同理对最长距离法算得 c＝0.953>0.944，即在这个例子中，用最长距离法好些。

3.6 本章小结

本章指出了组织微阵列因数据量过于庞大，造成很多分析不便，对它做聚类显得尤为重要，随后论述了对它做聚类分析的可行性。本章中的例子是针对一个医生的要求，对一些微阵列图片所做的聚类，但根据论文编写期间进一步的对组织微阵列的认识，我个人认为这个例子只是证明了对微阵列图片在组织形态分布上可以进行聚类，而且聚类效果也很好，只是用途值得考虑，当然了，它可以应用在临床教学上增强学生对各类相似病理图片的区分能力。在组织微阵列和病理切片的制备中有个很重要的过程就是染色，如把细胞核染成蓝色，而细胞核明显变大或核仁（染色后与细胞核颜色也有明显区别）增多又是癌症的一些反映，我们何不在这个重点的区域内提取出一些合适的条件对微阵列进行聚类呢？其次，每种病理论上都有它的病因，并且很多病又与性别、年龄、工作、地域等条件相关，比如癌症，它的致病原因至今还是人类医学上的一个迷，我们也可以把各种相关条件作为微阵列聚类的条件进行聚类研究，以掌握各种因素在致癌中的权重，为以后预防和攻克癌症打下基础。由于各种条件的限制，这只能作为我以后继续研究的内容。

第四章　程序实现探讨

在前章的论述中，我们可以看到，除了在提取组织微阵列图像信息如各段亮度分布等方面不得不用到计算机外，聚类中还有大量的计算工作，如果用人工去计算，单对计算已提取亮度分布（或其它分布）信息的两幅微阵列图的欧氏距离就得花上半天的时间，更别说将几十张、几千张这样的图像按照不同的要求进行聚类了。本章重点将探讨在组织微阵列聚类分析研究中，让计算机代替人做繁重计算的一些模块的实现问题。

4.1 信息快速提取模块

这里主要是针对组织微阵列图像中的组织形态分布特征的，即从图像的亮度和 RGB 分布的直方图里边分段提取数据。本模块的实现以附录 1 中的图 13 为例介绍。

实现上主要分成以下三个步骤：

1. 图象读取

程序的实现和读取形式和其他软件一样。

2. 四种直方图的生成

选择合适的分段提取各段数据，因为不均匀分段难度很高，在第三章的例子中，查看了很多幅图的柱状图以后才作出“13”段的分布，然而，在后边其他图像信息的提取中却发现这个分类还是漏掉了很多重要的信息。所以在软件的编写时，特意先将它实现为均匀分段，即可以选择分 2、4、8、16、32、64、128、256 段的八种情况。当然了，为了照顾某些特殊需要，在柱状图中设置了可以自由选择分段（分鼠标拖动和直接填写分段数字两种）来读取该段数据的功能。

图 4.1 提取直方图

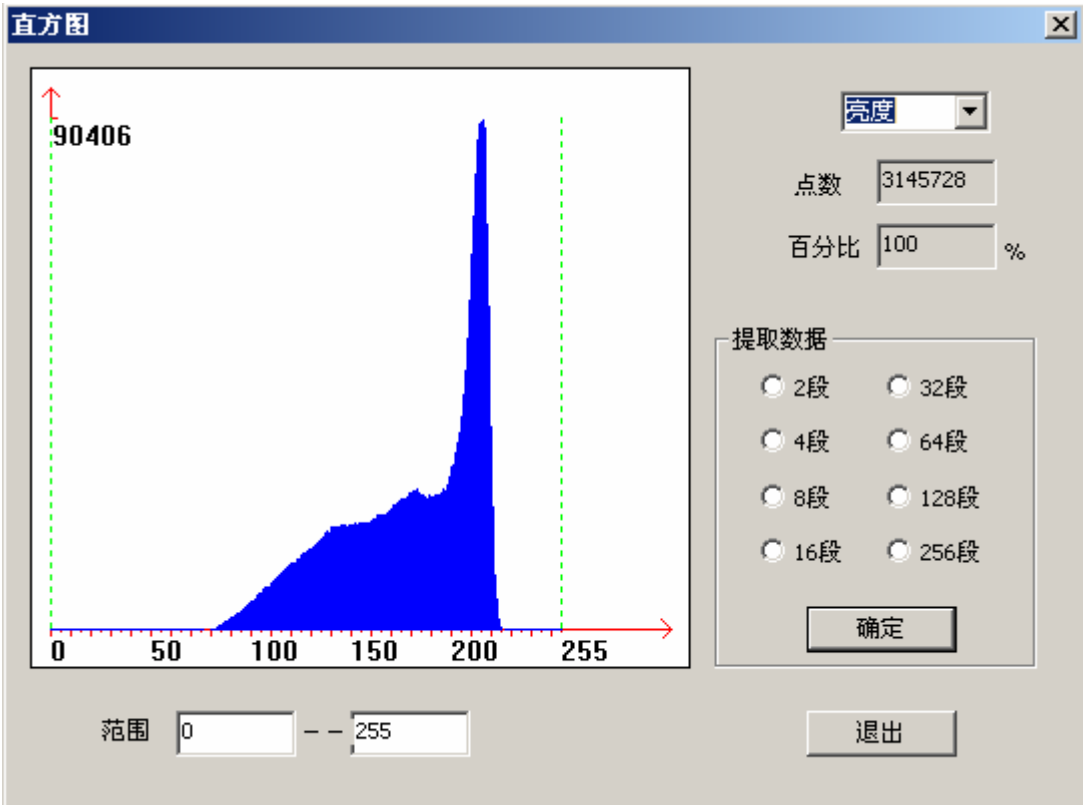


图 4.2 图 13 的直方图对话框

3. 将读出的数据自动存储

这里是以文本文件的形式实现逐个样本数据连续追加的，即把每次读出的数据可以存在自己任意指定的一个文本文件里边，标注的信息有：图片名（含图片目录）、何种柱状图、分段情况及对应数据（像素点及所占百分比）。

