

时序数据分析



时序数据是指通过一系列时间点上的观测获取到的数据,在社会生产和生活中广泛存在。本章首先介绍时序数据的基本概念和常见的一些分析方法,然后以脑电信号为例,介绍时序数据的预处理、特征分析以及分类等过程和方法。脑电图(electroencephalogram, EEG)是一种使用电生理指标记录大脑活动的方法,因其无创、快速和成本低等特点在医学临床、脑科学研究中被广泛使用,是一种具有独特性的时间序列数据。本章介绍了常用的时频分析、非线性和网络分析等信号处理方法,并结合实例对这些方法的基本原理和意义进行了阐述。

3.1 时序数据简介

通过一系列时间点上的观测来获取数据是司空见惯的活动。在商业上,我们会观测周利率、日股票收盘价、月价格指数、年销售量等。在气象上,我们会观测每天的最高温度和最低温度、年降水与干旱指数、每小时的风速等。在农业上,我们会记录每年作物和牲畜产量、土壤侵蚀、出口销售等方面的数字。在生物科学上,我们会观测每毫秒脑电活动的状况。实际上,需要研究时间序列的领域是难以罗列的。本节将介绍时序数据的概念和常见分析方法。

3.1.1 什么是时序数据

时序数据是指时间序列数据,时间序列数据是同一指标按时间顺序记录的数据列。在同一数据列中的各个数据必须是同口径的,要求具有可比性。时序数据可以是时期数,也可以是时点数。时期数是反映现象在一段时间内发生的总量。例如,某种产品的产量、职工工资总额、商品销售额、投资总额均是时期数。时期数是通过在一定时期内事物的数量进行连续登记并累计加总得到的。时点数是表明事物总体在某一时刻上的数量状态。例如,人口数、商品库存量、固定资产价值、设备台数都是时点数。时点指标的数值是通过事物在某一时刻上数量的登记,将同一时刻上各部分数量加总得到的。

时间序列数据分析的目的一般有两个方面:一是认识产生观测序列的随机机制,即建立数据生成模型;二是基于序列的历史数据,也许还要考虑其他相关序列或因素,对序列未来的可能取值给出预测或预报。

3.1.2 时序数据的常见分析方法

不同种类的时间序列数据由于其来源、采集方式、采集目的的不同,呈现出各种形式,其特点也各不相同,也因此产生了各种时序数据的分析方法。在分析数据前,需要进行预处理。常见的时序数据预处理过程包括:对缺失值的处理,如进行插值;查找离群点并处理;将非等间隔数据调整为等间隔;去除数据中的噪声;对数据进行归一化等。

预处理中还有一个重要步骤是对数据进行初步的检验,以确定后续选用适当的分析预测模型。这里的检验包括平稳性检验和纯随机性检验。平稳性检验可以断定数据是否具有平稳性,最直观的一种方法是图检验,即根据时序图和自相关图显示的特征作出判断;另一种方法是构造检验统计量进行假设检验。

在进行时序图检验时,根据平稳时间序列均值、方差为常数的性质,平稳序列的时序图应该显示出该序列始终在一个常数值附近随机波动,而且波动的范围有界、无明显趋势及周期特征。图 3-1 为平稳序列与非平稳序列的示意图。

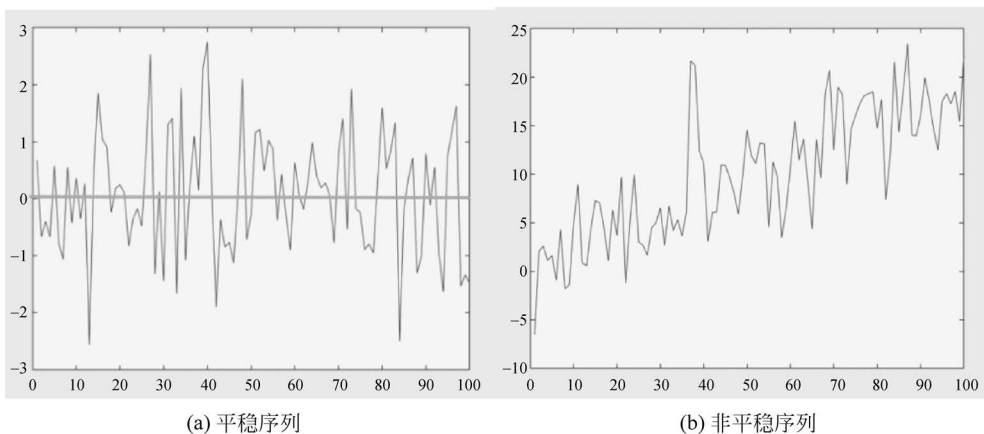


图 3-1 平稳序列与非平稳序列示意图

图检验是一种操作简便,运用广泛的平稳性判别方法,它的缺点是判别结论带有很强的主观色彩。所以最好能用统计检验的方法加以辅助判断。目前最常用的平稳性检验方法是单位根检验(unit root test)。

并不是所有的平稳序列都值得建模。只有那些序列值之间具有密切的相关关系,历史数据对未来的发展有一定影响的序列,才值得花时间去挖掘历史数据中的有效信息,用来预测序列未来的发展。如果序列值彼此之间没有任何相关性,那就意味着该序列是一个没有记忆的序列,过去的行为对将来的发展没有丝毫影响,这种序列称为纯随机序列。从统计的角度而言,纯随机序列是没有任何分析价值的序列。纯随机性检测也称为白噪声检测,是专门用来检测序列是否为纯随机序列的一种方法。如果判断出目标数据不是纯随机序列,那么就可以继续进行分析了。

在经过上述预处理后,对数据有了最初步的判断,也对其进行了规范化的处理,就以进行后续的分析了。传统的时间序列分析起源于经济领域,通常认为时间序列的成分可以分为四种:趋势、季节性或季节变动、周期性或循环波动,以及随机性或不规则波动。时间序列分析的一项主要内容就是把这些成分从时间序列中分离出来,并将它们之间的关系用一

定的数学关系式予以表达,而后分别进行分析。

对于短的或简单的时间序列,可用趋势模型和季节模型加上误差来进行拟合。

对于平稳时间序列,可用回归(autoregressive, AR)模型、滑动平均(moving average, MA)模型、自回归滑动平均模型(autoregressive moving average model, ARMA)或组合 ARMA 模型等来进行拟合。

对于非平稳时间序列则要先将观测到的时间序列进行差分运算,转化为平稳时间序列,再用适当模型去拟合这个差分序列。

对时间序列的均值、方差,以及自相关函数的计算等,都是对数据序列特征在时间域的观察和分析,这些被称为时域特征。

时间序列分析旨在发现序列的真实过程或者现象特征,如平稳性水平、周期性变化、振幅频率和相位等。其中,周期性变化、振幅频率和相位等属于时间序列的频域特征,对这些特征的研究属于频域分析。频域分析的主要方法包括频谱分析、功率谱分析等。

此外还有同时关注时域和频域特征的时频分析方法,主要有短时傅里叶变换、小波变换等方法。

3.2 脑电数据的获取与预处理

脑电信号是通过电极记录下来的脑电细胞群的自发性、节律性电活动,包含了大量的生理与病理信息,对其进行深入的研究有助于临床医生提高对大脑神经系统损伤病变诊断和检测的可靠性和准确性,同时对于脑疾病诊断和检测提供了有效的手段,所以脑电图检查在临床诊断中起着越来越重要的作用。本节介绍脑电数据的主要特点、获取方式,数据的读取和预处理流程。

3.2.1 脑电数据的特点和获取

脑电信号是由电极记录下来的大脑细胞群的自发性生物电活动,以电位为纵轴、时间为横轴将它以曲线的形式显示出来,就是脑电图。目前的脑电设备已基本实现电脑化,脑电信号被数字化后存于计算机中,通过屏幕来显示或在计算机控制下打印出来。

脑电信号具有在时间和空间分布上不断变化的特性,因此,它的电位(振幅)、时间(周期)及相位三者构成脑电图的基本特征。脑电信号的周期与物理学中正弦波的周期略有不同,它指的是一个波的波底到下一个波的波底之间的时间,用毫秒(ms)表示。每秒钟出现的周期的数目称为频率,用 Hz 表示。在脑电图上,除形态类似正弦波的波形外,还可见到由不同周期的脑波重叠在一起所构成的复合波。脑波的振幅通常是从波顶画一条垂直于基线的直线,并且与前后两个波的波底连线相交,此交点至波顶的距离称为该脑波的振幅,用微伏(μV)表示。采用这种计量方法的理由是,脑电图的基线通常不太稳定。这种波顶-波底的标记方式更可靠。脑波的振幅主要决定于脑内发生的电活动的强度和参考电极的选择。

实践中采用脑电信号采集设备来获取脑电数据。该设备通常包括电极帽、信号放大器、信号接收器和计算机等。数据采集时,受试者头戴电极帽,进行特定的实验活动,其脑电信号将经由信号放大器和接收器被计算机记录下来。图 3-2 为脑电信号采集的示意图。



图 3-2 脑电信号采集示意图

采集实验通常应选择在安静、避光和电磁干扰小的房间进行。电极帽上放置的电极直接接触头皮,是采集脑电信号的部件。临床使用的脑电图仪至少应有 8 个电极,此外还有 12、16、32 等多种规格型号。在认知研究中则一般使用 32、64、128 或 256 电极的脑电图仪。通常脑电图仪电极数目越多,所能获得的脑电时空信息也越丰富。但是,电极数越多,除了设备越加昂贵外,在使用时安装电极的时间也越长,信息处理的复杂度也相应增加,因此应根据具体情况作出合理的取舍。

电极的放置通常使用国际脑电图学会建议采用的标准电极安放法,即 10-20 系统法,其放置方法如图 3-3 所示。这种放置方法的特点是,头部电极的位置与大脑皮质的解剖学分区较为明确,电极的排列与头颅大小及形状成比例,在与大脑皮质凸面相对应的头部各主要区域均有电极放置。电极的命名是其位置的英文首字母,如 F 代表 frontal,即额叶区域。序号通常是用奇数代表左侧,偶数代表右侧。

脑电数据根据采集时被试的不同状态,可以分为自发脑电(无外界刺激)和诱发脑电(有外界刺激)。本章的案例以自发脑电为主。

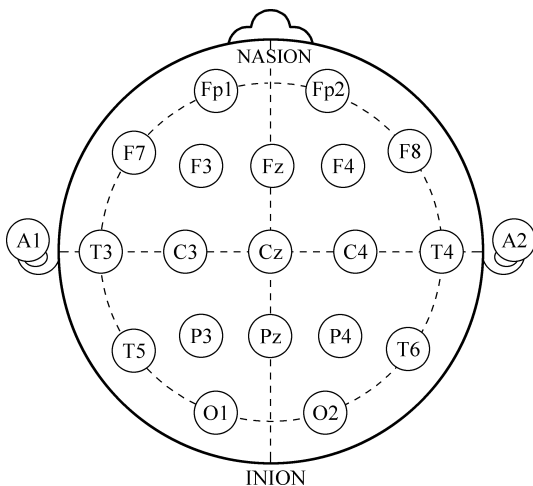


图 3-3 10-20 系统电极安放示意图

3.2.2 脑电数据读取与查看

脑电采集系统记录到的脑电信号可供后续分析和处理,这些数据被保存为特定的格式。不同设备厂商定义了各自的脑电数据格式,常见的如 EDF 格式, CNT 格式等。这些数据格式通常都包括标题和数据部分,其中,标题会包含一些一般信息(如患者标识、开始时间等),以及每个信号的技术规格(校准、采样率、过滤等),而数据部分就是各电极记录到的具体数值,以定义好的精度和格式进行存放。很多脑电设备厂商也有自己开发的分析系统,但不具有共性。

本章将以 MATLAB 软件为例,介绍脑电信号的分析方法。MATLAB 支持工具箱的开发和使用。对于脑电数据,EEGLAB 工具箱集成了对不同格式脑电数据的基本分析处理

功能,可以方便快速地完成预处理和时频分析等。

1. EEGLAB 的下载和配置

首先需要在计算机上安装 MATLAB 软件。在安装过程中,需要注意对 MATLAB 自带的工具箱进行选择,信号处理工具箱(Signal Processing Toolbox)需要被选中,否则将导致 EEGLAB 中的某些功能无法使用。

从 EEGLAB 官网: <https://scn.ucsd.edu/eeglab/download.php> (见图 3-4) 下载最新版 EEGLAB 工具箱,下载文件名为 eeglab_current.zip。

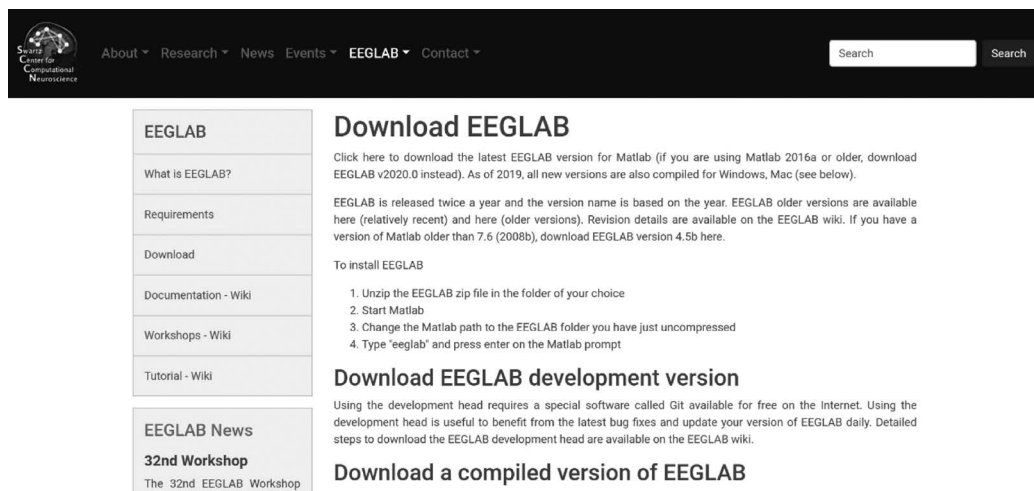


图 3-4 EEGLAB 下载页面

接下来对 EEGLAB 进行配置。将下载的文件解压缩并重命名为 eeglab,复制粘贴到 MATLAB 安装目录下的 toolbox 文件夹中,例如 D:\MATLAB2021a\toolbox。打开 MATLAB 软件,设置路径,单击“主页”→“设置路径”。在出现的界面中单击“添加并包含子文件夹”,选中刚刚添加的 eeglab 文件夹,注意最后要单击“保存”,如图 3-5 所示。

在 MATLAB 中输入命令 eeglab,若出现如图 3-6 所示界面,则说明路径设置成功。

若是没有设置成功,则需要更新工具箱缓存,单击“主页”→“预设”→“常规”→“更新工具箱路径缓存”,如图 3-7 所示。

配置完成后,可以进行数据的读取。

2. 数据的读取

EEGLAB 支持读取各种格式的脑电数据,如 ASCII、EDF、CNT 和 EGI 等。以 CNT 文件为例,单击“File”→“Import data”→“Using EEGLAB functions and plugins”→“From Neuroscan .CNT file”。根据要处理的数据格式选择相应的选项即可,如图 3-8 所示。

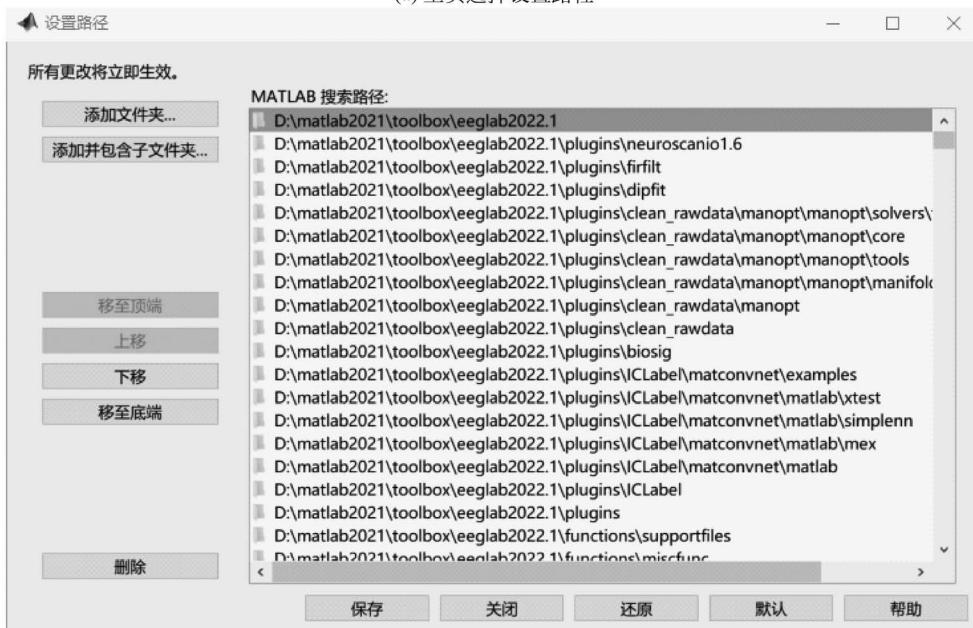
单击后会出现弹窗,如图 3-9 所示,点选“Autodetect”即可,其他根据需求填写,单击“Ok”。