如果我们在图 4.2 所示的对话框中选择“4 段”，然后点击“确定”，即可弹出路径选择，打开需要存放的文本文件，数据便自动存进去了。

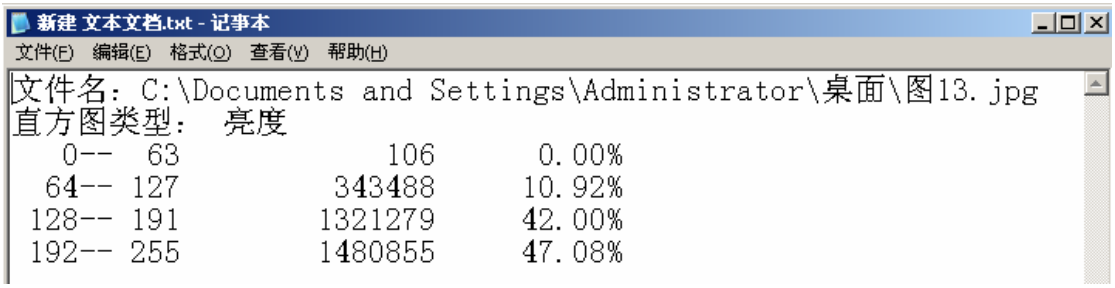


图 4.3 生成的图 13 的数据

如果要将 n 个样本提取数据，则依次打开每张图片，点击生成直方图，再选

取要做分析的直方图，选择分段，然后点确定，数据即可自动一一追加到指定文本文件中。

然后我们再将图 4.2 中所显示的直方图经过长宽调整后与与附录 2 中用 Photoshop7.0 得到的直方图进行对比。

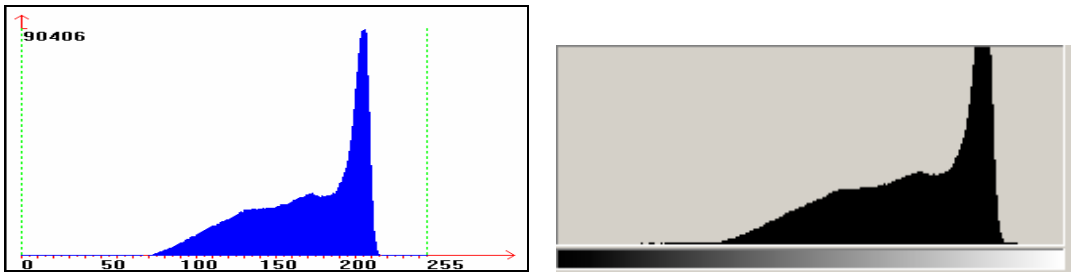


图 4.4 图像验证

可以看出，如果忽略长宽比，它们是完全一样的。，而像素点是由计算机统计的，稳定性很高。这样，该模块的实现证明是很成功的

4.2 距离阵生成模块

当提取出微阵列图像的数据后，我们需要计算图像两两之间的距离，以便进行以后的聚类。这里涉及到距离公式选择问题和矩阵大小的定义问题。距离以欧氏距离为例。对于矩阵的定义，在原实现的信息提取模块里边加了一项矩阵设置，为“文件”下拉菜单下的“聚类设置”，打开该软件，选择该项功能，即可弹出图 4.5 所示的对话框，填写要做分类的图片数（即定义一个存放图片信息的矩阵）和选择柱状图的分段数后，确认输入。

图 4.5 聚类设置对话框

在以后的功能扩展中，在图 4.5 的对话框中就可以再添加“距离选择”的功能，并逐渐完善它的自由选择分段功能。现在以将直方图等分 8 段，对 5 幅（附录 1 中的 11～15）图进行聚类为例。首先定义矩阵和选择分段信息，如图 4.5 中的显示，然后点“OK”弹出对话框提示“请加载 5 幅图像”。将图 4.2 中右下角的部分进行修改以扩展出这个功能，见图 4.6。在打开一幅图后，生成所需（本例为亮度）柱状图，同样选择“8 段”，然后点击新加的功能“导入内存”（这里把以前的“确认”按钮改成了“导入文本”），提取出来的数据便自动存进了内存，如此将 5 幅图的信息均导入内存，这时会弹出对话框提示已经读取了 5 幅图，请进行聚类分析。“聚类分析”功能添加在图 4.1 显示的“视图”下拉菜单中，在“直方图”下边。点击该功能，也会弹出文件选择的路径，打开要存放信息的文本文件，距离阵自动显示进去，见图 4.7。

图 4.6 将图片信息导入内存

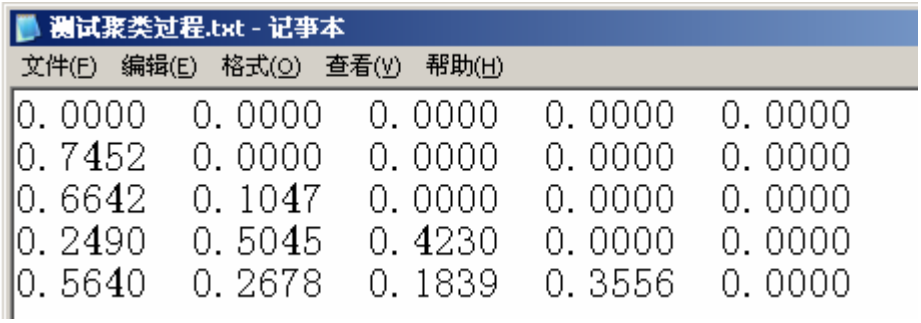


图 4.7 生成的距离阵

图上的距离显示的都小于 1，原因是为了程序简单，存入内存的数据设置成了各段像素所占柱状图总像素的百分比数（第三章中经比较证明了它的优点），在以后的程序扩展中根据需要用同样的方法很容易就添上了其它信息存储的功

能。

4.3 矩阵计算模块

本模块在聚类中很重要，每个新矩阵的生成实际上就是一步聚类的完成。在这一模块的实现上，继续以已编写的程序为基础。新程序添加在了“聚类分析”中，这样，上节中的文本文件除了显示距离阵以外还显示出每步的新距离阵及最后聚类的结果。

```
0.0000   0.0000   0.0000   0.0000   0.0000
0.6642   0.0000   0.0000   0.0000   0.0000
10.0000  10.0000  0.0000   0.0000   0.0000
0.2490   0.4230   10.0000  0.0000   0.0000
0.5640   0.1839   10.0000  0.3556   0.0000

0.0000   0.0000   0.0000   0.0000   0.0000
0.5640   0.0000   0.0000   0.0000   0.0000
10.0000  10.0000  0.0000   0.0000   0.0000
0.2490   0.4230   10.0000  0.0000   0.0000
10.0000  10.0000  10.0000  10.0000  0.0000

0.0000   0.0000   0.0000   0.0000   0.0000
0.5640   0.0000   0.0000   0.0000   0.0000
10.0000  10.0000  0.0000   0.0000   0.0000
10.0000  10.0000  10.0000  0.0000   0.0000
10.0000  10.0000  10.0000  10.0000  0.0000

0.0000   0.0000   0.0000   0.0000   0.0000
10.0000  0.0000   0.0000   0.0000   0.0000
10.0000  10.0000  0.0000   0.0000   0.0000
10.0000  10.0000  10.0000  0.0000   0.0000
10.0000  10.0000  10.0000  10.0000  0.0000

(  2       1)              0.1047
(  4       1)              0.1839
(  3       0)              0.2490
(  1       0)              0.5640
```

图 4.8 距离阵的计算结果

从上图中可以看出，每步进行完并类后，本应该删除并类后的两行（列），并将生成的新行（列）距离值放在原来靠近左上角的已删除行（列）的位置，而在这里的实现是将新行（列）的值赋给对应位置的行（列），把另外一行（列）

里边的值赋上一个远远大于 1 的值 10（因为提取信息中用的是各段所占百分数，算出的距离一定小于 1），这是为了计算机实现同址计算的方便，同时又不影响每步查找距离阵中的非对角元素的最小距离。对于最后结果的显示，如图 4.8 最下边的显示，左边括号内的数字是从 0～4，这是因为计算机中的矩阵是从“0”行（列）开始的，这样，初始距离阵中的最小值 0.1047 是距离阵中的第 2 行第 1 列，在本例中即为图片 13 和图片 12 的距离。将结果表示成附录 3 中显示的结果为：

1. G_2（图 13）和 G_1（图 12）并类成 G_1’，距离为 0.1047；
2. G_4（图 15）和 G_1’并类成 G_1”，距离为 0.1839；
3. G_3（图 14）和 G_0（图 11）并类成 G_0’，距离为 0.2490；
4. G_1”和 G_0’并类成 G_0”，距离为 0.5640，完成矩阵运算。

4.4 聚类图生成模块

本模块是利用矩阵计算的结果将对应的样本逐步连线生成图像的过程。第三章中的聚类图我是手工完成的，以找出程序编写时应该注意的地方。其中发现如果简单地认为按步骤（见附录 3）依次排列各个样本，然后画线的话就会出现连线交叉的情况。即 G_{12} 和 G_{13} 并类成 G_{17} 是第二步聚类完成的， G_{10} 和 G_{15} 并类成 G_{23} 是第八步完成的，到了第十步 G_{17} 和 G_{23} 并类成 G_{25}，但是实际上，聚类到第四步的时候，新类 G_{17} 已经被“包”了进去，在第十步与 G_{23} 并类时势必要造成连线交叉。

4.5 本章小结

本章用 VC 编程，实现了简单情况下的对组织微阵列图片的计算机聚类，以一个框架的形式出现，可以方便以后继续研究中的功能扩展。以举例的形式，逐渐介绍了各个功能模块。直方图实现的主要程序见附录 4，聚类过程及得出结果见附录 5。对于第四个模块，在编写过程中，由于作者软件水平有限，对以上遇到的问题没能解决，尚在寻找新的解决办法。

第五章 总结与展望

聚类分析广泛应用在需要做分类的很多学科中，特别是需要分类的事物间的关系较为模糊时更显它独特的优势。组织微阵列就属于这样的事物，虽然它被称为生命科学中一大突破性发明，但是它所包含的信息过于庞大，要对它进行分析处理去获得一些需要的信息难度特别大，如果再想去挖掘其他可能含有的我们难以把握的信息时则更是叫人无从下手。如果把聚类分析用在这里，问题便简单的多了。

5.1 总结

本文是针对组织微阵列研究的现状而做的。组织微阵列自其诞生也有将近八个年头的时间了，可是国内仍年年有大量文章在探讨怎么去制作组织微阵列，实际有用的内容往往只是介绍某个工艺实现的技巧。其它的应用文章则一般是将某个特殊病例的组织标本制成微阵列，像分析传统的病理切片一样，最后得出显阳性阵列占的百分比，应用范围很窄。

为了普及和拓宽对它的应用，论文的第二章对组织微阵列做了详细的论述，分析了它的应用范围，并总结出一套详尽的制作步骤，这对它的普及将起着很大的作用。第三章则主要为了使对它的研究简单化，引入了聚类分析，根据一个医生的要求举例，逐步实现对 15 张乳腺癌组织微阵列图片的聚类。论证了从组织形态特征分布上对它进行聚类的可行性。并根据聚类步骤中和对微阵列的分析中不断发现的问题，提出了一些进行实用性聚类的突破口。

论文最后是针对聚类分析需要进行大量繁重计算而做的，即用计算机软件的形式实现聚类的各个步骤。使对组织微阵列的聚类分析大大简单化，以便让不同的研究人员可以根据自己的研究目的应用聚类分析。为了使得分析结果更具有广泛性，更有实用价值，必须有足够大量的数据，而聚类分析软件程序的实现使样本在收集上可以真正做到“海量”，因为只要确定了分类标准，很容易就能对它们进行聚类。

5.2 展望

组织微阵列是连接宏观生物学和微观生物学重要的纽带，它将在整个生命科学的发展中起到不可估量的作用，希望以此文能带动起更多的人去对它进行研

究。从文献看，本文是国内第一次将聚类分析正式引入组织微阵列的研究中来的学术性论文，并实现了用亮度柱状图分布规律对组织微阵列图片的聚类。但是由于条件的限制，尚未获得更多的进行分类的标准来验证聚类效果，使得软件实现上功能不够强大，在以后研究中将把重点放在收集组织微阵列研究人员的意见上，以获取更多的分类标准，逐渐完善聚类软件系统。希望有一天能够用组织微阵列的聚类研究找出各种医学难题如癌症的真正致病原因，以实现对它的治疗和预防。甚至去解决我们现在还想不到的许多生物学及医学问题。

参考文献

[1] 张阳德. 生物信息学. 北京：科学出版社，2004. 248～262

[2] 杨军，张明娟. 一种新的生物芯片——组织芯片. 中国科学基金，2001，3：141～144

[3] 高勤，吴晓松，翁文. 生物芯片及其在医药领域中的应用. 广东药学院学报，2003，19（3）：260～262

[4] 周冬生. 生物芯片分类及其技术原理. 微生物学免疫学进展，2002，30（3）：101～107

[5] Yiang C C，Chen Y. cDNA microarray technology and its applications. Biotechnol Advance，2000，18：35～46