然后出现弹窗,给加载的数据集命名,默认或者填入自己的命名,单击“Ok”。如图 3-10 所示。

成功读入文件后会出现如图 3-11 所示界面。



(a) 主页选择设置路径



(b) 添加文件夹

图 3-5 设置路径

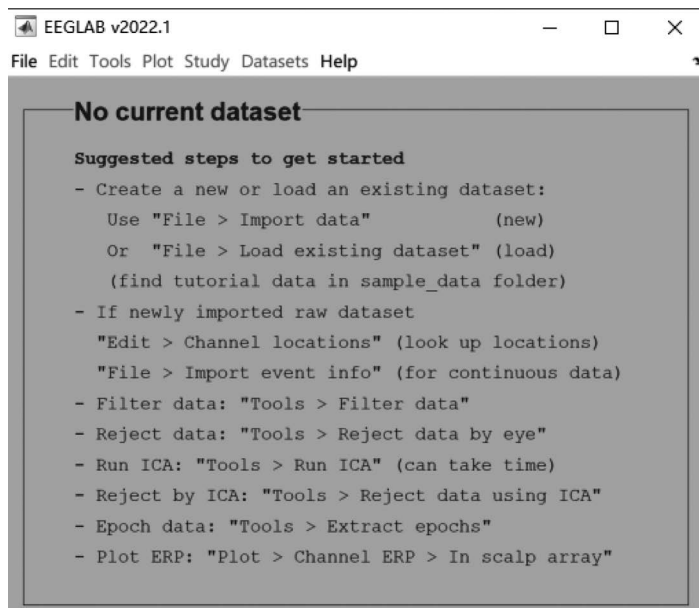


图 3-6 EEGLAB 成功启动



图 3-7 更新工具箱缓存

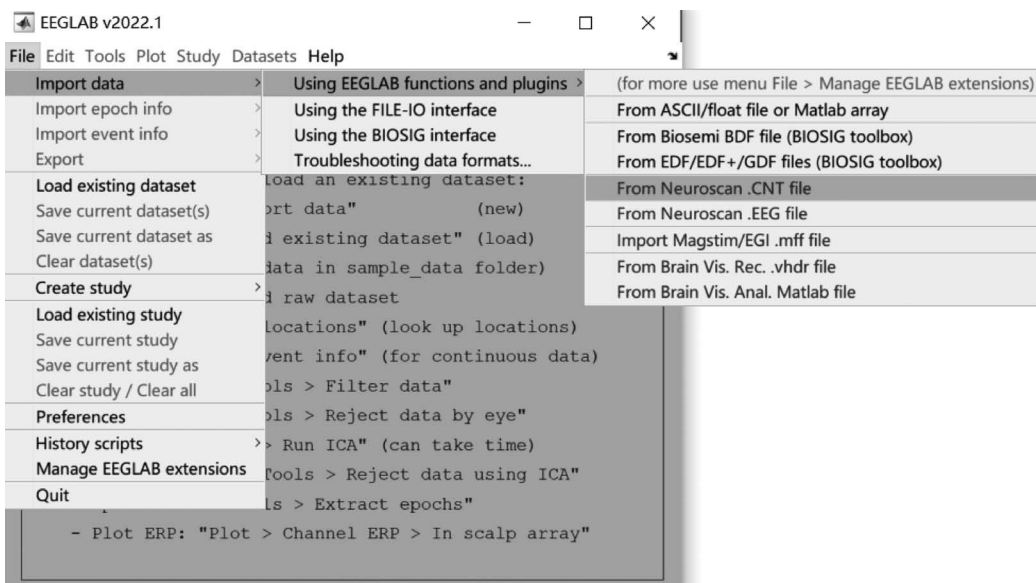


图 3-8 文件读取

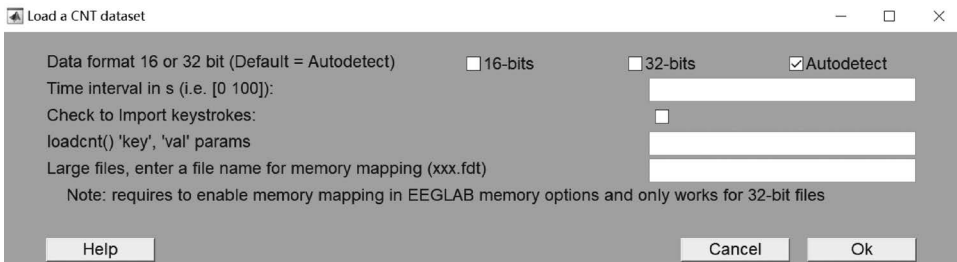


图 3-9 文件读入参数选择

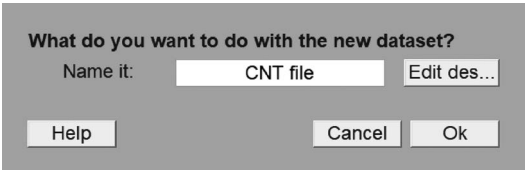


图 3-10 文件命名



图 3-11 文件读入成功

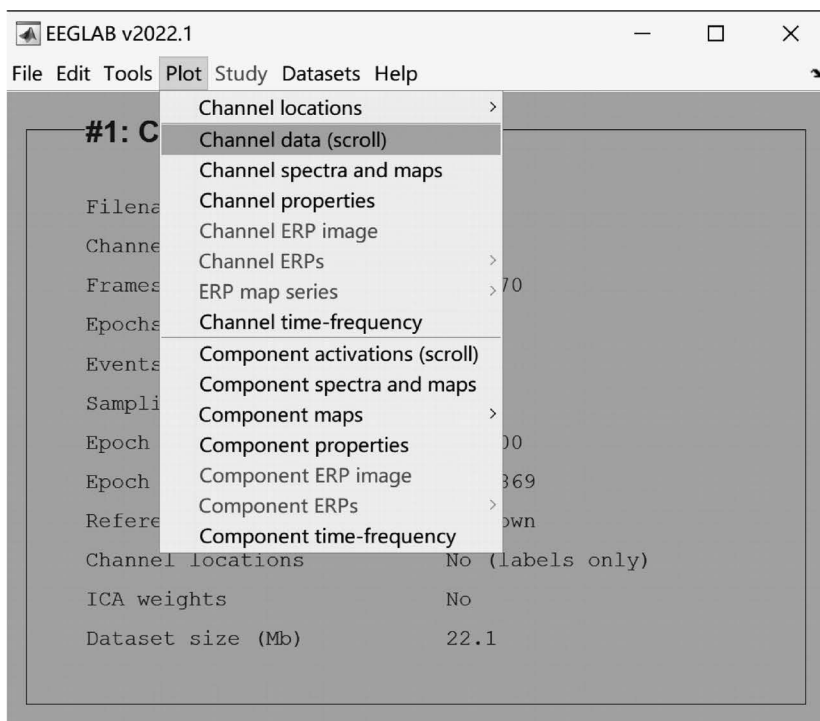
3. 查看数据

数据文件读入成功后,可以查看数据。单击“Plot”→“Channel data(scroll)”,查看每个通道的记录数据,如图 3-12 所示。

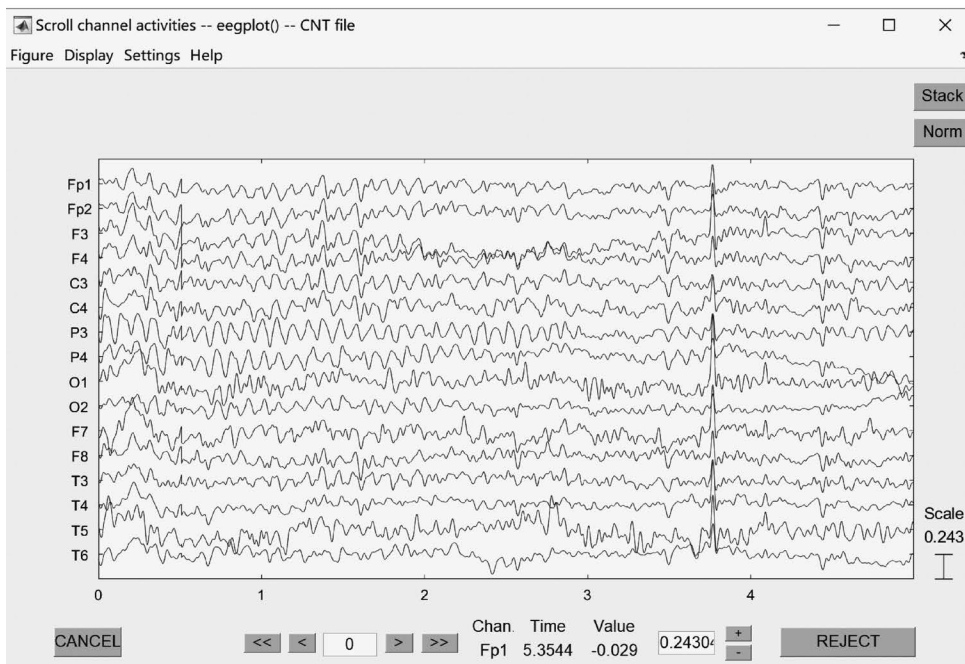
还可以查看电极位置。首先加载通道位置信息文件,单击“Edit”→“Channel locations”,如图 3-13 所示。

弹出如图 3-14 所示提示框,单击“Read locations”,读入相应的通道信息文件。

在图 3-14 所示界面中单击“Ok”后,在 EEGLAB 界面,单击“Plot”→“Channel locations”→“By name”,可以可视化各电极位置,如图 3-15 所示。



(a) 查看数据的操作



(b) 各通道的EEG数据

图 3-12 查看多通道的 EEG 数据

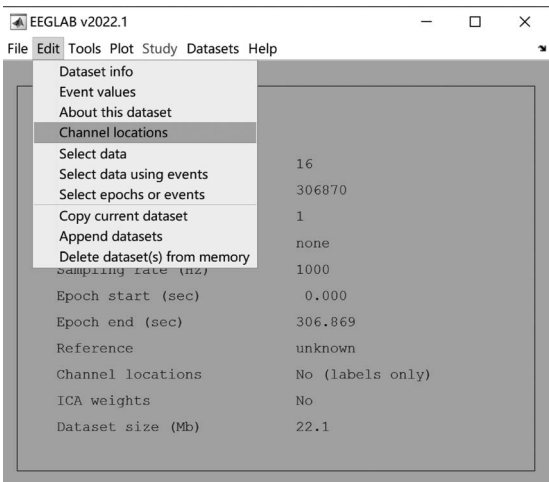


图 3-13 加载电极位置文件

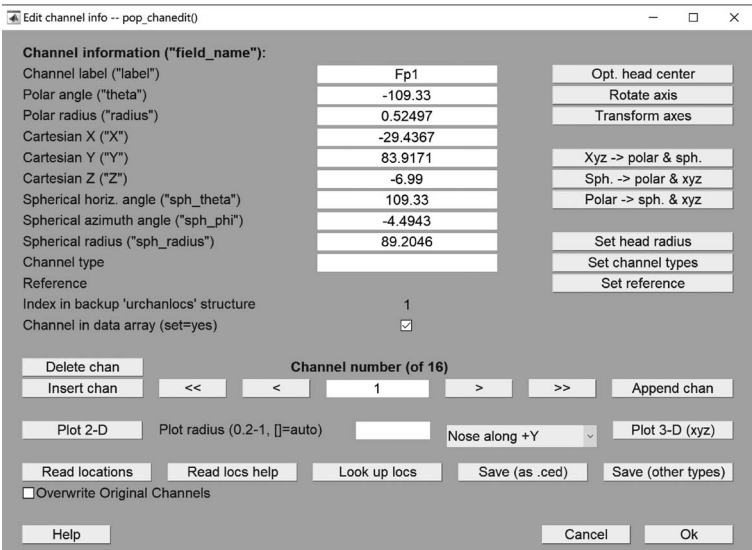
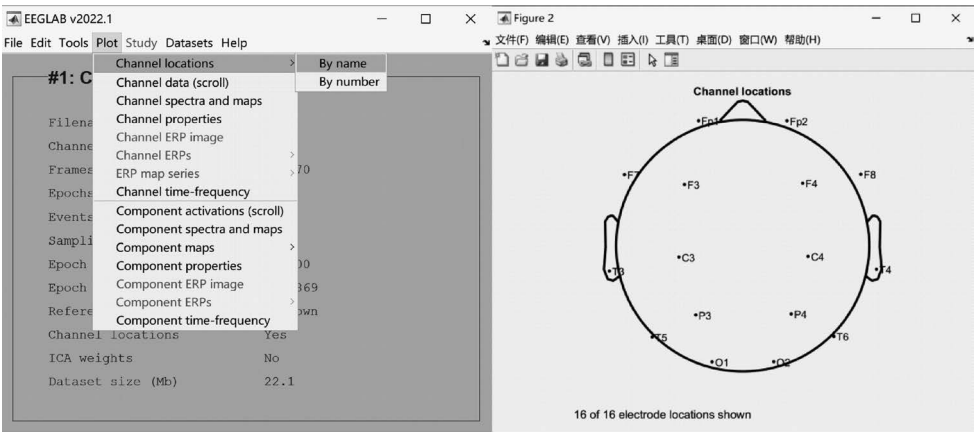


图 3-14 电极信息弹窗



(a) 查看电极位置操作

(b) 电极位置分布图

图 3-15 电极位置可视化

3.2.3 数据预处理

数据读入成功后开始进行数据的预处理,包括重采样、滤波、重参考、删除坏道、插值、去伪迹以及分段等操作,为后续的数据分析做好准备。

1. 重采样

采样率的单位是赫兹(Hz),1000Hz 代表着 1s 会采样 1000 次,有时候我们可能想要降低采样率,例如降到 500Hz,或者 250Hz,以减少数据量,提高计算速度。

具体步骤如下:单击“Tools”→“Change sampling rate”,如图 3-16 所示。

在图 3-17 所示弹窗中填入想要更改的采样率后,如 250,单击“Ok”即可。

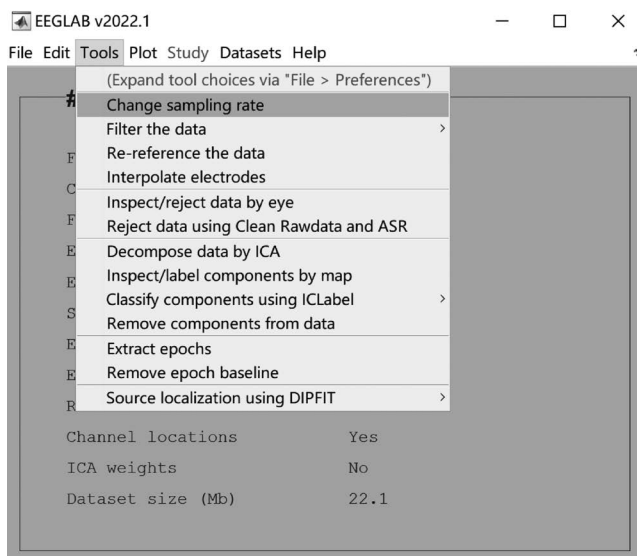


图 3-16 重采样操作

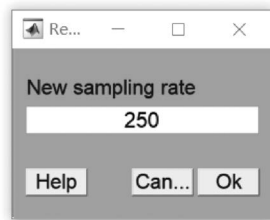


图 3-17 新采样率弹窗

2. 滤波

滤波是将信号中特定波段频率滤除的操作,是抑制和防止干扰的一项重要措施。对于脑电信号,国际上按照频率将其分成以下几类: δ 波,0.5~3Hz; θ 波,4~7Hz; α 波,8~13Hz; β 波,18~30Hz; γ 波,大于 31Hz。由此可见,我们关心的脑电信号大都处于 0.5~30Hz 的范围。因此,通过滤波操作可以去掉高频成分的干扰,尤其是 50Hz 的高频干扰,得到信噪比更高的信号。

EEGLAB 提供 5 种有限长单位冲激响应(finite impulse response,FIR)滤波器供选择,这种滤波器的特性适合 EEG 信号。在 EEGLAB 界面上,见图 3-18,单击“Tools”→“Filter the data”→“Basic FIR filter”,会出现图 3-19 所示弹窗,输入 0.5Hz 作为下边缘通带频率,30Hz 作为上边缘通带频率(可根据实际需求调整),然后单击“Ok”。

单击“Ok”后,如果在图 3-19 中勾选了“Plot frequency response”,就出现如图 3-20 所示弹窗,显示相应滤波器的频率响应的幅值和相位,从而可评估滤波器的性能。

根据自己的需要对滤波后的文件进行重命名、存储等操作,如图 3-21 所示。

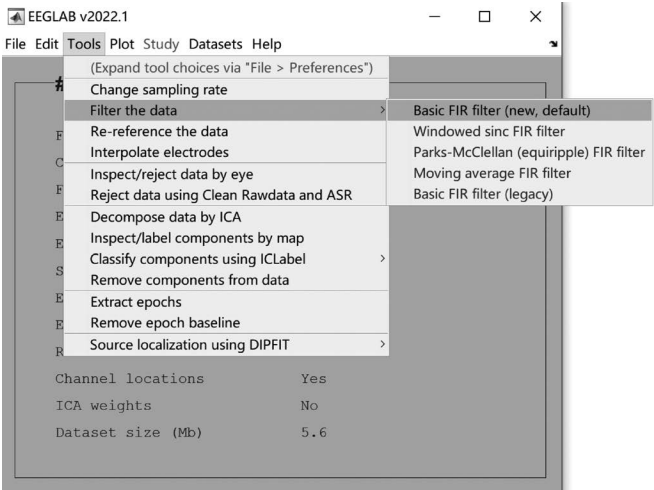


图 3-18 选择滤波器

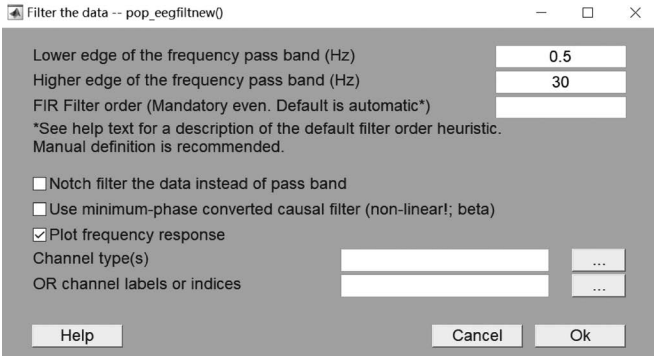


图 3-19 填写滤波器参数

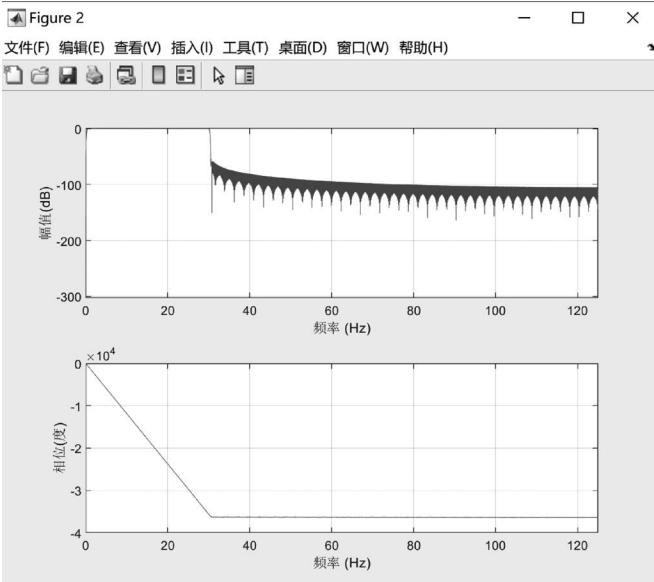


图 3-20 滤波器频率响应幅值和相位

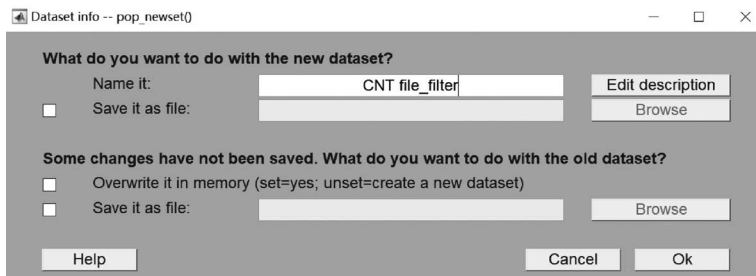


图 3-21 滤波后文件操作

单击“Ok”后,EEGLAB 就会重新打开经过滤波处理后的文件,如图 3-22 所示。可以发现,此时的数据波形少了很小毛刺,整体趋势也更加平稳,因为高频噪声和超低频的波动都被滤除了。

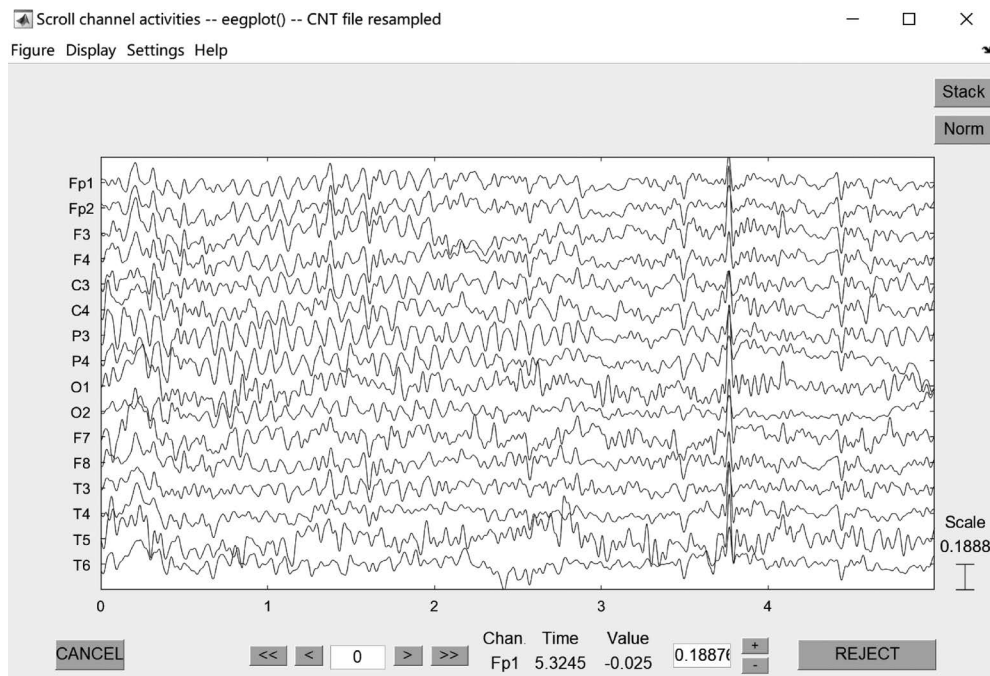


图 3-22 滤波后的数据

3. 重参考

脑电采集系统在采集数据的时候会有一个参考电极,记录的数据是相应电极与参考电极的电势差。不同的参考电极会影响数据的分析,因此在数据预处理时,会进行重参考以规范数据,方便不同实验数据之间的对比。一般情况下,实验采集时采用的参考电极有连接耳参考、中心电极参考等。

重参考的常见方法有双侧乳突参考、全脑平均参考、零参考等。此处以全脑平均为例,具体的操作见图 3-23,单击“Tools”→“Re-reference the data”。

在弹出窗口中,选择计算平均参考“Compute average reference”,如图 3-24 所示。

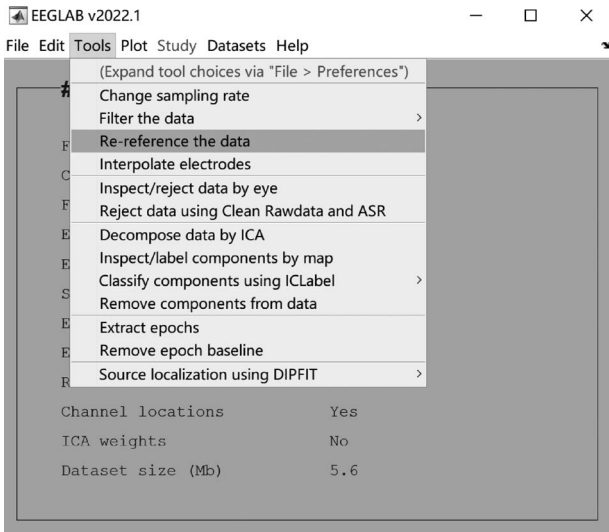


图 3-23 脑电数据重参考

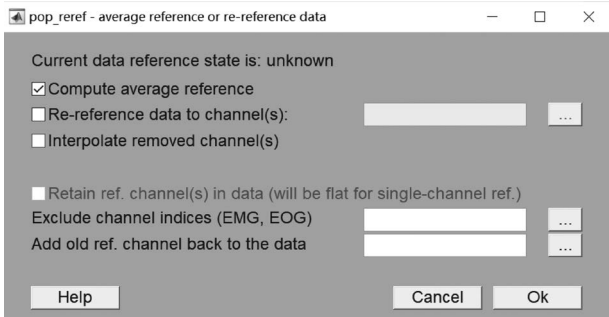


图 3-24 选择计算平均参考

4. 删除坏道

如果某些电极不需要或者信号质量太差,可以将其删除。具体步骤如图 3-25 所示,单击“Edit”→“Select data”。

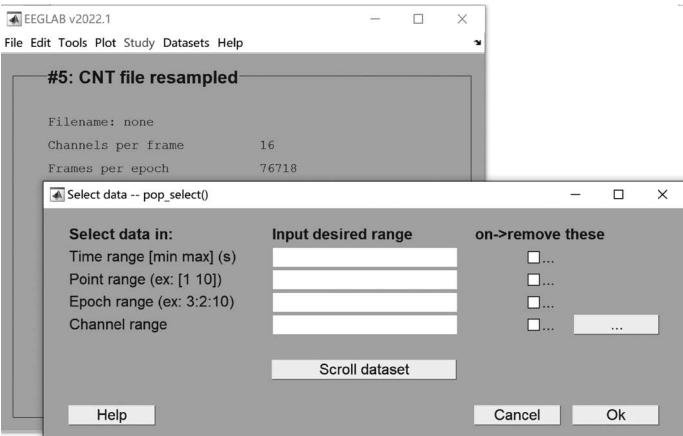


图 3-25 数据通道选择

在“Channel range”栏填写需要删除或者保留的电极,如图 3-26 所示。

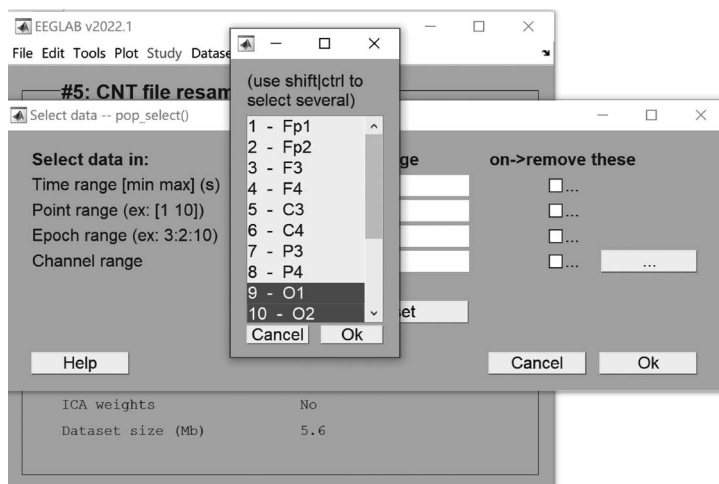


图 3-26 删除电极

然后选择存储方式,如“overwrite it in memory”,如图 3-27 所示。

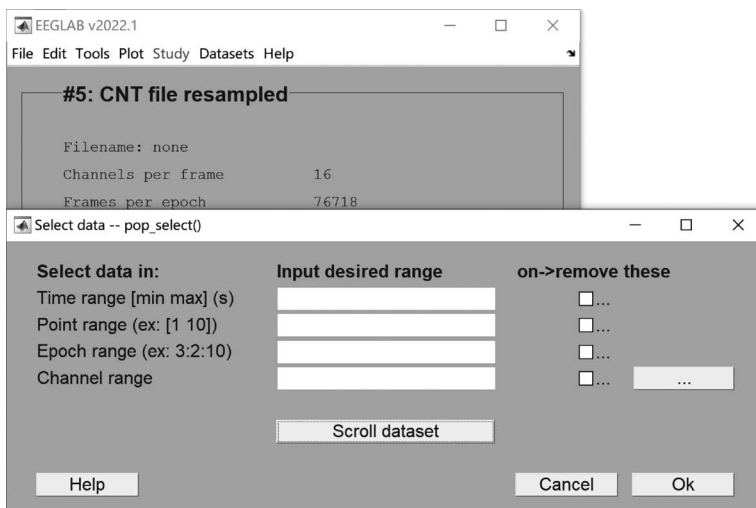


图 3-27 删除数据后文件的存储

5. 插值

对数据进行检查,如果发现某个需要的通道数据有缺损,如记录的时候出现了电极脱落等情况,导致数据不可用,可以用插值的方法进行校正。具体操作如图 3-28 所示,单击“Tools”→“Interpolate electrodes”。

在弹窗中单击“Select from data channels”,选择插值的通道,如图 3-29 所示。

如果不知道通道对应的数值名称是什么,可以通过单击“Plot”→“Channel locations”→“By name”画出电极通道图后,通过单击通道名字来查看通道数值,如图 3-30 所示。

另一种方法是直接编写代码来操作,其原理是利用缺失电极周围电极的平均值来对缺失通道进行插值。比如图 3-30 中,如果 F4 电极数据缺失,则取 Fp2、F8 和 C4 电极的平均值作为 F4 电极的数据。

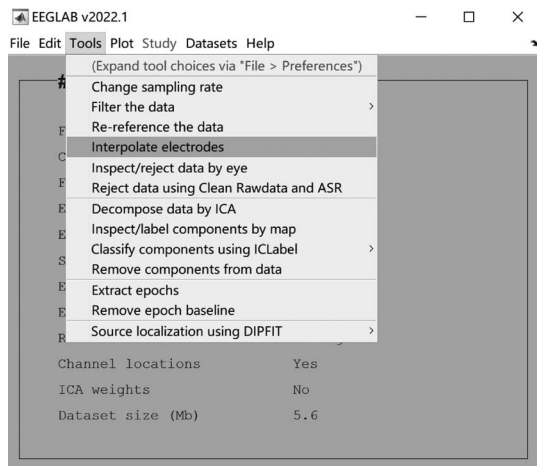


图 3-28 插值

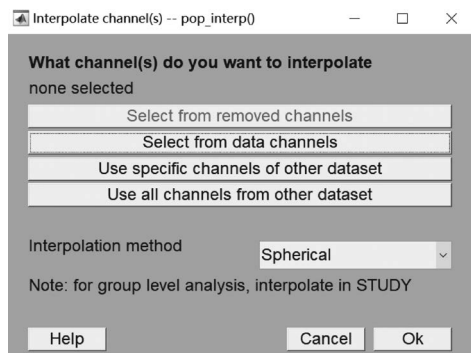


图 3-29 选择要插值的通道

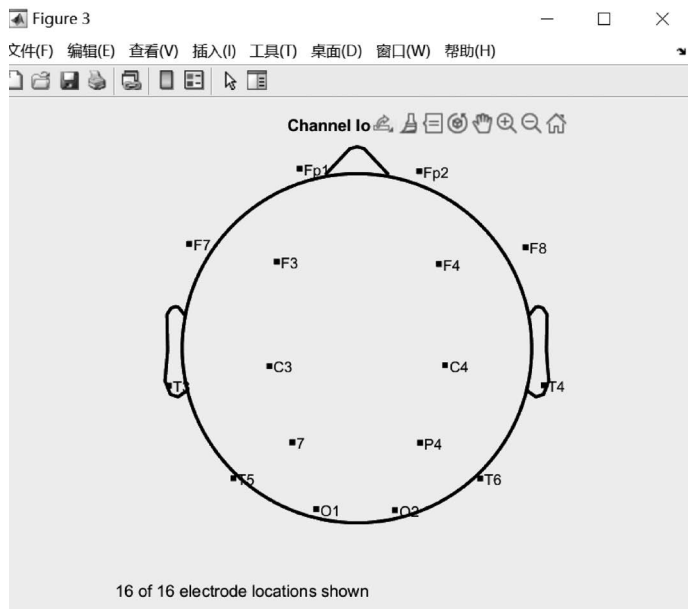


图 3-30 16 电极的位置分布

6. 伪迹去除

脑电数据有很多伪迹,这些脑电数据记录过程中出现的非脑电干扰,给脑电的分析带来了极大的困难。因此,在数据记录的过程中要注意减少伪迹的产生,同时对于记录的数据要进行判别,找出明显的伪迹并进行去除。

脑电数据的伪迹主要来自两个方面,其一是数据采集受试者的眼动、脉搏、吞咽、咳嗽等动作,因为肌电信号的幅度比脑电信号大得多,所以一旦受试者有各种动作,就很难采集到高质量的脑电信号。其二是仪器,仪器故障、电极接触不良或者电线晃动、空间的电磁干扰等。如何去除伪迹是脑电信号处理领域一个经典的研究问题。这里采用两种方式进行伪迹去除,一种是人工去除,另一种是利用独立成分分析(independent component analysis,

ICA)方法。

人工进行伪迹去除,一般是通过数据窗口观察数据,将有明显的伪迹数据段删除。

在信号处理中,ICA 是一种用于将多元信号分离为加性子分量的计算方法。它是通过假设子分量是非高斯信号,并且在统计上彼此独立来完成的。如果假设脑电信号与伪迹是彼此独立的,那么就可以用 ICA 来对真正的脑电信号和伪迹进行分离,从而达到去除伪迹的目的。

EEGLAB 中提供了用 ICA 来进行伪迹去除的功能。首先使用 ICA 分解数据,具体操作如图 3-31 所示,单击“Tools”→“Decompose data by ICA”。

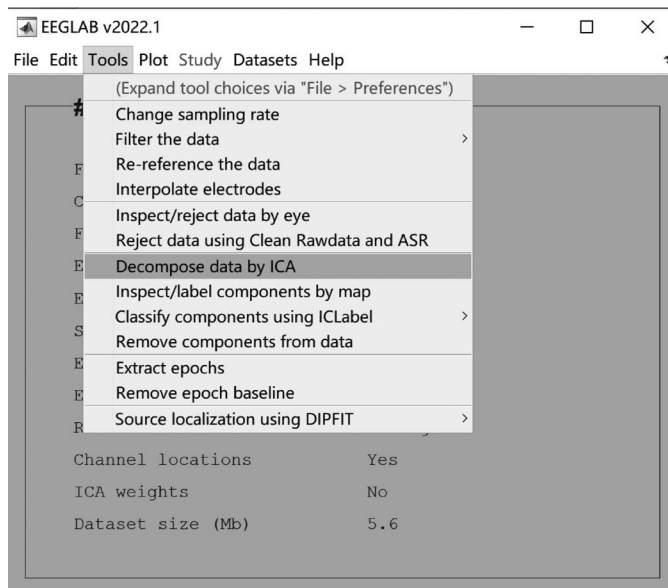


图 3-31 对数据进行 ICA 分解

在弹窗中选择默认算法“runica”即可,单击“Ok”运行,见图 3-32。

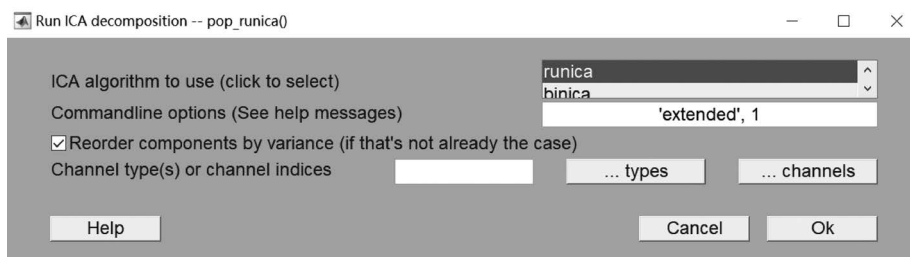


图 3-32 ICA 分解参数选择

此时,在命令行窗口,会出现如图 3-33 所示的提示信息。运行的时间可能有点长,注意不要单击“Interrupt”,耐心等待。

ICA 分解完成后,可以对分解得到的各独立成分进行可视化。具体操作如图 3-34 所示,单击“Plot”→“Component maps”→“In 2-D”。

在弹窗中填写成分数量,见图 3-35,如填写“1 : 16”,即画出第 1~16 个成分。

单击“Ok”后,得到如图 3-36 所示的图,该图展示了各独立成分在头皮中的分布情况。

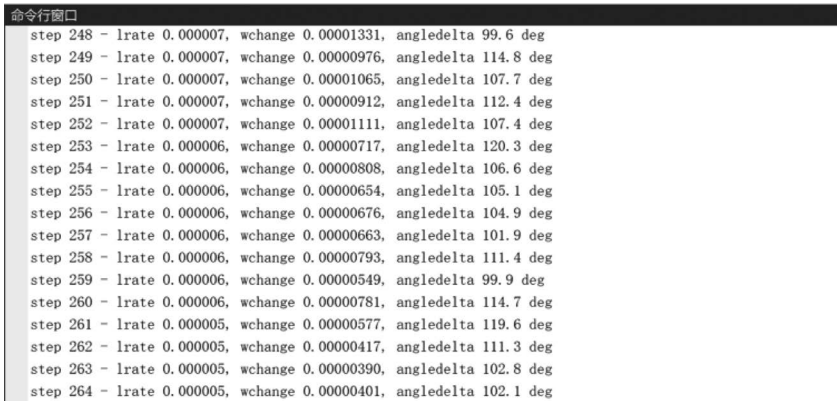


图 3-33 ICA 分解时命令行窗口的提示信息

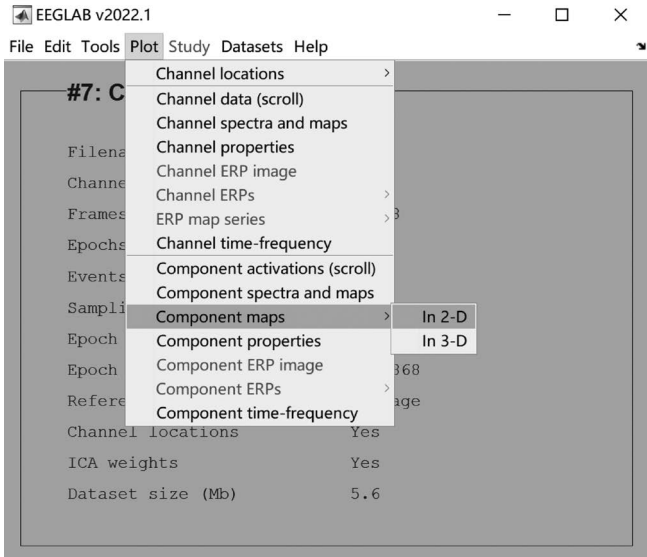


图 3-34 画 ICA 成分图

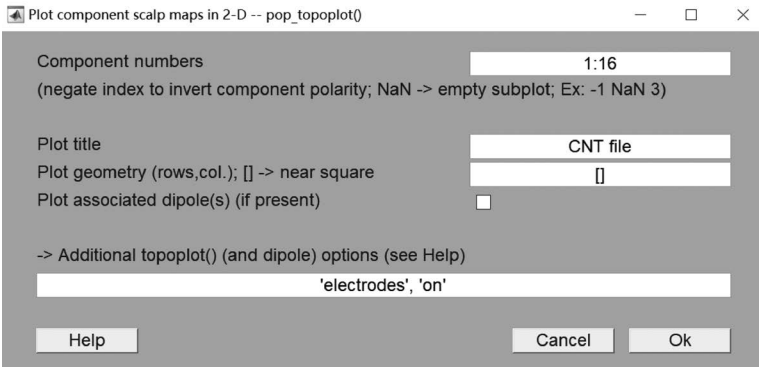


图 3-35 画图 ICA 成分选择

有经验的研究者根据 ICA 成分的特征,可以判断哪些成分是伪迹,应该去除。但对于初学者来说,难度很大。EEGLAB 为伪迹判断提供了辅助。首先通过单击“Tools”→