[6] 邓征浩，周建华. 芯片家族新成员——组织芯片. 中南大学学报(医学版)，2004，29(1)：102～104

[7] Kononen J，Bubendorf L. Tissuemicroarrays for high-through put molecularprofiling of tumor specimens. Nature Med，1998，4(7)：844～847

[8] 南广友，胡柏平. 组织芯片技术及其在体育生物中应用展望. 四川体育科学，2005，1：26～28

[9] Wan WH，Fortuna MB，Gurmanski P. A rapid and efficient method for testing immunohistochemical reactivity of monoclonal antibodies again multiple tissue samples simultaueously. J Immumol Meth，1987，103(2):121～129

[10] 方开泰，潘恩沛. 聚类分析. 北京：地质出版社，1982. 23～43

[11] 张维群. 聚类分析中考虑权重问题的应用探讨. 统计与信息论坛，1997，2：44～49

[12] Wishart，W. An Algorithm for Hierarchical Classification，Biometrics，1969，25(1):165～170

[13] 朱蔚萍，靳宏磊，叶桦，等. 二维灰度直方图上的距离判别分割方法. 东南大学学报，1999，29：16～19

[14] 李占利，张群会，张家彬. 一种扩展的动态聚类分析方法. 数理统计与管理，1994，13（5）：50～52

[15] David Tritchler，Shafagh Fallah，Joseph Beyene. A spectral clustering method for microarray data. Computational Statistics & Data Analysis，2005，49：63－76

[16] Chin-Hsiung Wu，Shi-Jinn Horng，Horng-Ren Tsai. Efficient Parallel Algorithms for Hierarchical Clustering on Arrays with Reconfigurable Optical

Buses. Journal of Parallel and Distributed Computing，2000，60：1137～1153

[17] R. p. Baker ，P. G. Maropoulos. An automatic clustering algorithm suitable for use by a computer-based tool for the design，management and continuous improvement of cellular manufacturing systems. Computer Integrated Manufacturing，1997，10(3)：217～230

[18] 胡钟山，丁震，杨静宇，等. 一种改进的 Fuzzyc-means 聚类算法. 南京理工大学学报，1997，21（4）：337～340

[19] 杨威，张田文，师海峰. 一种用于二值图象分割的快速聚类算法. 计算机研究与发展，1998，35（8）：719～723

[20] Yiu-MingCheung，k^*-Means: A new generalized k-means clustering algorithm. Pattern Recognition Letters，2003，24：2883～2893

[21] Niina Paivinen，Clustering with a minimum spanning tree of scale-free-like structure. Pattern Recognition Letters，2005，26：921～930

[22] 杨海玉，刘勇. 组织芯片技术在肿瘤研究中的进展. 九江学院学报（自然科学版），2005，1：109～111

[23] 朱丛中，王新允，刘婷，等. 应用组织微阵列技术研究肺癌组织中 IGF-Ⅱ的表达. 中国癌症杂志，2005，15（2）：126～129

[24] 张喜平，居同法. 组织芯片技术及应用. 中国中西医结合外科杂志，2005，11（2）：167～169

[25] 石群立，孟奎，陈琴，等. 组织芯片应用的现状与前景. 诊断学理论与实践，2005，1：4～8

[26] 滕猛，王翠芳. 组织芯片技术. 中国生物学文摘，2005，1：69～71

[27] 庞永刚，崔鹏程，陈文弦. 新兴细胞生物学技术—— 组织微阵列研究进展. 中国临床康复，2004，8（8）：1520～1521

[28] 申姜颖，宫敏，赵斋川，等. 组织芯片技术在实验教学中的研究与应用. 解剖科学进展，2004，10（3）：285～286

[29] 王杨，陈茂怀. 组织芯片技术的进展及其应用. 汕头大学医学院学报，2004，17（3）：185～186

[30] 齐宗利，杜孟刚. 组织微阵列在生物制药研发中的应用. 国外医学，2005，28（3）：123～126

[31] 邱志强，孙保存，张诗武，等. 乳腺癌转移相关基因表达蛋白组织微阵列的初步研究. 中国肿瘤临床，2004，31（5）：252～255

[32] 杨军，苏宝山，王康敏，等. 组织芯片技术的发展及应用. 中国实验诊断学，2003，7（3）：195～198

[33] 杨军，王康敏，苏宝山，等. 组织芯片技术在原位杂交中的应用. 临床与实验病理学杂志，2002，18（3）：344～344

[34] 张林杰，方嬿，黄必军，等. 应用组织微阵列技术分析各期鼻咽癌组织p16的表达. 中华病理学杂志，2002，31（2）：132～134

[35] 周小鸽，张劲松，张小平，等. 组织芯片技术在检测正常组织和肿瘤组织抗原表达中的应用. 中华病理学杂志，2002，31（2）：181～182

[36] Bubendorf L，Kononen J，Koivisto P，et al. Survey of gene amplifications during prostate cancer progression by high through put fluorescence in situ hybridization on tissue microarrays. Cancer Res，1999，59：803～806

[37] Chen Wenjin，Foran David J. Advances in cancer tissue microarray technology: Towards improved understanding and diagnostics. Analytica Chimica Acta，2006，564(1)：78～81

[38] Simon Ronald，Mirlacher Martina，Sauter Guido. Tissue microarrays. Bio Techniques，2004，36（1）：98～105