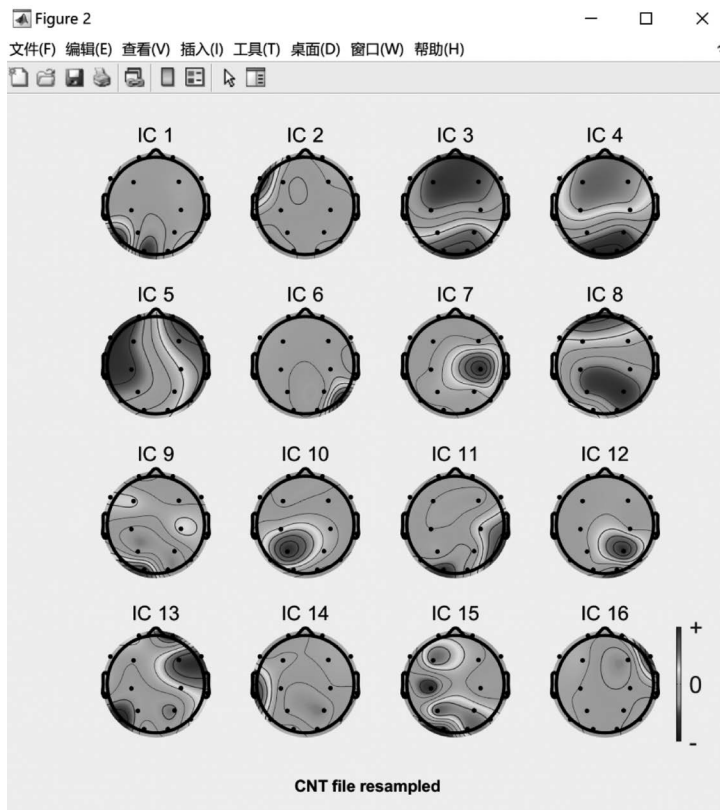


图 3-36 ICA 成分示例

“Classify components using ICLabel”→“Label components”查看各独立成分具体情况,如图 3-37 所示。

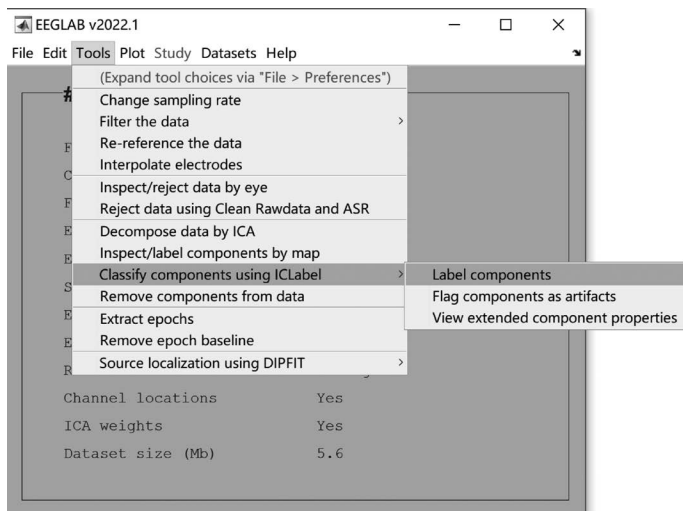


图 3-37 独立成分标记

在弹出窗口中单击“Ok”,如图 3-38 所示。

在弹出窗口中可以填写需要看的成分的个数,如图 3-39 所示,这里填写“1 : 16”。

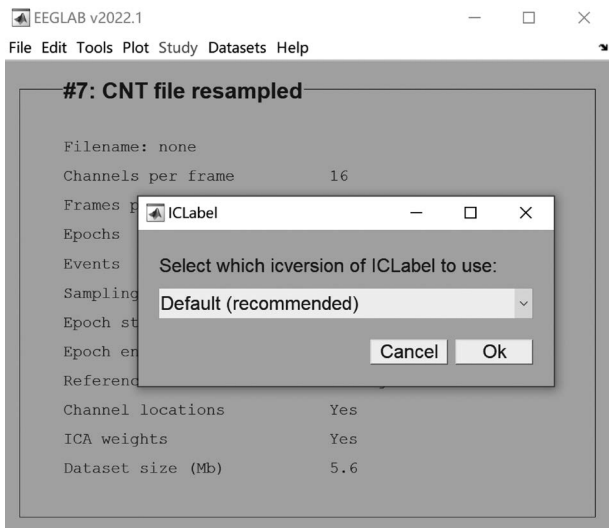


图 3-38 独立成分标记方法选择

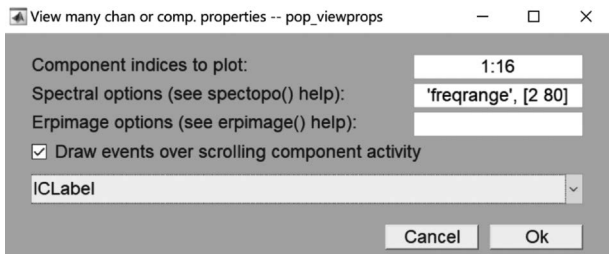


图 3-39 独立成分画图的参数选择

单击“Ok”之后,得到显示结果,如图 3-40 所示。该结果显示了每个成分的提示信息。其中,“Eye”表示眼动相关成分。

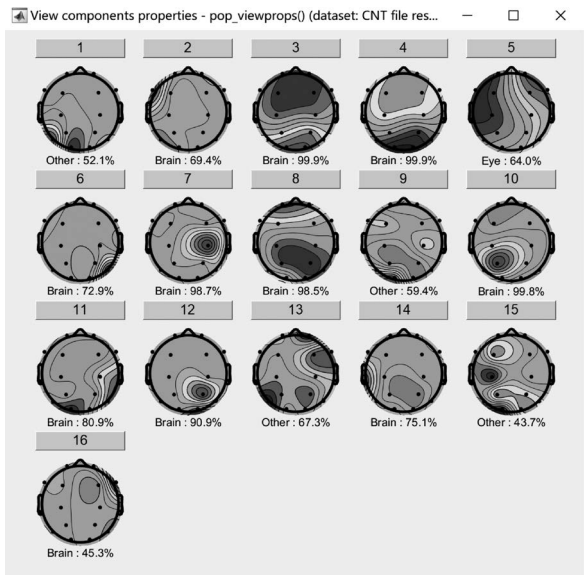


图 3-40 带标记的独立成分显示

单击数字按钮可以浏览具体信息,如在图 3-40 所示界面中单击“5”,会弹出图 3-41 所示界面。

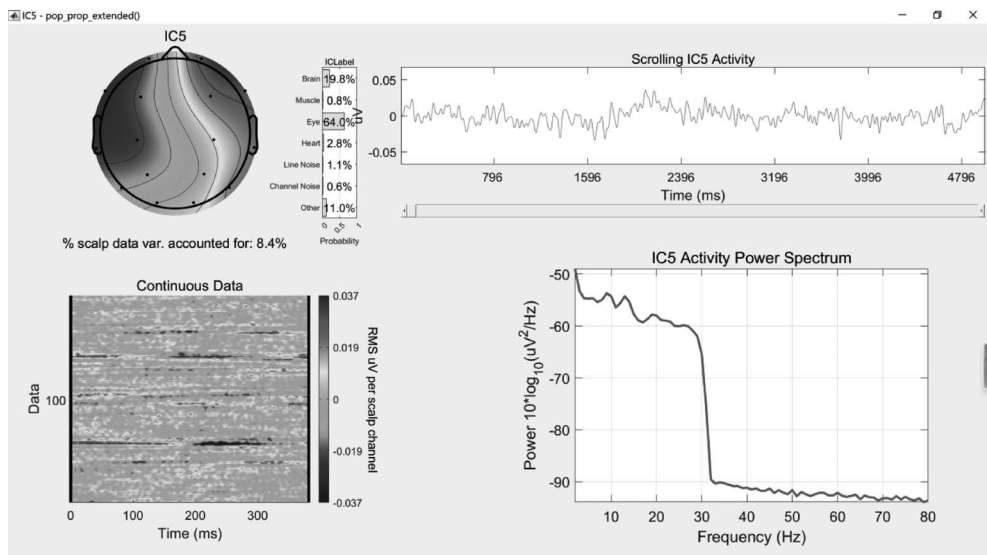


图 3-41 独立成分 5 的细节展示

可以看到 IC5 的成分主要是眼动,还有一些通道噪声和其他,因此可以去除。

由此,最终决定删除 IC5。操作为:单击“Tools”→“Remove components from data”,如图 3-42 所示。

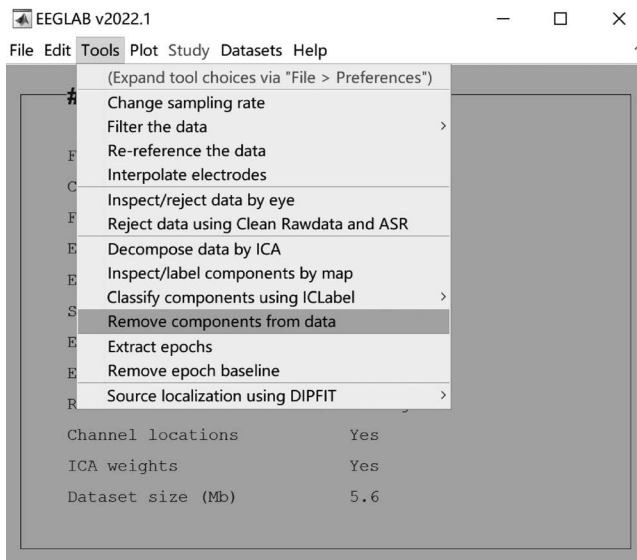


图 3-42 删除成分

在弹窗中填入要删除的成分,如“5”,见图 3-43。

然后会弹出确认窗口,单击“Accept”以后,所选成分被去除,如图 3-44 所示。

删除完成后,需要保存新的数据,如图 3-45 所示。至此,利用 ICA 方法去除伪迹的操作完成。

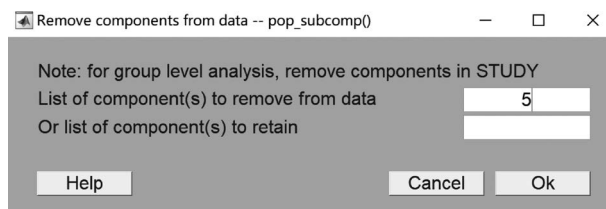


图 3-43 填入删除成分列表

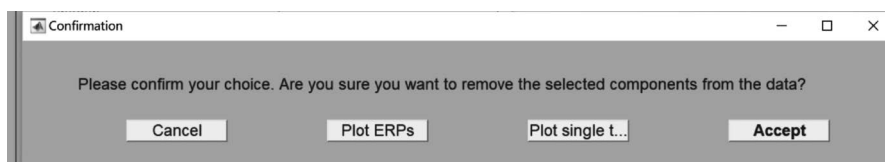


图 3-44 独立成分删除确认弹窗

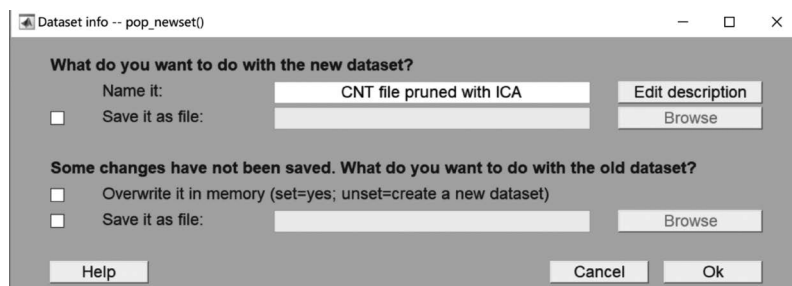


图 3-45 新数据命名和保存

注意：每个数据去除成分时需要非常慎重,不建议删除过多成分。同时,并非每个数据都必须要去除其中的成分,如果标签 Eye 的成分比例较低,如低于 50%,那么建议保留。

去除伪迹之后的效果如图 3-46 所示。

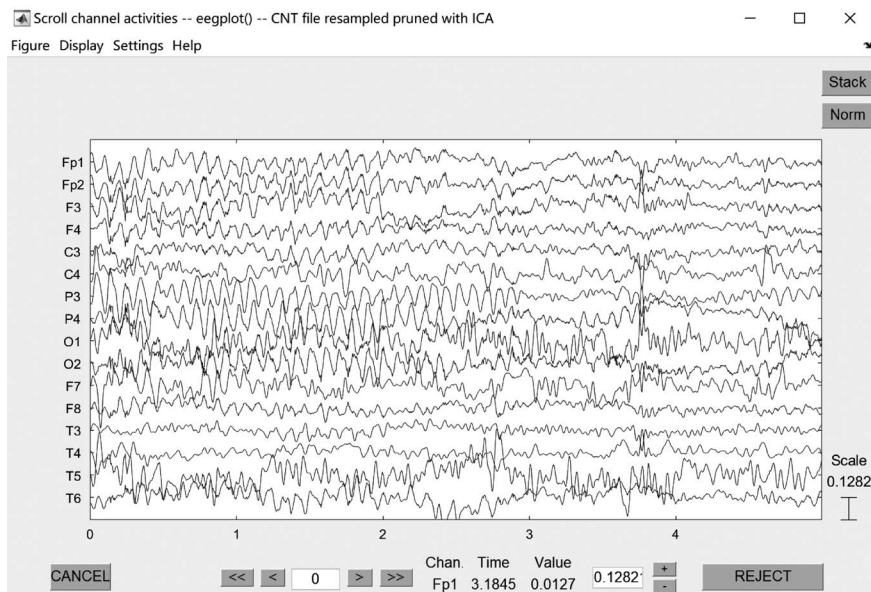


图 3-46 去除伪迹后的数据示意图

7. 分段和移除基线值

采集记录到的脑电数据通常持续时间都比较长,所以通常需要进行分段。尤其对于事件相关电位(event-related potential,ERP)数据,必须按照单次任务的起始时间和结束时间进行分段,才能进行后续的分析。

对于有事件标记的数据,分段操作如图 3-47 所示,单击“Tool”→“Extract epochs”。

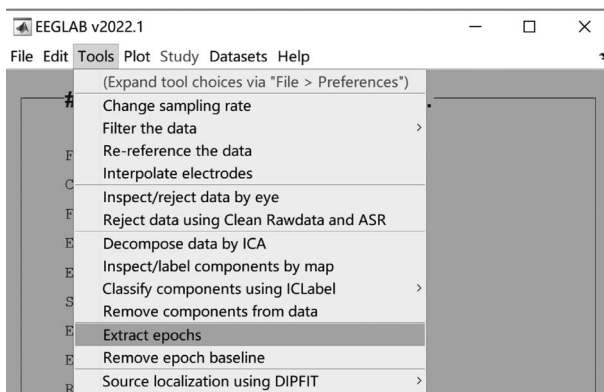


图 3-47 分段操作

在弹窗中根据需要设置参数,保留默认的时间限制(从时间锁定事件之前的 1 秒到时间锁定事件之后的 2 秒),单击“Ok”,如图 3-48 所示。

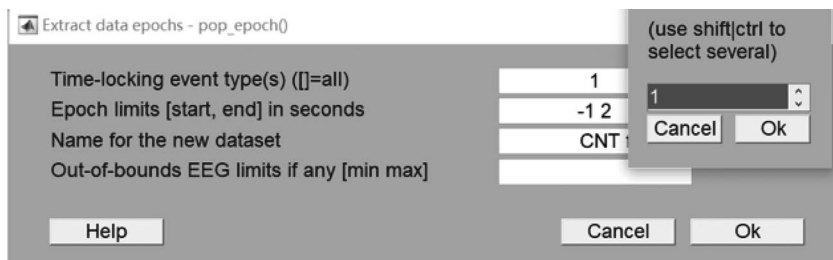


图 3-48 分段参数选择

根据需要,保存并重命名提取的数据文件,单击“Ok”即可,如图 3-49 所示。

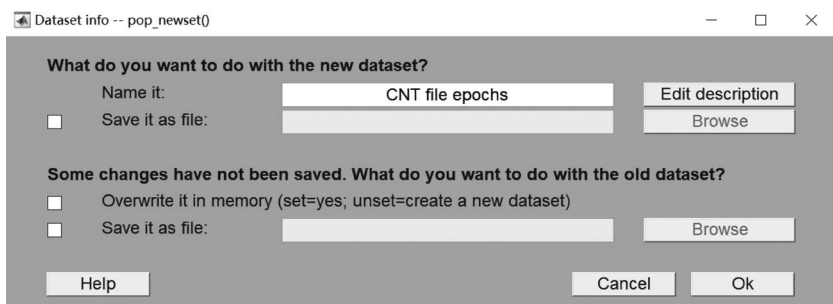


图 3-49 保存分段后的数据

如果数据中不包含事件标记,那么需要自己编写代码进行分段操作,并且可以根据需要采用滑动窗口的形式来分段。这种滑动窗口分段的形式可以增加数据样本量,提升模型效果。

当数据时段之间存在基线差异时,可能会需要从每个时段移除平均基线值,以免影响数据的分析,具体的操作如图 3-50 所示,单击“Tools”→“Remove epoch baseline”。

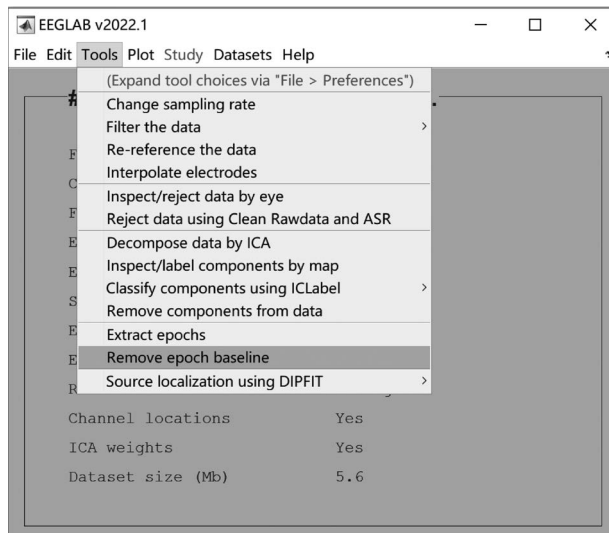


图 3-50 移除基线值操作

在弹窗中选择通道参数,默认为所有通道,单击“Ok”即可,如图 3-51 所示。

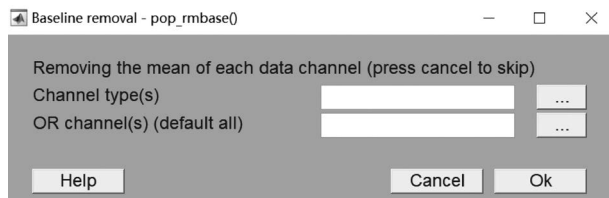


图 3-51 移除基线参数选择

8. 文件保存

预处理完成后,需要保存好新生成的文件。EEGLAB 中默认保存处理好的数据为 .set 文件,如图 3-52 所示。这种类型的文件同时保留了通道数、采样率等信息,可以再次用 EEGLAB 直接打开。