[39] 茹晓荣. 组织微阵列技术. 解剖科学进展，2004，10：63～63

[40] 王新允，朱丛中，刘婷，等. 组织芯片研制的几点体会. 天津医科大学学报，2005，11（1）：24～25

[41] 孟奎，石群立，马恒辉，等. 组织芯片制备新方法. 医学研究生学报，2005，18（3）：287～288

[42] 王文勇，李玉松，赵一岭，等. 石蜡组织芯片制备技术方法的改良. 第四军医大学学报，2005，26（1）：93～94

[43] 龙汉安，程显魁，肖秀丽，等. 组织芯片制备的研究进展. 泸州医学院学报，2004，27（5）：459～460

[44] 王翠芝，周小鸽，王鹏，等. 组织芯片制作技术. 临床和实验医学杂志，2004，3（3）：183～184

[45] 胥维勇，杨群，范小莉. 石蜡包埋组织芯片制作的探讨. 中国组织化学与细胞化学杂志，2004，13（2）：255～256

[46] 罗晓青，曹进，王瑞国. 手工组织芯片制作. 数理医药学杂志，2004，17（3）：278～279

[47] 常峰，吴起嵩，王东关，等. 介绍一种组织芯片的制作方法. 诊断病理学杂志，2004，11（1）：63～63

[48] 但汉雷，张亚历，王亚东. 一种制作组织芯片的新方法. 癌症，2003，22（7）：778～781

[49] 孙保存，张诗武，赵秀兰，等. 组织芯片制备过程中的体会与注意事项. 临

床与实验病理学杂志，2002，18（6）：658～659

[50] 王学民，李军，杜波，等. 组织芯片的制备技术. 诊断病理学杂志，2002，9（6）：370～371

[51] 张诗武，李宏伟，张永亮，等. 组织芯片制作过程中石蜡的使用. 武警医学院学报，200211（3）：182～183

[52] 张巧英，姚根有. 石蜡组织芯片的制备方法及技巧. 实用肿瘤杂志，2005，20（6）：551～553

[53] 王丽君，刘勇，王红，等. 2001-2004 年组织芯片文献分析. 中华医学图书情报杂志，2005，14（3）：63～64

[54] Henke Ralf T，Eun Kim Sung，Maitra Anirban，et al. Expression analysis of mRNA in formalin-fixed，paraffin-embedded archival tissues by mRNA in situ hybridization. Methods，2006，38（4）：253～262

[55] Chen Wenjin，Foran David J. Advances in cancer tissue microarray technology：Towards improved understanding and diagnostics. Analytica Chimica Acta，2006，564（1）：74～81

[56] Devi Sachin S，Mehendale Harihara M. Microarray analysis of thioacetamide-treated type 1 diabetic rats. Toxicology and Applied Pharmacology，2006，212（1）：69～78

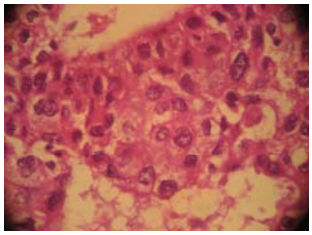
[57] McKay Jennifer S，Bigley Alison，Bell Alex，et al. A pilot evaluation of the use of tissue microarrays for quantitation of target distribution in drug discovery pathology. Experimental and Toxicologic Pathology，2006，57（3）：181～193

[58] Mougeot Jean-Luc C，Bahrani-Mostafavi Zahra，Vachris Judy C，et al. Gene Expression Profiling of Ovarian Tissues for Determination of Molecular Pathways Reflective of Tumorigenesis. Journal of Molecular Biology，2006，358（1）：310～329

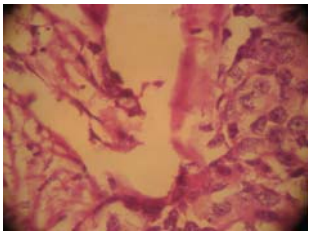
附 录

附录 1

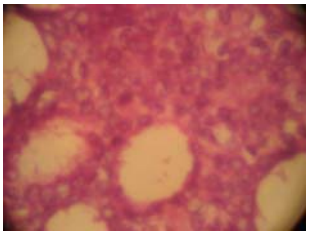
论文中引用的 15 张组织微阵列图片，显微镜放大倍数为 40*10，为 HE 染色的不同人的乳腺癌微阵列照片。