如果后续的处理不再使用 EEGLAB,而是自己编写代码,那么可以在命令行窗口输入“save filename EEG.data”,可以保存当前的数据为 .mat 格式。其中,“filename”是自己指定的文件名。

还可以把前面的处理存成脚本文件,即 .m 文件,以供下一个数据继续使用。具体操作如图 3-53 所示,单击“File”→“History scripts”→“Save dataset history script”。

3.2.4 案例：脑电数据的采集和预处理

1. 抑郁症 EEG 数据集

本章所用的抑郁症 EEG 数据集由北京大学第六医院提供,包含 20 名抑郁症患者(D1~D20)和 20 名健康被试(S1~S20)的静息态脑电数据。抑郁症患者包括 14 名女性和 6 名男性,年龄为 29.50 ± 9.20 岁(均值±方差),健康被试包括 13 名女性和 7 名男性,年龄为

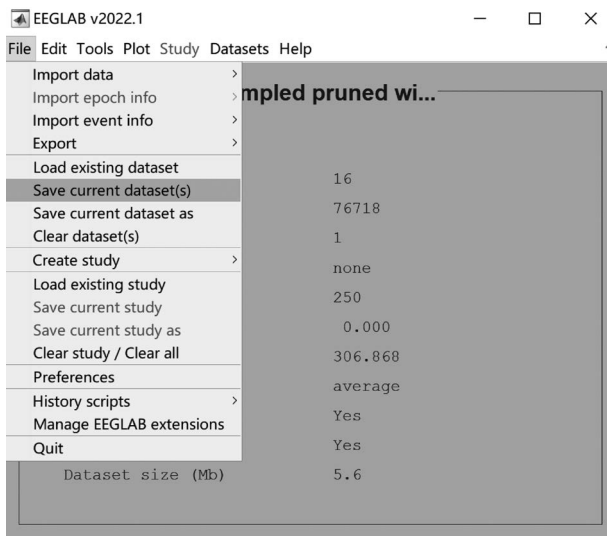


图 3-52 保存当前数据

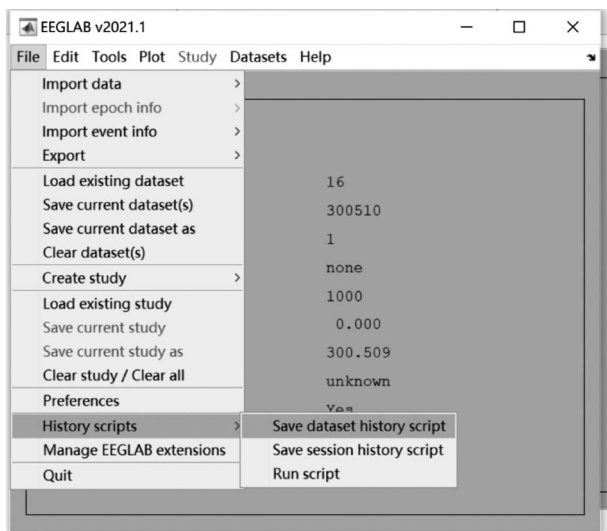


图 3-53 保存预处理操作为脚本文件

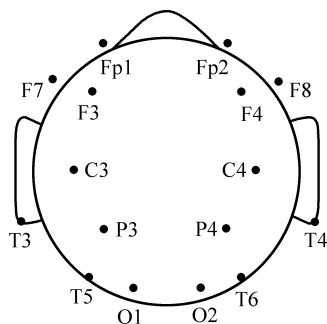


图 3-54 脑电采集设备的电极分布图

29.60 ± 6.48 岁。所有被试均自愿签署知情同意书。

使用 BrainMaster-NM 设备进行 EEG 信号的采集，电极的放置符合国际 10-20 电极放置系统的标准。共记录了每位受试者闭眼静息状态 5 分钟的 EEG 信号，采样频率为 1000Hz，采集的电极包括 Fp1/2、F3/4、C3/4、P3/4、O1/2、F7/8、T3/4 以及 T5/6，这 16 个电极分布在大脑的全部 5 个区域，即额区、中央区、颞区、顶区以及枕区，足以反映被试脑功能的情况，具体分布位置见图 3-54。在整个信号采集的过程中，所有受试者均以舒适的姿态坐在椅子上，并尽量避免周围环境的干扰。

2. 预处理

对给定数据集中 1 个抑郁症患者(数据 D1)和 1 个健康被试(数据 S1)的数据进行预处理。

预处理步骤包括:①将信号的采样频率从 1000Hz 降到 256Hz;②进行 0.5~35Hz 滤波;③手动去伪迹;④ICA 去眼动伪迹;⑤保存为 .mat 文件。

预处理后得到的数据参见配套的数据文件。

3.3 脑电数据的功率谱和时频分析

时域和频域分析是脑电数据分析中最基本的方法。脑电的基本分类就是按照不同频率来定义的,因此本节首先介绍功率谱分析,再介绍时频分析中常用的短时傅里叶变换和小波变换的基本原理和应用。

3.3.1 功率谱分析

脑电信号是一种非平稳的随机信号,一般而言随机信号的持续时间是无限长的,因此随机信号的总能量是无限的,而随机过程的任意一个样本函数都不满足绝对可积条件,所以其傅里叶变换不存在。

不过,尽管随机信号的总能量是无限的,但其平均功率却是有限的,因此,要对随机信号的频域进行分析,应从功率谱出发进行研究才有意义。正因为如此,在研究中经常使用功率谱密度(power spectral density, PSD)来分析脑电信号的频域特性。

1. 基本概念和原理

功率谱密度是对随机变量均方值的量度,是单位频率的平均功率量纲。对功率谱在频域上积分就可以得到信号的平均功率。功率谱密度是一个以频率为自变量的映射,反映了在频率成分上信号有多少功率。

常用功率谱估计方法如图 3-55 所示。

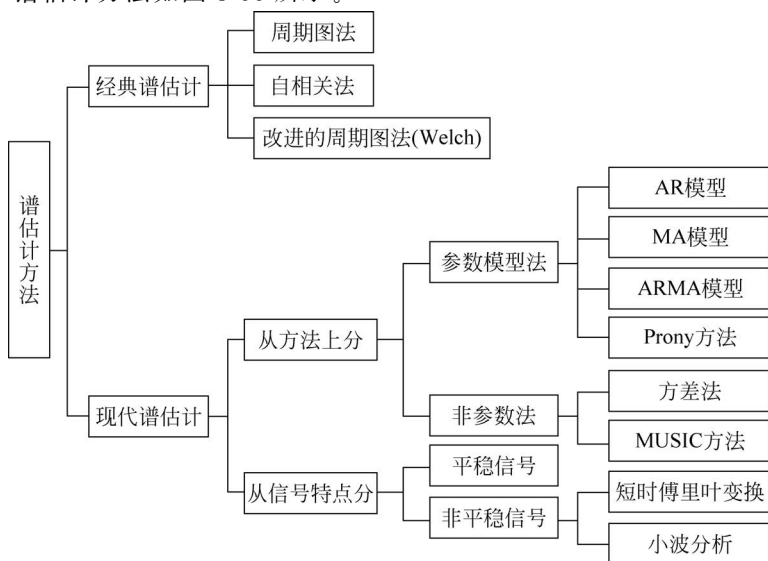


图 3-55 常用功率谱估计方法

经典功率谱估计方法可以分为两种,直接法和间接法。直接法也称为周期图法,它是直接对有限个样本数据进行傅里叶变换来得到功率谱。样本数据越长,直接法获得的分辨率越高。间接法则是先对有限个样本数据进行自相关估计,再进行傅里叶变换,最后得到功率谱。

周期图法是把随机序列 $x(n)$ 的 N 个观测数据视为一个能量有限的序列,直接计算 $x(n)$ 的离散傅里叶变换,得 $X(k)$,然后再取其幅值的平方,并除以 N ,作为序列 $x(n)$ 真实功率谱 $P(x)$ 的估计。即式(3-1)所示。

$$P(x) = \frac{1}{N} |X(k)|^2 \quad (3-1)$$

原始的周期图法误差较大,本章将采用其改进方法,Welch 法。Welch 法是一种修正周期图功率谱密度估计方法,它通过选取的窗口对数据进行加窗处理,先分段求功率谱之后再再进行平均。其中,窗口的长度表示每次处理的分段数据长度,相邻两段数据之间可以有重叠部分。窗口长度越大,得到的功率谱分辨率越高(越准确),但方差加大(即功率谱曲线不太平滑);窗口长度越小,结果的方差会变小,但功率谱分辨率较低(估计结果不太准确)。窗的选择也对结果有一定影响,常见的如汉宁窗、矩形窗、凯撒窗等,一般对脑电信号选择汉宁窗。

2. 功率谱分析的具体操作

EEGLAB 通过调用 MATLAB 信号处理工具箱中的 Welch 方法,来进行功率谱估计。具体操作为:单击“Plot”→“Channel spectra and maps”,可以查看功率谱图,如图 3-56 所示。

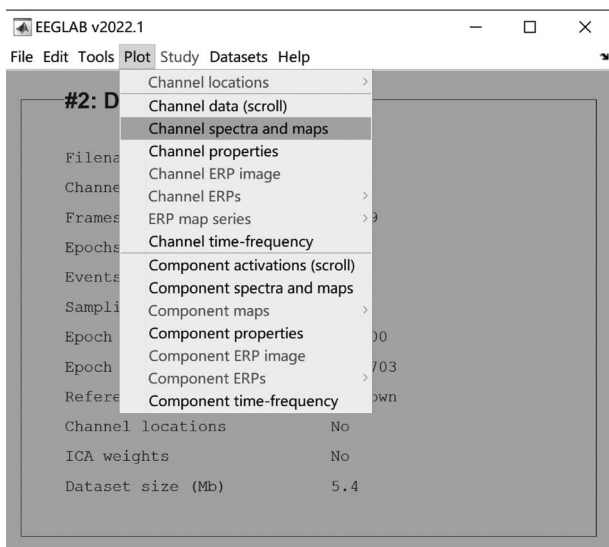


图 3-56 查看功率谱图

弹出窗口如图 3-57 所示,按照需求填写参数,这里为默认参数,单击“Ok”。

单击完成后出现图 3-58 所示界面。

如果要对多个脑电数据进行功率谱分析的批处理,通过代码调用 MATLAB 的 `pwelch` 函数更为方便快捷,还可以根据需要设定参数。该函数的一种格式为

```
[pxx, f] = pwelch(x, window, noverlap, NFFT, fs)
```

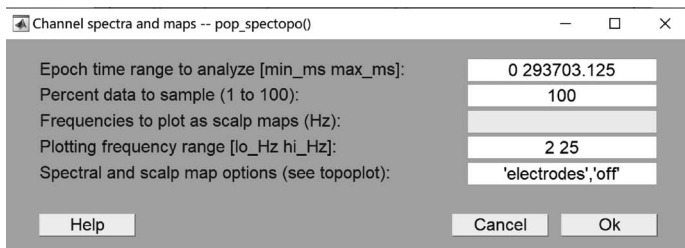


图 3-57 填写功率谱图参数

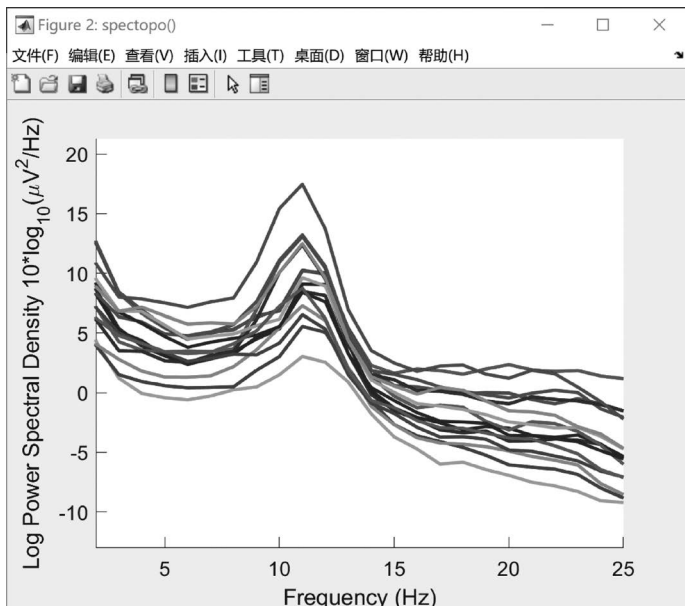


图 3-58 功率谱图示例

其中,函数的输入中 x 是信号数据,当 x 是向量时,它被当作一个单通道信号,当 x 是矩阵时, x 的每一列被当作一个通道的信号。 $window$ 输入为整数时,代表计算功率谱每个窗口的信号长度,选择的窗口越长,越能分辨低频的信号。 $noverlap$ 是每个窗口之间重叠的长度,通常取 33%~50%,窗口之间重叠得越多,图像越平滑,反之则越粗糙。 $NFFT$,即 FFT 数据点的个数,可以变化,但是最大长度不能超过每一段的点数。通常设置 $NFFT$ 为大于每一段的点数的最小 2 次幂,这样可以得到最高的频域分辨率, $NFFT$ 越小,最终分辨率会越粗糙。 fs 是采样频率,最终的结果中,横坐标的最大值为采样频率的一半。函数的输出中, pxx 为计算得到的功率谱数值, f 为功率谱数值对应频率的位置。

3.3.2 短时傅里叶变换

1. 基本概念和原理

对平稳随机信号 $x(t)$,傅里叶变换可得到其频域的信息,它建立了信号从时域到频域的变换桥梁,可以用式(3-2)来表示:

$$X(f) = \mathcal{F}[x(t)] = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad (3-2)$$

式中, f 代表频率; t 代表时间; $e^{-j2\pi ft}$ 为复变函数。

傅里叶变换认为一个周期函数(信号)包含多个频率分量,任意函数(信号) $x(t)$ 可通过多个周期函数(基函数)相加而合成。从物理角度理解傅里叶变换是以一组特殊的函数(三角函数)为正交基,对原函数进行线性变换,物理意义便是原函数在各组基函数的投影。图 3-59 为一平稳信号 $x(t)=0.7\times\sin(2\pi\times 50t)+\sin(2\pi\times 120t)$ 的傅里叶变换,做完快速傅里叶变换(fast Fourier transform,FFT)后,可以在其频谱上看到清晰的两条线(50Hz 和 120Hz),信号包含两个频率成分。而如图 3-60 所示的非平稳信号经 FFT 后,频谱图上就显示出很多其他频率的分量。

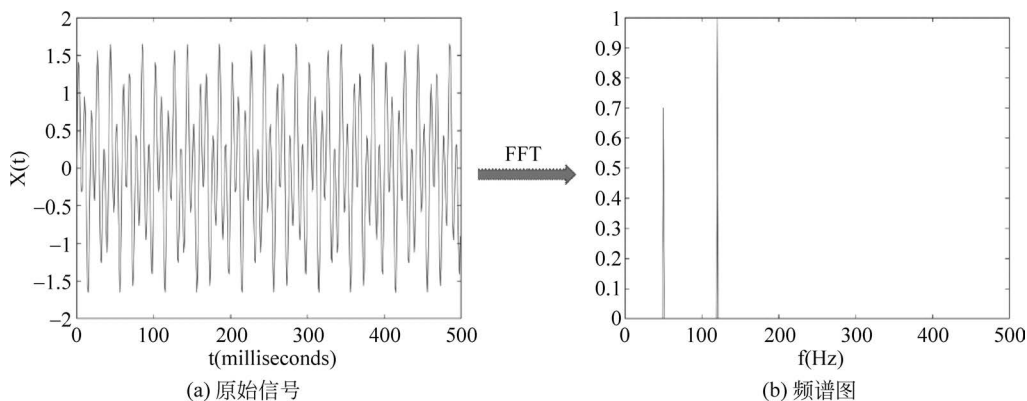


图 3-59 平稳信号的傅里叶变换

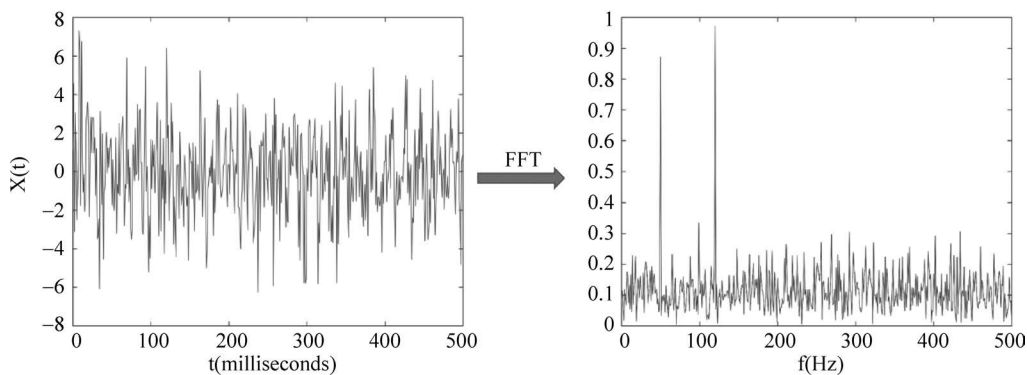


图 3-60 非平稳信号的傅里叶变换

傅里叶变换是一种分析信号的方法,它可分析信号的成分,也可用这些成分合成信号。在分析信号时,主要应用于处理平稳信号,通过傅里叶变换可以得到一段信号总体上包含哪些频率的成分,但是对各成分出现的时刻无法得知。傅里叶变换会忽略信号的时间信息,因此对非平稳信号的处理有天生的缺陷。

为了克服这种局限性,考虑使用局部变换的方法,由此出现了短时傅里叶变换(short-time Fourier transform,STFT)。短时傅里叶变换实质上就是加窗傅里叶变换,它把整个时域过程分解成无数个等长的小过程,假定每个小过程近似的平稳,再进行傅里叶变换,通过窗在时间轴上的移动从而使信号逐段进入被分析状态,最后得到信号的一组“局部”频谱。

给定一个时间宽度很短的窗函数 $\eta(t)$,让窗口滑动,则信号 $z(t)$ 的短时傅里叶变换如

式(3-3)所示:

$$\text{STFT}_z(t, f) = \int_{-\infty}^{\infty} z(t') \eta(t' - t) e^{-j2\pi f t'} dt' \quad (3-3)$$

窗函数 $\eta(t)$ 的存在,使得短时傅里叶变换具有了局域特性,既是时间的函数,也是频率的函数。特别地,当窗函数 $\eta(t) \equiv 1, \forall t$ 时,短时傅里叶变换退化为传统傅里叶变换。

选定窗函数之后,这个时频窗与时间 t 和频率 f 无关,考虑傅里叶变换的理论极限,即不确定性原理,当窗口越小,就越能知道信号中某个频率出现在哪里,但是对自身的频率值了解越少;窗口越大,对频率值的了解就越多,而对时间信息了解就越少。

2. 短时傅里叶变换的具体操作

通过编写代码,调用 MATLAB 的信号处理函数 spectrogram 即可, spectrogram 函数的功能是利用短时傅里叶变换求信号的功率谱,格式为

```
[s, f, t] = spectrogram(x, window, noverlap, f, fs)
```

其中,输入参数: x 为待分析的信号; window 为窗函数,默认值为汉宁窗; noverlap 为窗口重叠的长度,默认值为 50%; f 为指定计算的频率点; fs 为采样的频率。输出参数: s 为函数返回的短时傅里叶变换后的值; f 为对应的频率点; t 为时间序列。

取一段 EEG 数据(抑郁症患者 D1 的第一个电极的 10 秒数据),调用 spectrogram 函数,执行如下代码:

```
D1_top10s = D1(1,1:2561)
[s, f, t] = spectrogram(D1_top10s, 256)
```

可得到如图 3-61 所示的结果。图 3-61(a)为原始数据波形,图 3-61(b)为短时傅里叶变换得到的功率谱图。

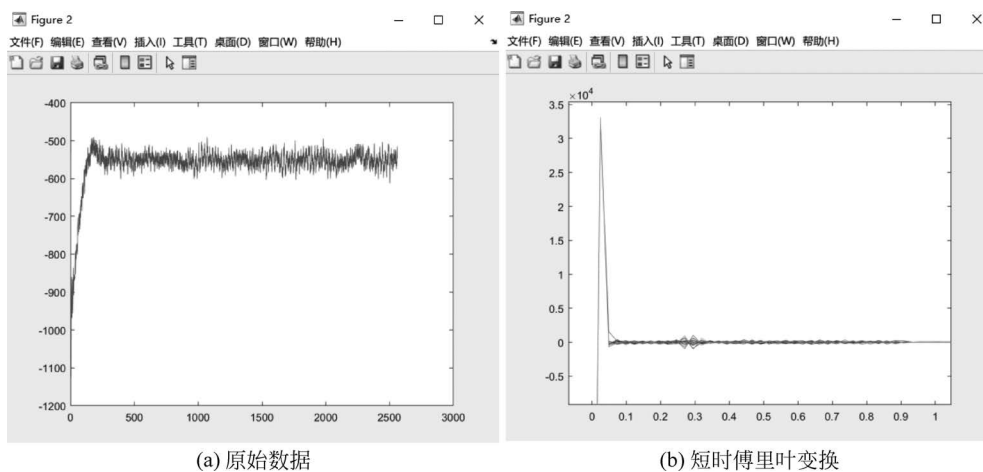


图 3-61 EEG 数据短时傅里叶变换示例

3.3.3 小波分析

1. 基本概念和原理

小波变换是 20 世纪 80 年代后期出现的一个应用数学的分支,后被学者引入到信号处理领域。小波变换在频域和时域上都有很好的表现。小波变换具有多分辨特性,也叫多尺度特性,可以由粗到精地逐步观察信号,也可以看成是用一组带通滤波器对信号滤波。这种

特性是通过不同的伸缩变换实现的。首先,看一个大尺度/窗口的信号,并分析“大”特征,然后看一个小尺度的信号,以便分析更小的特征。和短时傅里叶变换相比,小波变换的窗口大小是可变的。图 3-62 显示了不同方法的时间分辨率和频率分辨率。

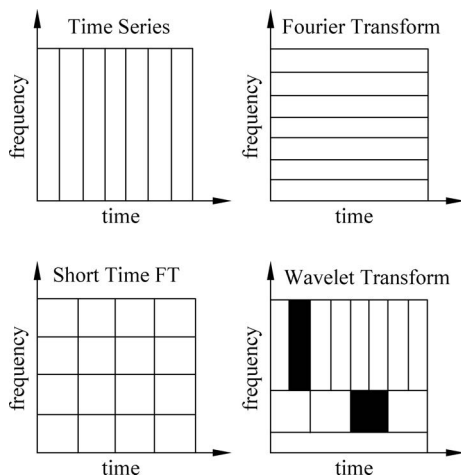


图 3-62 各种时频分析方法的分辨率

傅里叶变换是通过一系列频率不同的正弦波来拟合信号。也就是说,信号是通过正弦波的线性组合来表示的。小波变换使用一系列称为小波的函数,每个函数具有不同的尺度。正弦波和小波分别是傅里叶变换和小波变换的基。正弦波是无限延长的,小波则是在一个时间点上的局域波。图 3-63 为正弦波和小波的示意图。



图 3-63 正弦波和小波

将信号 $z(t)$ 在小波基下展开,称为 $z(t)$ 的连续小波变换 (continuous wavelet transform, CWT),其表达式如式(3-4)所示:

$$\text{CWT}_z(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} z(t) \varphi\left(\frac{t-\tau}{a}\right) dt \quad (3-4)$$

从式(3-4)可以看出,不同于傅里叶变换的变量只有频率 f ,小波变换有两个变量:尺度 a (scale) 和平移量 τ (translation)。 a 控制小波函数的伸缩, τ 控制小波函数的平移。 a 对应于频率(反比), τ 对应于时间。

根据时频分析的要求,小波基函数 $\varphi(t)$ 应满足以下条件:

- (1) 只有小的局部非零区域,在窗口之外函数为零。
- (2) 本身是振荡的,具有波的性质,并且没有直流趋势成分,即满足式(3-5)。

$$\Psi(0) = \int_{-\infty}^{+\infty} \varphi(t) dt = 0 \quad (3-5)$$

式中, $\Psi(f)$ 是函数 $\varphi(t)$ 的傅里叶变换。

- (3) 包含尺度参数 a ($a > 0$) 和平移参数 τ 。

常用的连续小波基函数有 Morlet 小波和 Marr 小波等。

小波基函数不断和信号相乘。某一个尺度(宽窄)下相乘的结果,就可以理解成信号所包含的当前尺度对应频率成分有多少。于是,基函数会在某些尺度下,与信号相乘得到一个很大的值,因为此时二者有一种重合关系。那么就知道信号包含该频率的成分为多少。与傅里叶变换不同的是,小波变换不但可以知道信号有这样频率的成分,而且知道它在时域上存在的具体位置。当在每个尺度下都不断平移并和信号相乘后,就可以得到信号在每个位置都包含哪些频率成分的信息。

2. 小波分析的具体操作

MATLAB 有小波工具箱,提供了丰富的函数以供选用,还提供了图形用户界面,可以方便地进行小波分析的操作。如图 3-64 所示为小波分析的工具包。图 3-65 所示为小波分析工具包界面。



图 3-64 小波分析工具包

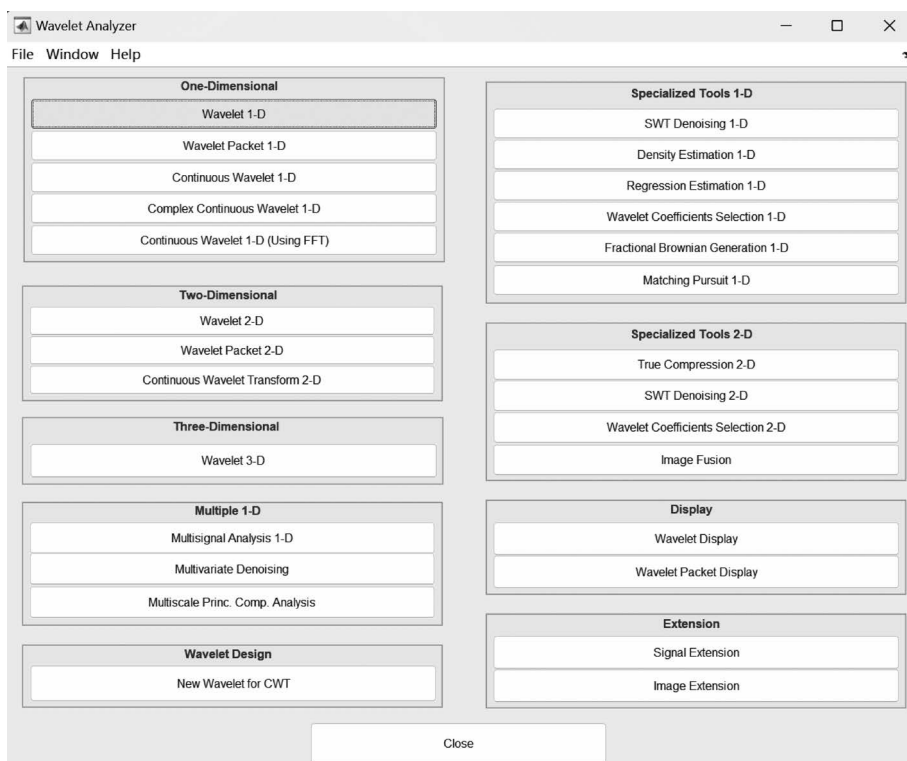


图 3-65 小波分析工具包界面

根据需要设定参数,即可进行分析。如果想进行数据的批处理,也可以直接编写代码进行分析。

3.3.4 案例:脑电数据功率谱和时频分析

在 3.2.4 节中对数据 D1 和 S1 进行了预处理,此处继续进行分析,分别进行功率谱、短时傅里叶变换和小波分析。

(1) 功率谱分析:对抑郁症患者 EEG(数据 D1)经过预处理后的数据进行功率谱分析,可得图 3-66。

对健康被试 EEG(数据 S1)的数据进行功率谱分析,结果如图 3-67 所示。

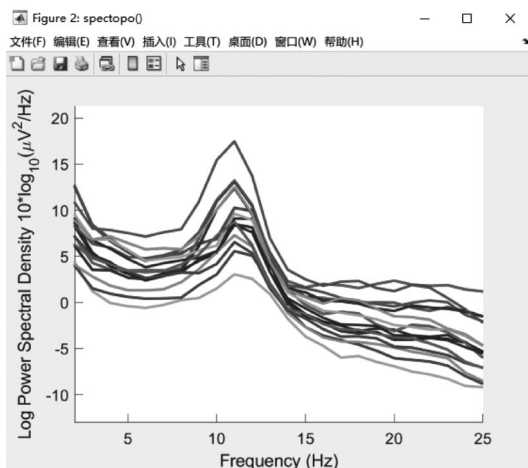


图 3-66 抑郁症患者 EEG 功率谱分析

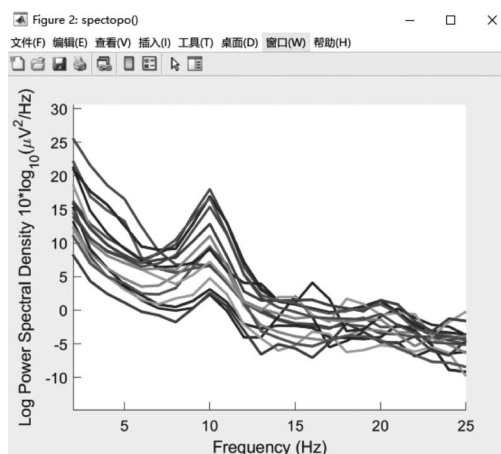


图 3-67 健康被试 EEG 功率谱分析

(2) 对数据 D1 和 S1 进行短时傅里叶变换,代码如下:

```
% B 是 F 大小行 T 大小列的频率峰值,P 是对应的能量谱密度
[B, F, T, P] = spectrogram(D1_top10s, 100, 99, 0.5:30, 256);;
% spectrogram 函数返回输入信号的短时傅里叶变换,第一个 100 为窗函数大小,99 为重叠的采样
点数
% 第二个 100 为计算离散傅里叶变换的点数,Fs = 256 为采样频率
figure
imagesc(T, F, P);
% imagesc(T, F, B)
% 设置 y 轴数值为正常显示
set(gca, 'YDir', 'normal')
ylim([0, 35]); % y 轴范围
colorbar; % 色标
xlabel('时间 t/s');
ylabel('频率 f/Hz');
title('短时傅里叶时频图');
```

此处对 D1 和 S1 的 Fp1 通道的前 10s 数据进行短时傅里叶变换,结果见图 3-68 和图 3-69。

(3) 小波变换:对 EEG 数据 D1 和 S1 进行小波分析,结果如图 3-70 和图 3-71 所示。

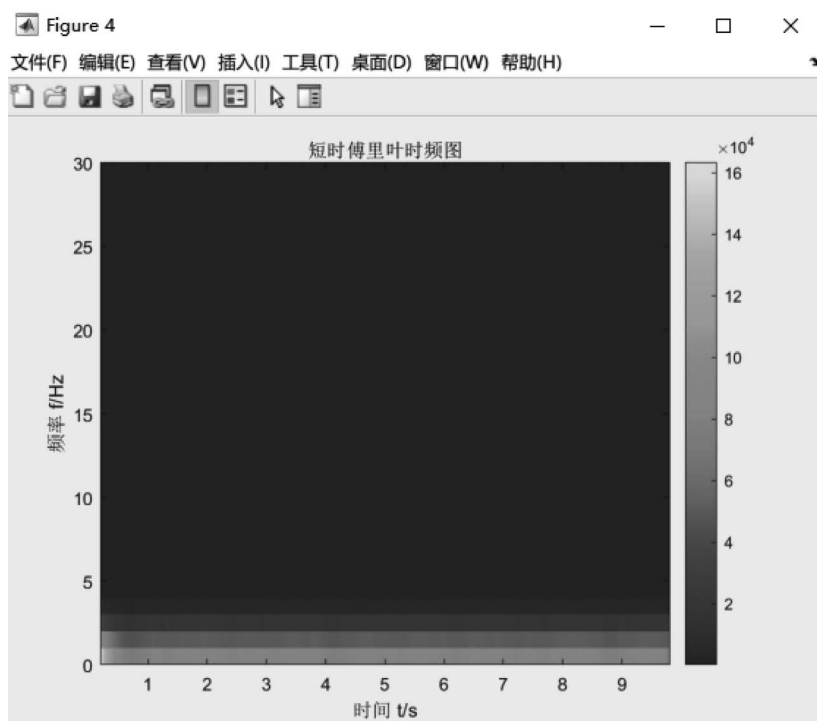


图 3-68 抑郁症患者 EEG 短时傅里叶变换结果

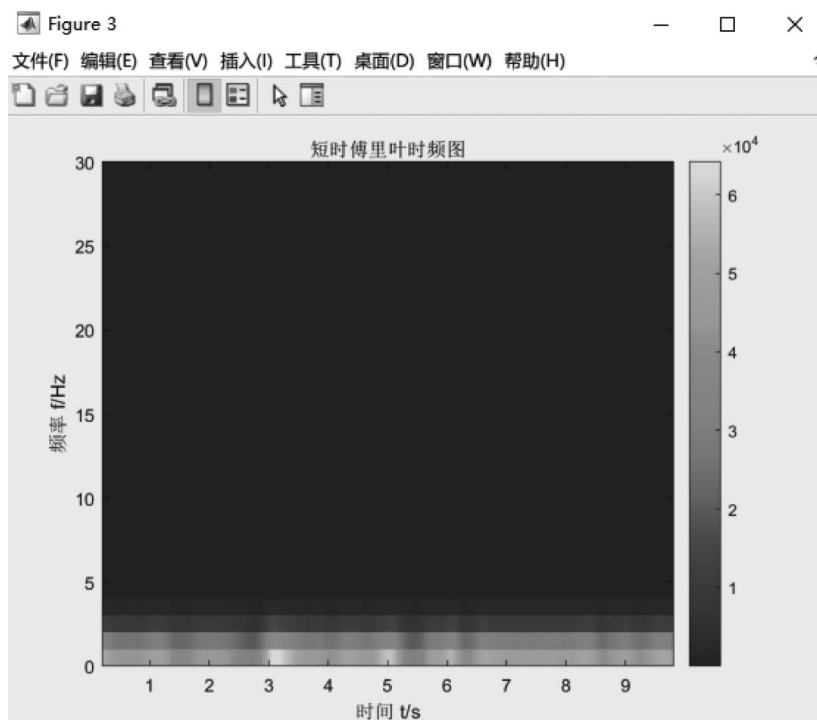


图 3-69 健康被试 EEG 短时傅里叶变换结果

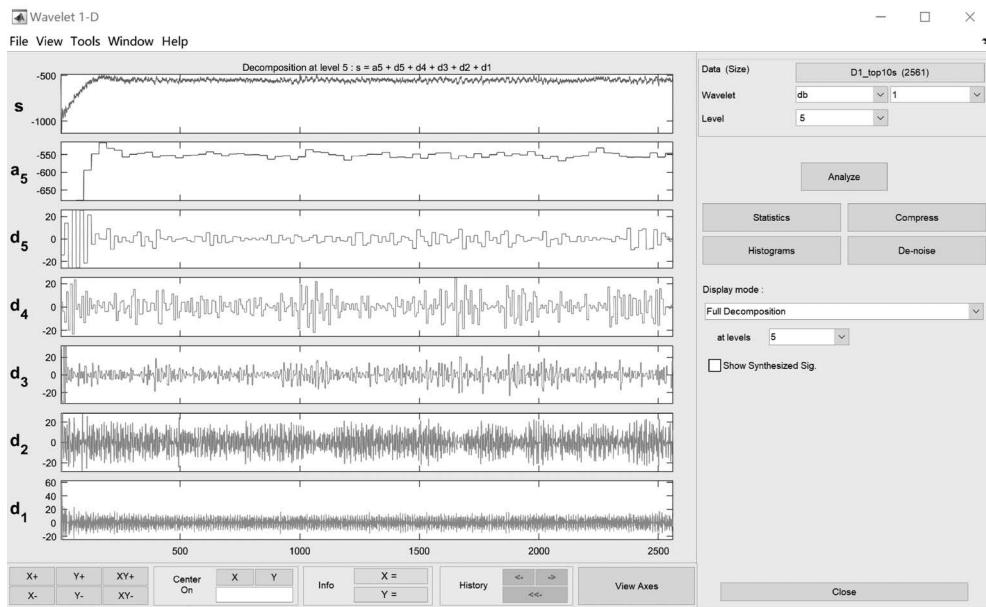


图 3-70 抑郁症患者 EEG 小波分析结果

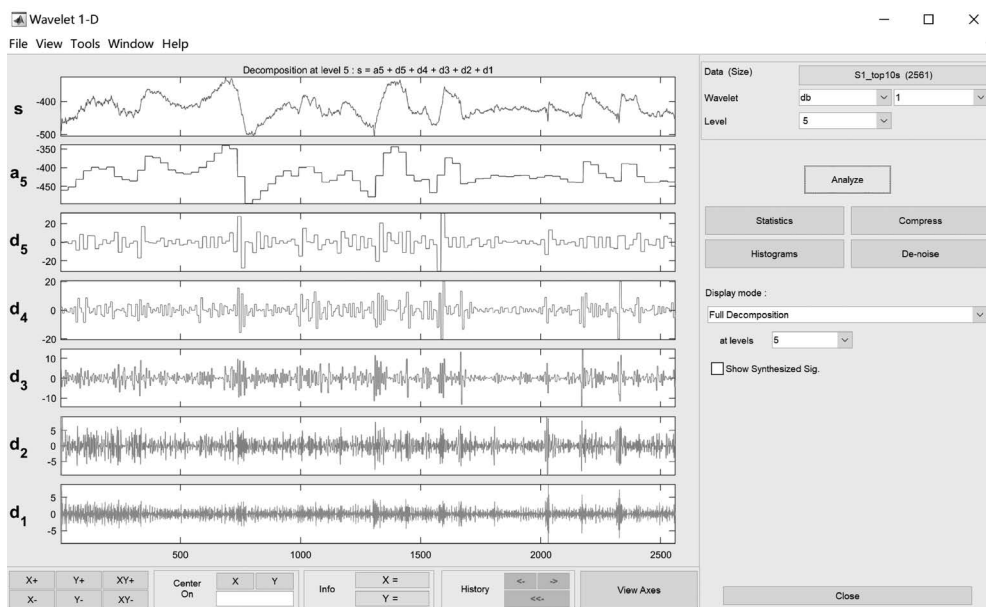


图 3-71 健康被试 EEG 小波分析结果

3.4 脑电数据的非线性分析

大脑是一个非线性的系统, EEG 实质上是一种非线性信号, 因此非线性分析也是对脑电数据进行分析的重要方向。本节将介绍 Lemple-Ziv 复杂度 (Lemple-Ziv Complexity, LZC)、小波熵以及分形维数这些非线性描述量在脑电数据分析中的应用。


```

end
% 循环得到的 c 是字符串断点的数目, 所以要加 1
c = c + 1;
b = length(x)/log2(length(x));
lzc = c/b;
fprintf('\n\n 序列 data 的 LZC 复杂度是\n\n');
fprintf('%f', lzc);
return;