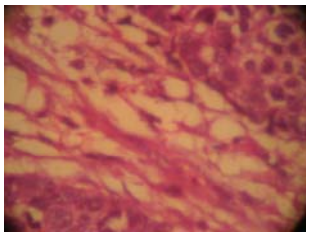
（1）



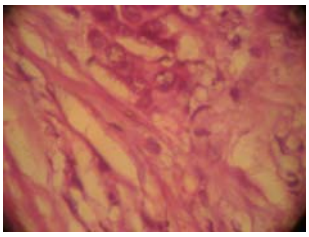
（2）



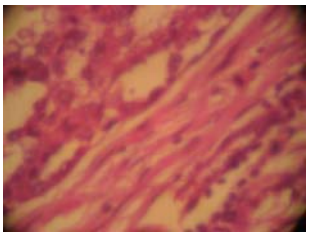
（3）



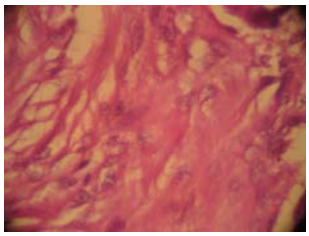
（4）



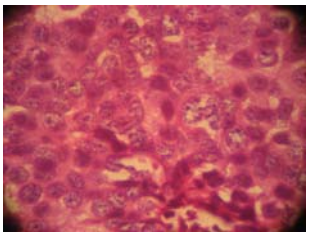
（5）



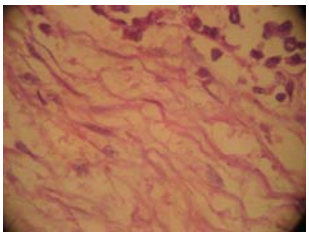
（6）



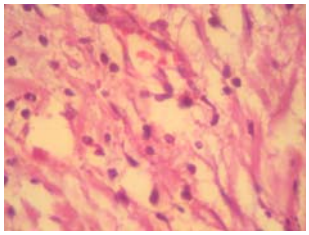
（7）



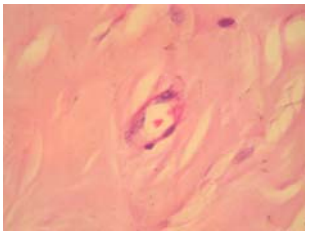
（8）



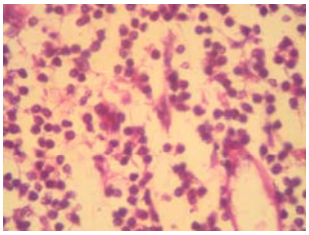
（9）



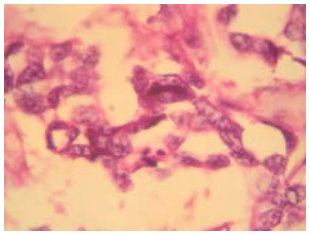
（10）



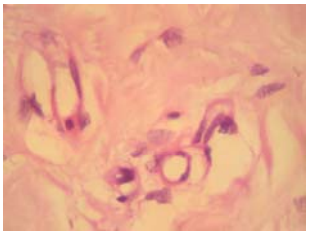
（11）



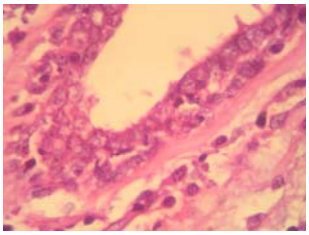
（12）



（13）



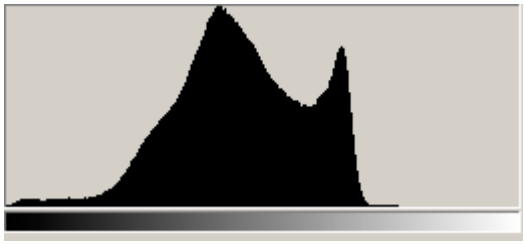
（14）



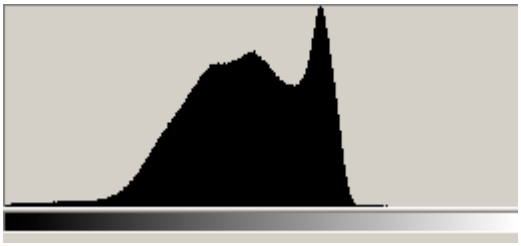
（15）

附录 2

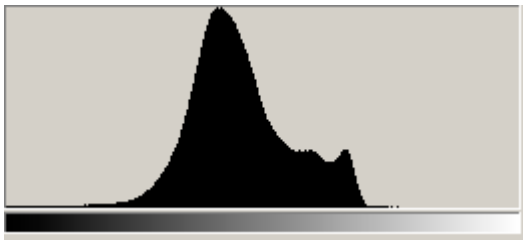
论文中引用的 15 张组织微阵列图片对应的亮度分布图。



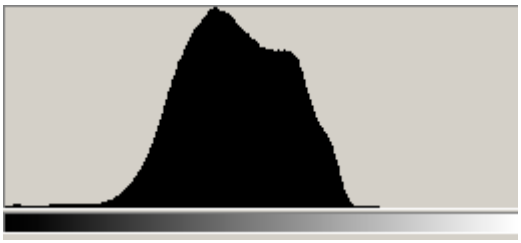
（1’）



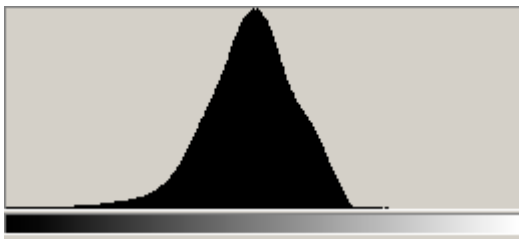
（2’）



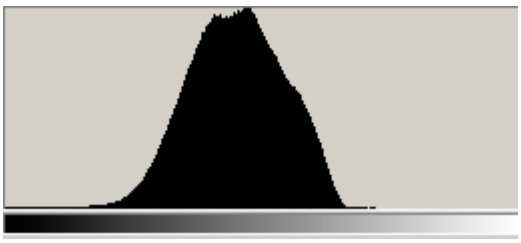
（3’）



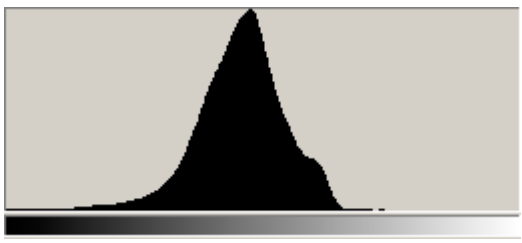
（4’）



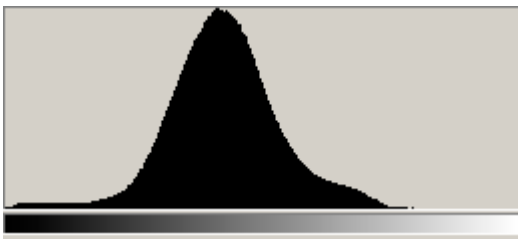
（5’）



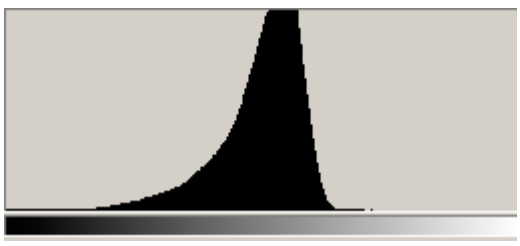
（6’）



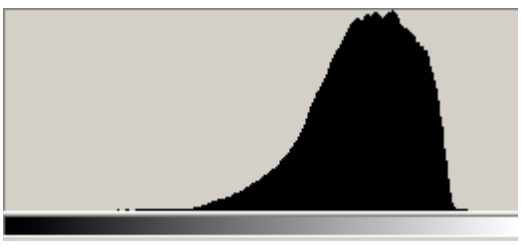
（7’）



（8’）



（9’）



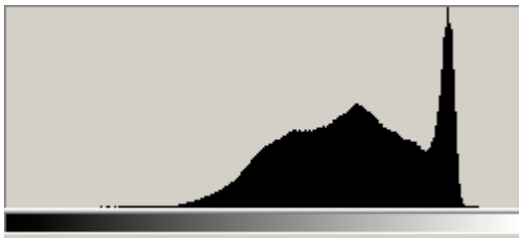
（10’）

（11’）

（12’）

（13’）

（14’）



（15’）

附录 3

论文中对 15 张微阵列图用最短距离法进行聚类时各步骤的详细结果。

1. G_4，G_6 并类成 G_{16}，距离 5.13；
2. G_{12}，G_{13} 并类成 G_{17}，距离 7.22；
3. G_5，G_7 并类成 G_{18}，距离 7.83；
4. G_1，G_2 并类成 G_{19}，距离 8.93；
5. G_3，G_8 并类成 G_{20}，距离 9.22；
6. G_{16}，G_{18} 并类成 G_{21}，距离 9.40；
7. G_{19}，G_{21} 并类成 G_{22}，距离 9.62；
8. G_{10}，G_{15} 并类成 G_{23}，距离 9.72；
9. G_{20}，G_{22} 并类成 G_{24}，距离 13.10；
10. G_{17}，G_{23} 并类成 G_{25}，距离 17.45；
11. G_9，G_{24} 并类成 G_{26}，距离 18.31；
12. G_{14}，G_6 并类成 G_{27}，距离 18.59；
13. G_{11}，G_{27} 并类成 G_{28}，距离 35.09；