```

相关研究的一些结果如下:

(1) Li 等对抑郁症患者、精神分裂症患者和正常人闭眼休息以及精神活跃状态下进行心算连减的脑电信号 LZC 进行分析, 发现抑郁症患者和精神分裂症患者的 LZC 均高于正常人, 且存在着显著性的差异。

(2) Méndez 等统计了 20 名未用药的重度抑郁症患者和 19 名健康对照 5 个脑区(148 个通道)的 LZC, 发现所有脑区患者的 LZC 均高于对照组, 且经过 6 个月的抗抑郁药物治疗后, 患者的 LZC 有所降低。

(3) Akar 等对 15 名重度抑郁症患者和 15 名健康对照被试静息态、听噪音(消极情绪内容)以及听音乐(积极情绪内容)的脑电信号进行非线性动力学分析, 发现在任何状态下, 患者的 LZC 均高于健康被试, 且在额顶头皮部位获得显著差异, 在颞区和中央区, 重度抑郁症患者的脑电图与对照组的脑电图之间没有发现明显的复杂性差异。

3.4.2 小波熵

小波熵, 可以区分自身或刺激下特定的脑状态是从小波分解后的信号序列中计算的一种熵值, 反映了信号谱能量在各个子空间分布的有序或无序程度。小波熵越大, 表明信号的能量分布越分散, 信号本身越无序。具体计算方式如下:

(1) 首先计算每一段信号尺度 j 的小波能量 E_j :

$$E_j^{(N_j)} = \sum_{k=m_j}^{m_j+N_j-1} d_j(k)^2 \quad (3-6)$$

式中, m_j 表示尺度 j 的第 m 个数据段; d 表示尺度 j 的小波系数; N 表示运行窗口的长度。

(2) 计算总能量 E_{tot} :

$$E_{\text{tot}} = \sum_j E_j^{(N_j)} = \sum_j \sum_{k=m_j}^{m_j+N_j-1} d_j(k)^2 \quad (3-7)$$

(3) 将小波能量除以总能量, 得到各尺度 j 的相对小波能量和运行窗口长度:

$$p_j^{(N_j)} = \frac{E_j^{(N_j)}}{E_{\text{tot}}} = \frac{\sum_{k=m_j}^{m_j+N_j-1} d_j(k)^2}{\sum_j \sum_{k=m_j}^{m_j+N_j-1} d_j(k)^2} \quad (3-8)$$

(4) WE 即为不同尺度间 $p_j^{(N_j)}$ 分布的熵:

$$\text{WE} = - \sum_j p_j^{(N_j)} \log p_j^{(N_j)} \quad (3-9)$$

代码如下:

```
function wentropy = wavelet_entropy_func(x,Fs)

N_ceng = round(log2(Fs)) - 3;
wentropy = 0;
E = waveletdecom_cwq(x, N_ceng, 'db4');
P = E/sum(E);
P = P(find(P~=0));
for j = 1:size(P, 2)
wentropy = wentropy - P(1, j). * log(P(1, j));           % 小波熵 Swt = - sum(Pj * logPj)
end

end

function [ E ] = waveletdecom_cwq(x, n, wpname)
[C,L] = wavedec(x,n,wpname);           % 对数据进行小波包分解
for k = 1:n
% wpcoef(wpt1,[n,i-1])是求第 n 层第 i 个节点的系数
% disp('每个节点的能量 E(i)');
SRC(k, :) = wrcoef('a', C, L, 'db4', k);           % 尺度
SRD(k, :) = wrcoef('d', C, L, 'db4', k);           % 细节系数
% 求第 i 个节点的范数平方,其实也就是平方和
E(1, k) = norm(SRD(k, :)) * norm(SRD(k, :));
end
E(1, n+1) = norm(SRC(n, :)) * norm(SRC(n, :));
% disp('小波包分解总能量 E_total');
E_total = sum( E );           % 求总能量
y = E_total ;
end
```

相关研究的一些结果:

(1) 张胜等对安静闭目状态下的抑郁症患者和正常人的 16 导自发脑电小波熵进行分析,发现抑郁症患者组有 13 个导联的小波熵值大于正常人组,表明该状态下抑郁症患者的信号更为分散,成分更多,大脑活动更为无序。

(2) 盖淑萍等提出了一种基于小波包分解节点重构信号的功率谱熵值的脑电信号分析方法,并针对抑郁症患者和正常健康人的静息态脑电信号进行计算和分析,发现抑郁症患者脑电信号的熵值在部分脑区显著大于正常健康人。

3.4.3 分形维数

分形维数定量地描述了研究对象的复杂程度,在复杂系统和复杂信号分析中得到了广泛应用。具体计算方式描述如下:

将尺寸分别为 $r=1/4$ 和 $1/8$ 的网格覆盖在分形曲线上,计算网格中包含有图像像素的方格数目,不断减小网格尺寸 r ,继续计算包含图像像素的网格数 $N(r)$,直至最小的网格尺寸达到像素为止。绘制出 $\ln(N(r)) \sim \ln(1/r)$ 图像,进行直线拟合,得到一条直线,那么直线的斜率 FD 即为该曲线的分形维数。

代码如下:

```
function D = FractalDim(y, cellmax)
% 求输入一维信号的计盒分形维数
% y 是一维信号
% cellmax: 方格子的最大边长,可以取 2 的偶数次幂次(1,2,4,8,...),取大于数据长度的偶数
```

```

% D 是 y 的计盒维数(一般情况下  $D > 1$ ),  $D = \lim(\log(N(e))/\log(k/e))$ ,
if cellmax < length(y)
error('cellmax must be larger than input signal!')
end
L = length(y); % 输入样点的个数
y_min = min(y);

% 移位操作, 将 y_min 移到坐标 0 点
y_shift = y - y_min;
% 重采样, 使总点数等于 cellmax + 1
x_ord = [0:L-1]./(L-1);
xx_ord = [0:cellmax]./(cellmax);
y_interp = interp1(x_ord, y_shift, xx_ord);
% 按比例缩放 y, 使最大值为  $2^c$ 
ys_max = max(y_interp);
factory = cellmax/ys_max;
yy = abs(y_interp * factory);

t = log2(cellmax) + 1; % 迭代次数
for e = 1:t
Ne = 0; % 累积覆盖信号的格子的总数
cellsize = 2^(e-1); % 每次的格子大小
NumSeg(e) = cellmax/cellsize; % 横轴划分成的段数

for j = 1:NumSeg(e); % 由横轴第一个段起通过计算纵轴跨越的格子数累积 N(e)
begin = cellsize * (j-1) + 1; % 每一段的起始
tail = cellsize * j + 1;
seg = [begin: tail]; % 段坐标
yy_max = max(yy(seg));
yy_min = min(yy(seg));
up = ceil(yy_max/cellsize);
down = floor(yy_min/cellsize);
Ns = up - down; % 本段曲线占有的格子数
Ne = Ne + Ns; % 累加每一段覆盖曲线的格子数
end
N(e) = Ne;
end
% 对  $\log(N(e))$  和  $\log(k/e)$  进行最小二乘的一次曲线拟合, 斜率就是 D
r = -diff(log2(N)); % 去掉 r 超过 2 和小于 1 的野点数据
id = find(r <= 2 & r >= 1); % 保留的数据点
Ne = N(id);
e = NumSeg(id);

% plot(log(e), log(Ne), '--b*');
P = polyfit(log(e), log(Ne), 1); % 一次曲线拟合返回斜率和截距
D = P(1);

```

相关研究的一些结果:

(1) Akar 等对 15 个重度抑郁症患者和 15 个健康被试脑电信号的 Katz 分形维数(Katz fractal dimension, KFD)进行了分析, 发现重度抑郁症患者的该指数均高于健康被试。

(2) Ahmadlou 等分析了抑郁症患者额区脑电信号不同节律的分形维数变化规律, 发现抑郁症患者 beta 频段左额区, 右额区, 整个额区的分形维数均显著大于正常人。

3.4.4 案例: 脑电数据非线性分析

在 3.2.4 节中, 对数据 D1 和 S1 进行了预处理, 此处将对这两个数据进行几种非线性分

析: LZC、小波熵和分形维数。将预处理后的原始数据以 10s 为一段进行分段,每位被试共有 29 段,计算每一段的 LZC、小波熵以及分形维数,并求平均值作为该被试脑电数据的最终结果。

三种方法的计算结果分别如图 3-73、图 3-74 和图 3-75 所示。

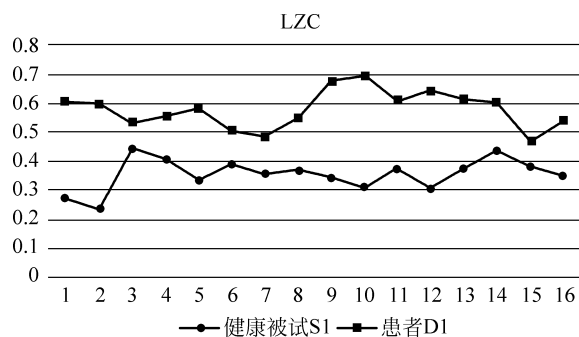


图 3-73 LZC 结果

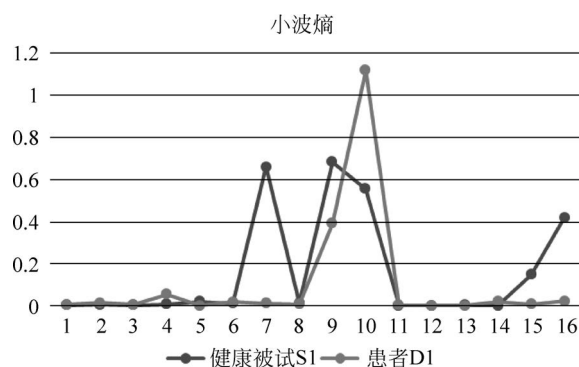


图 3-74 小波熵结果

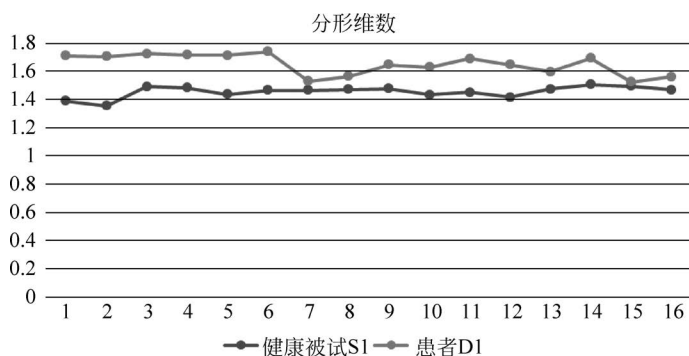


图 3-75 分形维数结果

3.5 脑电数据的网络分析

人类大脑是自然界中结构最为复杂、功能最为高效的器官之一,被认为是一个复杂的系统,其精巧和完善的结构以及功能连接模式使得大脑具有强大的信息分化与整合功能。

目前,有很多分析方法可以用来刻画脑网络,其中基于图论的分析方法可以对大脑功能

网络和结构网络进行描绘和评估。作为科学计算的一个分支,图论通过定义脑网络的节点和边,对脑网络拓扑结构进行量化,从而帮助我们更深入地理解大脑的网络结构。

本节将从脑网络的构建、参数的计算与分析进行介绍。

3.5.1 脑网络的构建

脑网络的构建需要根据关注的问题来决定选择何种参数进行网络构建,常用的参数有:相关系数,衡量不同电极信号间的相关程度;相位滞后指数,衡量不同电极信号间的相位关系等。一个脑网络的构建可以分为以下四个步骤。

(1) 载入预处理后的数据。

设待分析的数据为 $\mathbf{X}$, $\mathbf{X}$ 为一个 $n \times \text{time}$ 的矩阵,其中, n 为电极数目,本章 EEG 的 n 均为 16; time 是时间点。可用 load 函数载入预处理后的 .mat 数据,同时将数据分成 10s 一小段进行计算,每段计算完毕后再进行平均以得到最后结果。

(2) 计算相位滞后指数。

相位滞后指数(phase lag index, PLI),代码如下:

```
% PhaseLagIndex.m
function PLI = PhaseLagIndex(X)
% Given a multivariate data, returns phase lag index matrix
ch = size(X,2); % column should be channel
% % % % % Hilbert transform and computation of phases
phil = angle(hilbert(X));
PLI = ones(ch,ch);
for ch1 = 1:ch-1
    for ch2 = ch1 + 1:ch
        % phase lage index
        PDiff = phil(:,ch1) - phil(:,ch2); % phase difference
        PLI(ch1,ch2) = abs(mean(sign(sin(PDiff)))); % only count the asymmetry
        PLI(ch2,ch1) = PLI(ch1,ch2);
    end
end
for i = 1:16
    PLI(i,i) = 0;
end
```

运行结果如图 3-76 所示,得到一个 16×16 的连接矩阵。

(3) 构建二值矩阵。

得到连接矩阵后,选取合适的阈值,将其二值化。这个二值矩阵就是所构建的脑网络。

代码如下:

```
function MIb = get_binary(M)
th = 0.03
[m,n] = size(M);
for i = 1:m
    for j = 1:n
        if(M(i,j) < th)
            MIb(i,j) = 0;
        else
            MIb(i,j) = 1;
        end
    end
end
end
```

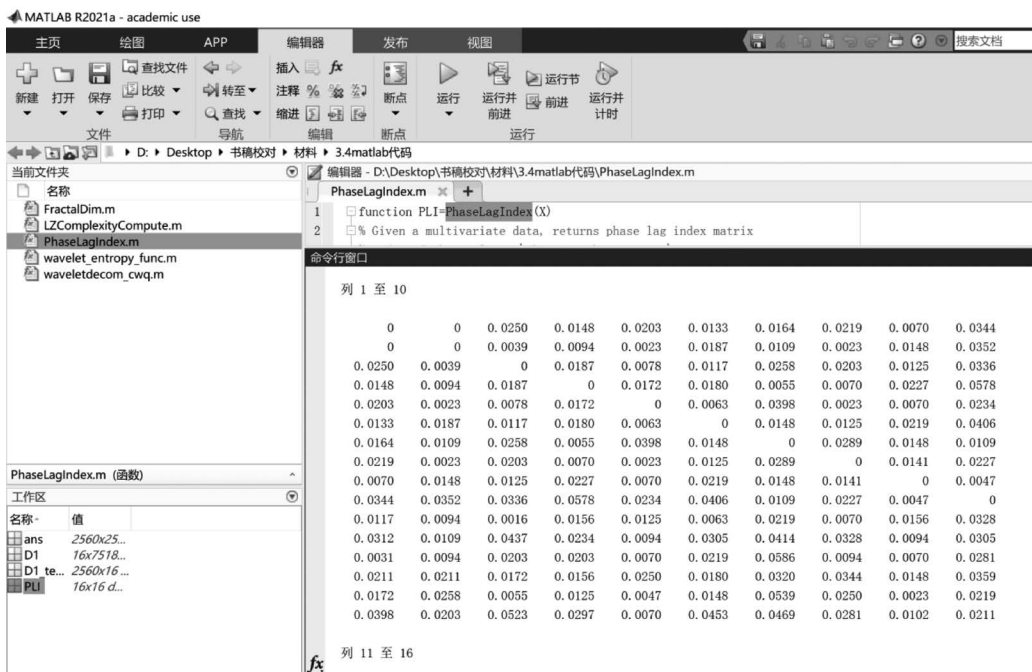


图 3-76 连接矩阵的获取

(4) 画出网络图。

用 `fieegnet` 函数可以画出网络图。

当阈值取 0.03 时,网络图如图 3-77 所示。

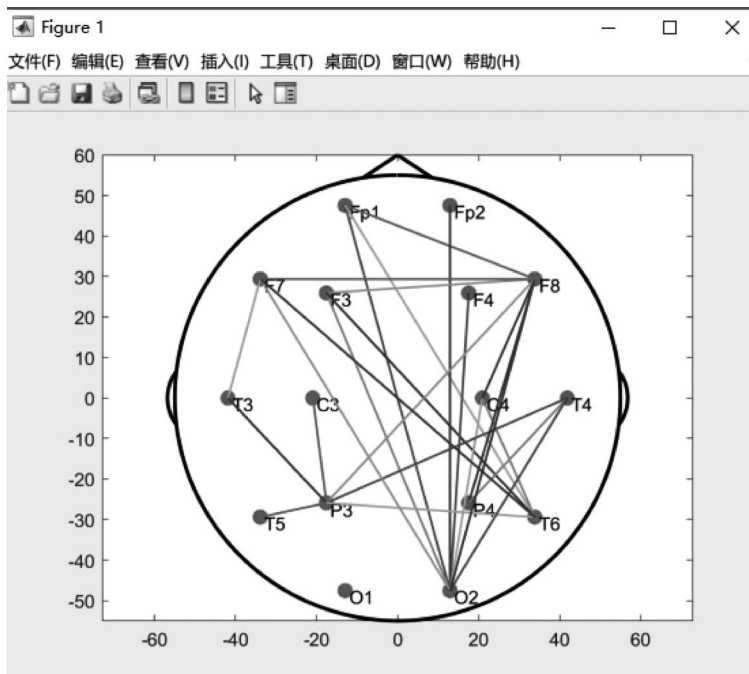


图 3-77 得到脑网络图

3.5.2 网络参数的计算与分析

1. 调用相应函数计算节点度数、聚类系数、特征路径长度和全局效率四个参数。

```
function [degree, ccb, cp, eb] = getParam(M)
degree = degrees_und(M);      % 节点度数
ccb = clustering_coef_bu(M);   % 聚类系数
cp = charpath(M);             % 特征路径长度
eb = efficiency_bin(M);       % 全局效率
```