14. G_{26}，G_{28} 并类成 G_{29}，距离 43.91。

附录 4

直方图实现中的主要程序代码。

```
BOOL CDlgIntensity::OnInitDialog()
{
    CDialog::OnInitDialog();
    m_colorCombo.AddString("亮度");
    m_colorCombo.AddString("红色");
    m_colorCombo.AddString("绿色");
    m_colorCombo.AddString("蓝色");
    m_colorCombo.SetCurSel(0);
    unsigned char * lpSrc;
    LONG i;
    LONG j;
    CDialog::OnInitDialog();
    CWnd* pWnd = GetDlgItem(IDC_COORD);
    pWnd->GetClientRect(m_MouseRect);
    pWnd->ClientToScreen(&m_MouseRect);
    CRect rect;
    GetClientRect(rect);
    m_MouseRect.top -= rect.top;
    m_MouseRect.left -= rect.left;
    m_MouseRect.top += 25;
    m_MouseRect.left += 10;
    m_MouseRect.bottom = m_MouseRect.top + 255;
    m_MouseRect.right = m_MouseRect.left + 256;
    for (i = 0; i < 256; i++)
    {
        m_RCount[i] = 0;
        m_GCount[i] = 0;
        m_BCount[i] = 0;
        m_HCount[i] = 0;
    }
```

```
    int liangdu = 0;
    double r,g,b;
    for (i = 0; i < m_Height; i ++)
    {
        for (j = 0; j < m_Width; j ++)
        {
            lpSrc = m_lpDIBBits + 40 + i * m_LineBytes + j * 3 + 2;
            m_RCount[*(lpSrc)]++;
            r = (*lpSrc) * 0.299;
            lpSrc = m_lpDIBBits + 40 + i * m_LineBytes + j * 3 + 1;
            m_GCount[*(lpSrc)]++;
            g = 0.587 * (*lpSrc);
            lpSrc = m_lpDIBBits + 40 + i * m_LineBytes + j * 3 + 0;
            m_BCount[*(lpSrc)]++;
            b = 0.114 * (*lpSrc);
            liangdu = (int)(r + g + b);
            m_HCount[(int)liangdu]++;
        }
    }
    m_iIsDraging = 0;
    return TRUE;
}
void CDlgIntensity::OnPaint()
{
    CString str;
    LONG i;
    LONG lMaxCount = 0;
    m_selectCount = 0.0;
    m_selectRatio = 0.0;
    CPaintDC dc(this);
    CWnd* pWnd = GetDlgItem(IDC_COORD);
    CDC* pDC = pWnd->GetDC();
    pWnd->Invalidate();
    pWnd->UpdateWindow();
```

```
pDC->Rectangle(0,0,330,300);
CPen* pPenRed = new CPen;
pPenRed->CreatePen(PS_SOLID,1,RGB(255,0,0));
CPen* pPenBlue = new CPen;
pPenBlue->CreatePen(PS_SOLID,1,RGB(0,0, 255));
CPen* pPenGreen = new CPen;
pPenGreen->CreatePen(PS_DOT,1,RGB(0,255,0));
CGdiObject* pOldPen = pDC->SelectObject(pPenRed);
pDC->MoveTo(10,10);
pDC->LineTo(10,280);
pDC->LineTo(320,280);
str.Format("0");
pDC->TextOut(10, 283, str);
str.Format("50");
pDC->TextOut(60, 283, str);
str.Format("100");
pDC->TextOut(110, 283, str);
str.Format("150");
pDC->TextOut(160, 283, str);
str.Format("200");
pDC->TextOut(210, 283, str);
str.Format("255");
pDC->TextOut(265, 283, str);
for (i = 0; i < 256; i += 5)
{
    if ((i & 1) == 0)
    {
        pDC->MoveTo(i + 10, 280);
        pDC->LineTo(i + 10, 284);
    }
    else
    {
        pDC->MoveTo(i + 10, 280);
        pDC->LineTo(i + 10, 282);
```

```
        }
    }
    pDC->MoveTo(315,275);
    pDC->LineTo(320,280);
    pDC->LineTo(315,285);
    pDC->MoveTo(10,10);
    pDC->LineTo(5,15);
    pDC->MoveTo(10,10);
    pDC->LineTo(15,15);
    switch(m_colorSelect)
    {
    case 0:     for (i = m_iLowGray; i <= m_iUpGray; i ++)
                {
                    if (m_HCount[i] > lMaxCount)
                    {
                        lMaxCount = m_HCount[i];
                    }
                    m_selectCount += m_HCount[i];
                }break;
    case 1:     for (i = m_iLowGray; i <= m_iUpGray; i ++)
                {
                    if (m_RCount[i] > lMaxCount)
                    {
                        lMaxCount = m_RCount[i];
                    }
                    m_selectCount += m_RCount[i];
                }break;
    case 2:     for (i = m_iLowGray; i <= m_iUpGray; i ++)
                {
                    if (m_GCount[i] > lMaxCount)
                    {
                        lMaxCount = m_GCount[i];
                    }
                    m_selectCount += m_GCount[i];
```

```
                }break;
    case 3:     for (i = m_iLowGray; i <= m_iUpGray; i ++)
                {
                    if (m_BCount[i] > lMaxCount)
                    {
                        lMaxCount = m_BCount[i];
                    }
                    m_selectCount += m_BCount[i];
                }break;
    }
    m_selectRatio = 100 * m_selectCount / (m_Height * m_Width);
    UpdateData(FALSE);
    pDC->MoveTo(10, 25);
    pDC->LineTo(14, 25);
    str.Format("%d", lMaxCount);
    pDC->TextOut(11, 26, str);
    pDC->SelectObject(pPenGreen);
    pDC->MoveTo(m_iLowGray + 10, 25);
    pDC->LineTo(m_iLowGray + 10, 280);
    pDC->MoveTo(m_iUpGray + 10, 25);
    pDC->LineTo(m_iUpGray + 10, 280);
    pDC->SelectObject(pPenBlue);
    if (lMaxCount > 0)
    {
        for (i = m_iLowGray; i <= m_iUpGray; i++)
        {
            pDC->MoveTo(i + 10, 280);
            switch(m_colorSelect)
            {
            case 0:     pDC->LineTo(i + 10, 281 - (int) (m_HCount[i] * 256 /
lMaxCount));break;
            case 1:     pDC->LineTo(i + 10, 281 - (int) (m_RCount[i] * 256 /
lMaxCount));break;
            case 2:     pDC->LineTo(i + 10, 281 - (int) (m_GCount[i] * 256 /
```

```
lMaxCount));break;
                case 3:         pDC->LineTo(i + 10, 281 - (int) (m_BCount[i] * 256 /
lMaxCount));break;
                }
            }
        }
        pDC->SelectObject(pOldPen);
        delete pPenRed;
        delete pPenBlue;
        delete pPenGreen;
        pWnd->ReleaseDC(pDC);
}
```

附录 5

聚类过程实现中主要程序代码。

```
void CDlgIntensity::OnOK()
{
    m_pSectdata = new double[m_sect];
    int* temp, i,j,k;
    double total = 0;
    double ratio;
    i = m_colorCombo.GetCurSel();
    char *type;
    switch(i)
    {
    case 0: temp = m_HCount; type = "亮度"; break;
    case 1: temp = m_RCount; type = "红"; break;
    case 2: temp = m_GCount; type = "绿"; break;
    case 3: temp = m_BCount; type = "蓝"; break;
    }
    for(j = 0; j < m_sect; j++)
        {
            for(k = 256 / m_sect * j; k < 256 / m_sect * (j + 1); k++)
            {
```

```
                total += temp[k];
            }
            ratio = total / (m_Height * m_Width);
            m_pSectdata[j] = ratio;
            total = 0;
            ratio = 0;
        }
        CDialog::OnOK();
}
void CwangfeiView::OnViewCluster()
{
    int i,j,k;
    double temp = 0;
    for(i = 0; i < m_ImageCount - 1; i ++)
    {
        for(j = i + 1; j < m_ImageCount; j++)
        {
            for(k = 0; k < m_SectCount; k++)
            {
                temp += (m_RatioData[i][k] - m_RatioData[j][k]) *
(m_RatioData[i][k] - m_RatioData[j][k]);
            }
            m_MatrixData[j][i] = sqrt(temp);
            temp = 0;
        }
    }
    char FileNameString[] = " ";
    char FilterString[] = "Text File|*.txt|";
    CString m_target_file;
    CFileDialog filedlg(TRUE, NULL,
        (LPSTR)FileNameString,
        OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT |
OFN_ALLOWMULTISELECT,
        (LPSTR)FilterString);
```

```
if(filedlg.DoModal() == IDOK)
{
    m_target_file=filedlg.GetPathName();
    FILE *stream;
    stream = fopen(m_target_file, "a+");
    for(i = 0; i < m_ImageCount; i++)
    {
        for(j = 0; j < m_ImageCount; j++)
        {
        fprintf(stream, "%.4lf\t", m_MatrixData[i][j]);
        }
        fprintf(stream,"\n");
    }
    fprintf(stream,"\n");
double min;
 int  imin,jmin;

 for(k = 0; k < m_ImageCount - 1; k++)
 {
    min = 10;
    for(i = 1; i < m_ImageCount; i++)
    {
        for(j = 0; j <i; j++)
        {
            if(m_MatrixData[i][j] < min)
            {
                min = m_MatrixData[i][j];
                imin = i;
                jmin = j;
            }
        }
    }
    m_Result[k][0] = imin;
    m_Result[k][1] = jmin;
```

```
            m_Result[k][2] = m_MatrixData[imin][jmin];
            for(i = 0; i < jmin; i++)
            {
                if(m_MatrixData[imin][i] < m_MatrixData[jmin][i] &&
m_MatrixData[jmin][i] < 10.0)
                {
                    m_MatrixData[jmin][i] = m_MatrixData[imin][i];
                }
            }
            for(j = 0; j < imin; j++)
            {
                m_MatrixData[imin][j] = 10;
            }
            for(i = imin + 1; i < m_ImageCount; i++)
            {
                if(m_MatrixData[i][imin] < m_MatrixData[i][jmin] &&
m_MatrixData[i][jmin] < 10.0)
                {
                    m_MatrixData[i][jmin] = m_MatrixData[i][imin];
                }
                m_MatrixData[i][imin] = 10.0;
            }
            for(i = 0; i < m_ImageCount; i++)
            {
                for(j = 0; j < m_ImageCount; j++)
                {
                fprintf(stream, "%.4lf\t", m_MatrixData[i][j]);
                }
                fprintf(stream,"\n");
            }
            fprintf(stream,"\n");
        }
        for(i = 0; i < m_ImageCount - 1; i++)
            {
```

```
                fprintf(stream,                    "(%3.0lf\t%3.0lf)\t\t%.4lf\n",
m_Result[i][0] ,m_Result[i][1], m_Result[i][2]);
            }
            fprintf(stream,"\n");
            fclose(stream);
        ::ShellExecute(NULL,NULL,m_target_file,NULL,NULL,SW_SHOWNORMAL
);
        }
}
```

致 谢

本论文是在导师何继善院士的悉心指导下完成的，论文从选题到完成的整个过程中，得到了何先生的热情帮助和精心指导。先生严谨的治学态度，渊博的专业知识，敏锐的学术眼光，精益求精的精神让我终身难忘，并将对我以后的学习和工作产生极大地促进作用。在这里，我表示衷心的感谢。

需要特别感谢是熊平教授，在我论文前期工作中，做了大量指导工作。三年来，赵于前老师、瓮晶波老师在我的生活和学习中给了很多的关心和帮助，在这里一并表示我真诚的谢意。

还要感谢湘潭华鉴科技有限公司让我有机会全程参与了一次组织微阵列的制备工作，感谢那里的病理学专家许医生在微阵列图片病理识辨上给予的指导。为了研究聚类标准，我走访了湘雅附一、附三医院，感谢那些给出各类意见的血液科和病理科医生们。

三年来，生物医学 03 级研究生一直是一个很好的团队，我要感谢每一个成员对我的帮助。也谢谢我的师弟王凯，他在我做论文期间帮我做了很多事情。

最后我要说，非常感谢我的父母，是他们省吃俭用，几乎把他们这半辈子全部的收入都送进了学校，为的是让我受到良好的教育。感谢天下所有为子女辛劳的父母们，祝他们生活幸福、健康长寿。

攻读硕士学位期间主要研究成果

[1] 王飞，熊平，梁红波，等. 造血干细胞护送箱的设计. 中国医学工程，2006，14（1）：106～108

[2] 王飞，罗东礼，汤井田，等. 一种基于图的交互式目标分割算法. 计算机工程与应用，2006 年 11 月刊出

[3] 梁红波，陈一平，王飞. C8051F020 单片机及其在双频激电仪中的应用. 企业技术开发，2006，4：37～39