(1) 节点度数：节点度数是用来衡量一个图中一个节点与其他节点之间连接性的属性指标,定义为图 G 中一个节点的相应边数,是节点的出度与其入度的和。

代码如下：

```
function [deg] = degrees_und(CIJ)
% Input: CIJ, undirected (binary/weighted) connection matrix
% Output: deg, node degree
CIJ = double(CIJ ~ 0);
Deg = sum(CIJ);
```

(2) 聚类系数：聚类系数是用来表示图中节点的聚集程度,规则网络与小世界网络具有较高的聚类系数,而随机网络的聚集系数则较低,计算方式如下：

$$C = n / C_k^2 = 2n / (k(k-1)) \quad (3-10)$$

式中, C 表示的是该节点的聚类系数; k 表示该节点的全部邻接点数目。

代码如下。

```
function C = clustering_coef_bu(G)
% Input: A, binary undirected connection matrix
%
% Output: C, clustering coefficient vector
n = length(G);
C = zeros(n, 1);

for u = 1:n
    V = find(G(u, :));
    k = length(V);
    if k >= 2 % degree must be at least 2
        S = G(V, V);
        C(u) = sum(S(:)) / (k^2 - k);
    end
end
```

(3) 特征路径长度：在网络中,任选两个节点,连通这两个节点的最少边数定义为这两个节点的路径长度,网络中所有节点对路径长度的平均值,定义为网络的特征路径长度。

代码如下：

```
function [lambda, efficiency, ecc, radius, diameter] = charpath(D, diagonal_dist, infinite_dist)
% Input:      D, distance matrix
%            diagonal_dist optional argument
%            include distances on the main diagonal
%            (default: diagonal_dist = 0)
%            infinite_dist optional argument
%            include infinite distances in calculation
%            (default: infinite_dist = 1)
%
% Output:     lambda network characteristic path length
```

```

%           efficiency      network global efficiency
%           ecc             nodal eccentricity
%           adius           network radius
%           diameter        network diameter
n = size(D, 1);
if any(any(isnan(D)))
    error('The distance matrix must not contain NaN values');
end
if ~exist('diagonal_dist', 'var') || ~diagonal_dist || isempty(diagonal_dist)
    D(1:n+1:end) = NaN; % set diagonal distance to NaN
end
if exist('infinite_dist', 'var') && ~infinite_dist
    D(isinf(D)) = NaN; % ignore infinite path lengths
end
Dv = D(~isnan(D)); % get non - NaN indices of D
% Mean of entries of D(G)
lambda = mean(Dv);
% Efficiency: mean of inverse entries of D(G)
efficiency = mean(1./Dv);
% Eccentricity for each vertex
ecc = nanmax(D, [], 2);
% Radius of graph
Radius = min(ecc);
% Diameter of graph
Diameter = max(ecc);

```

(4) 全局效率：全局效率用于衡量网络中并行信息传输的全局效率。最短路径长度越短，网络全局效率越高，则网络节点间传递信息的速率就越快。计算方式如下：

$$E_{\text{glob}} = \frac{1}{n(n-1)} \sum_{j \in V, j \neq i} \frac{1}{l_{ij}} \quad (3-11)$$

式中， l_{ij} 表示两个节点间的最短路径。

代码如下：

```

function E = efficiency_bin(A, local)

% Inputs:  A,          binary undirected or directed connection matrix
%          local       optional argument
%                  local = 0 computes global efficiency (default)
%                  local = 1 computes local efficiency
%
% Output:  Eglob       global efficiency (scalar)
%          Eloc        local efficiency (vector)

n = length(A); % number of nodes
A(1:n+1:end) = 0; % clear diagonal
A = double(A ~ 0); % enforce double precision

if exist('local', 'var') && local % local efficiency
    E = zeros(n, 1);
    for u = 1:n
        V = find(A(u, :) | A(:, u).'); % neighbors
        sa = A(u, V) + A(V, u).'; % symmetrized adjacency vector
        e = distance_inv(A(V, V)); % inverse distance matrix
        se = e + e.'; % symmetrized inverse distance matrix
        numer = sum(sum((sa.' * sa) .* se))/2; % numerator
        if numer ~ 0
            denom = sum(sa).^2 - sum(sa.^2); % denominator

```

```

            E(u) = numer / denom;           % local efficiency
        end
    end
else                                           % global efficiency
    e = distance_inv(A);
    E = sum(e(:))./(n^2 - n);
end

function D = distance_inv(A_)
l = 1;                                       % path length
Lpath = A_;                                % matrix of paths l
D = A_;                                     % distance matrix
n_ = length(A_);

Idx = true;
while any(Idc(:))
    l = l + 1;
    Lpath = Lpath * A_;
    Idc = (Lpath ~ 0) & (D == 0);
    D(Idc) = 1;
end

D(~D | eye(n_)) = inf;                     % assign inf to disconnected nodes and to diagonal
D = 1./D;                                   % invert distance

```

设定阈值范围,进行遍历,对这些参数再次进行计算,保存各阈值对应的结果。

输出结果。对每个人的数据都分段进行上述处理,然后求得每个人的平均结果。

3.5.3 案例:疼痛脑电数据分析

1. 疼痛 EEG 数据集

该疼痛 EEG 数据集由电子科技大学提供,被试 3 名,均为医学院学生,年龄在 18 到 28 岁之间,右利手,整个数据采集过程在安静的室内空间完成。实验开始时,要求被试舒服地坐在椅子上,指示被试保持放松无杂念的状态,主要分以下三步:

(1) 令被试调整状态至平静,记录安静状态下闭眼和睁眼的脑电各 5 分钟。

(2) 被试稍作休息后,要求其测试手的掌心向上,半开放,实验人员将冰块放入其手掌,刺激持续时间以不引起伤害为前提。当被试表示无法继续忍受疼痛时,刺激结束,并进行计时。进行冷压刺激的同时记录脑电(120 秒)。

(3) 每段刺激完毕后要求被试填写 McGill 疼痛问卷量表,量表以纸张打印的形式提供。

McGill 疼痛问卷量表是国际公认的描述和测定疼痛的量表,主要包括:①目测类比定级法(visual analogue scale,VAS):定量描述、测量评价疼痛程度;②疼痛分级指数(pain rating index,PRI):描述疼痛的性质;③现有疼痛强度(present pain intensity scale,PPI):可定性描述疼痛程度。

采集的电极包括 Fp1/2、F3/4、C3/4、P3/4、O1/2、F7/8、T3/4 以及 T5/6,采样频率为 250Hz。

2. 数据处理步骤

(1) 按照 3.2.4 节的步骤对数据进行预处理。

(2) 为每位被试选择连续的 3 个 30s 数据段,分别计算其 PLI 连接矩阵并求平均值,作为该被试的 PLI 连接矩阵。

(3) 从 0.1~0.3,以 0.05 为步长,设置 5 个阈值,分别求其二值矩阵并计算相应的指标。

3. 数据计算结果

按照上述计算步骤,可以得到 3 名被试的脑网络分析结果,如表 3-1~表 3-6 所示。

表 3-1 被试 1 的疼痛脑网络分析结果

阈值\参数	特征路径长度	聚类系数	节点度数	全局效益
0.1	1.316667	0.790677	10.25	0.841667
0.15	2.033333	0.588393	6.5	0.638889
0.2	3.341667	0.524752	4.25	0.370833
0.25	3.691667	0.455357	3.125	0.2875
0.3	4.341667	0.3125	2	0.154167
平均值	2.945	0.534336	5.225	0.458611

表 3-2 被试 1 的静息脑网络分析结果

阈值\参数	特征路径长度	聚类系数	节点度数	全局效益
0.1	1.15	0.86755	12.75	0.925
0.15	1.341667	0.766674	9.875	0.829167
0.2	1.575	0.614644	7.125	0.729167
0.25	2.266667	0.349603	4	0.544444
0.3	3.066667	0.075	2	0.377083
平均值	1.88	0.535	7.15	0.681

表 3-3 被试 2 的疼痛脑网络分析结果

阈值\参数	特征路径长度	聚类系数	节点度数	全局效益
0.1	1.091667	0.913647	13.625	0.954167
0.15	1.325	0.752001	10.125	0.8375
0.2	1.8	0.552778	5.875	0.665278
0.25	2.791667	0.43254	4.375	0.465278
0.3	3.458333	0.277083	2.625	0.315278
平均值	2.093333	0.58561	7.325	0.6475

表 3-4 被试 2 的静息脑网络分析结果

阈值\参数	特征路径长度	聚类系数	节点度数	全局效益
0.1	1.041667	0.962798	14.375	0.979167
0.15	1.175	0.889345	12.375	0.9125
0.2	1.316667	0.820962	10.25	0.841667
0.25	1.441667	0.783211	8.375	0.779167
0.3	1.608333	0.599616	6.125	0.701389
平均值	1.316667	0.811186	10.3	0.842778

表 3-5 被试 3 的疼痛脑网络分析结果

阈值\参数	特征路径长度	聚类系数	节点度数	全局效益
0.1	1.2	0.81327	12	0.9
0.15	1.575	0.403716	7	0.726389

续表

阈值\参数	特征路径长度	聚类系数	节点度数	全局效益
0.2	3.191667	0.272024	3.5	0.380556
0.25	3.966667	0.222917	2.125	0.219444
0.3	4.425	0.135417	1.125	0.120833
平均值	2.871667	0.369469	5.15	0.469444

表 3-6 被试 3 的静息脑网络分析结果

阈值\参数	特征路径长度	聚类系数	节点度数	全局效益
0.1	1.025	0.975435	14.625	0.9875
0.15	1.108333	0.915495	13.375	0.945833
0.2	1.233333	0.846198	11.5	0.883333
0.25	1.383333	0.812699	9.25	0.808333
0.3	1.558333	0.605114	7.125	0.731944
平均值	1.261667	0.830988	11.175	0.871389

以 0.2 为阈值,画出 3 位被试的脑网络连接图,分别如图 3-78、图 3-79 和图 3-80 所示。从图中可以清楚看到,与静息状态相比,疼痛状态下脑网络的连接明显变少,此时大脑处理问题的效率明显降低。

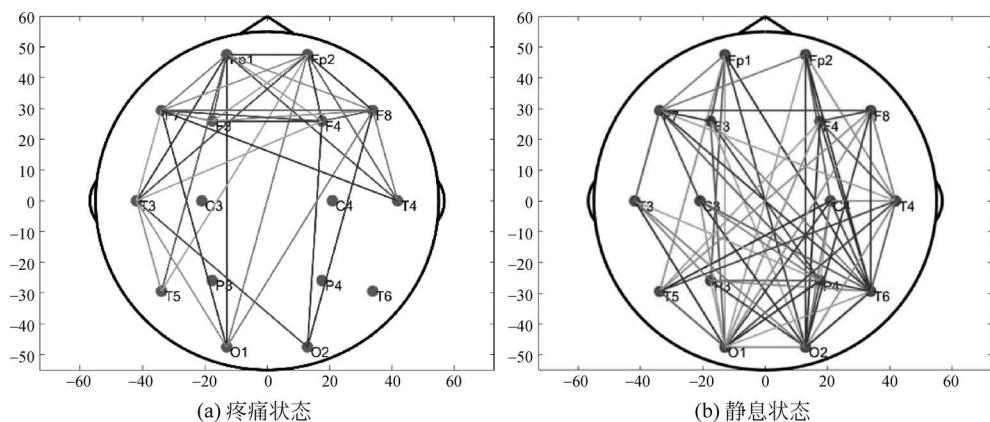


图 3-78 被试 1 脑网络连接图

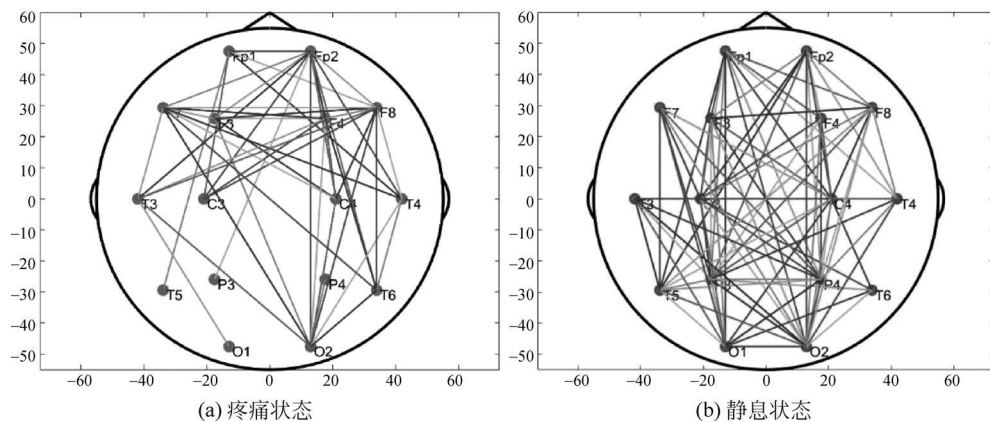


图 3-79 被试 2 脑网络连接图

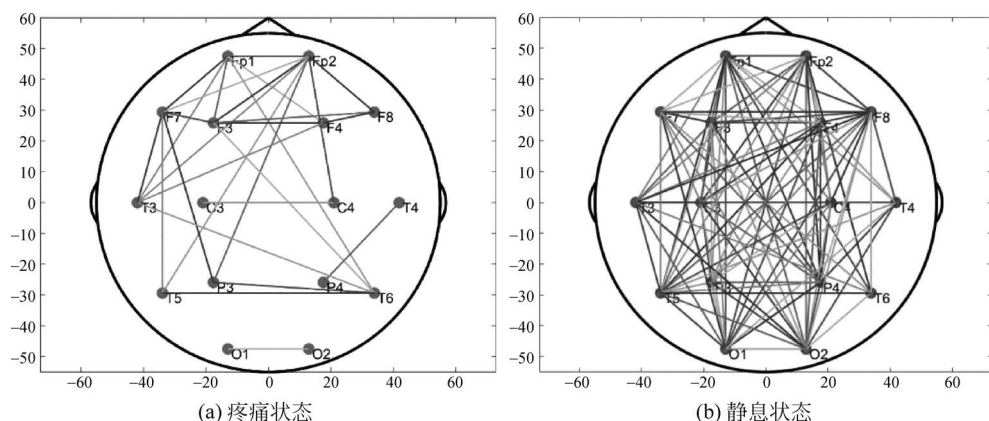


图 3-80 被试 3 脑网络连接图

3.6 脑电综合案例分析(抑郁症脑电的分类)

本节将以对抑郁症 EEG 数据集的数据进行分类为例,介绍前 5 节的方法在处理实际问题中的综合应用。

3.6.1 问题提出

抑郁症是一种常见的情感障碍性疾病,核心症状主要为心境低落、兴趣丧失以及精力缺乏,已经在全球范围内严重影响到人们的日常行为和生活,严重者会有自杀等行为,因此对抑郁症患者进行尽早的诊断和治疗十分必要。并且脑电信号反映了大脑皮层神经元细胞群自发性、节律性的电生理活动,含有丰富的生理与病理信息,因其非侵入式的操作、费用较低、时间分辨率高等特点成为临床脑神经与精神疾病诊断的重要依据。

3.6.2 数据分析流程

数据分析的流程为:

- (1) 载入数据,将原始数据载入 MATLAB,整理好文件名以便于批处理。
- (2) 数据预处理,步骤包括:①进行 0.5~35Hz 滤波;②手动去除伪迹;③按 75000 个时间点截取,再按 2s 的长度分段。
- (3) 特征提取,利用前几节介绍的方法,提取数据的功率谱特征,时频分析特征,非线性特征,脑网络特征等。
- (4) 分类,利用机器学习的方法(如 SVM)和深度学习的方法(如 CNN)对抑郁症患者和健康被试的数据进行分类。

3.6.3 基于 SVM 分类

SVM 是一种二分类模型,它将实例的特征向量映射为空间中的一些点,SVM 的目的是画出一条线,以“最好地”区分这两类点,如果以后有了新的点,这条线也能作出很好的分类。SVM 适合中小型数据样本、非线性、高维的分类问题。

SVM 学习方法包括构建由简至繁的模型：线性可分 SVM、线性 SVM 及非线性 SVM。
 (1) 当训练数据线性可分时,通过硬间隔最大化,学习一个线性分类器,即线性可分 SVM,又称为硬间隔 SVM;
 (2) 当训练数据近似线性可分时,通过软间隔最大化,也学习一个线性分类器,即线性 SVM,又称为软间隔 SVM;
 (3) 当训练数据线性不可分时,通过使用核技巧及软间隔最大化,学习非线性 SVM。

MATLAB 中使用一些自带的内置函数即可完成 SVM 分类,svm.m 代码如下:

```
% %
% 样本数:(24 + 24) * 150 = 7200;特征大小:1 * 3(第一列:lzc;第二列:小波熵;第三列:分形维数)
clc;
clear;
close all;
tic
% % 加载数据和标签
% NC - 健康 P - 患者
all_datas_NC = csvread('D:\Desktop\书稿校对\材料\3.6 数据集及代码\计算特征结果\
nonlinearNC.csv',0,0)
all_datas_P = csvread('D:\Desktop\书稿校对\材料\3.6 数据集及代码\计算特征结果\
nonlinearP.csv',0,0)
all_samples = [all_datas_NC(:,1:3);all_datas_P(:,1:3)]
all_labels = [all_datas_NC(:,4);all_datas_P(:,4)]
% % 归一化
% all_samples_reverse = mapminmax(all_samples',0,1)
% all_samples = all_samples_reverse'

% %
K = 5; % K 折
accuracy = zeros(K,1)
all_precision = zeros(K,1)
all_recall = zeros(K,1)
indices = crossvalind('Kfold',all_labels,K); % 生成交叉验证的索引
for k = 1:K % K iterations
    cv_test_idx = find(indices == k); % 交叉验证中第 k 折测试数据的索引
    cv_train_idx = find(indices ~= k);
    train_data = all_samples(cv_train_idx,:);
    test_data = all_samples(cv_test_idx,:);
    train_label = all_labels(cv_train_idx,:);
    test_label = all_labels(cv_test_idx,:);
    model = fitcsvm(train_data,train_label,'ClassNames',{ '0','1'},'KernelFunction','gaussian');
    [pre_label,score] = predict(model,test_data);
    pre_label = str2num(char(pre_label))
    C = confusionmat(test_label,pre_label); % 'Order'指定类别的顺序
    c1_p = C(1,1) / (sum(C(:,1)) + 0.001);
    c1_r = C(1,1) / (sum(C(1,:)) + 0.001);
    c1_F = 2 * c1_p * c1_r / (c1_p + c1_r);
    fprintf('c1 类的查准率为 %f,查全率为 %f,F 测度为 %f\n\n',c1_p,c1_r,c1_F);

    c2_p = C(2,2) / (sum(C(:,2)) + 0.001);
    c2_r = C(2,2) / (sum(C(2,:)) + 0.001);
    c2_F = 2 * c2_p * c2_r / (c2_p + c2_r);
    fprintf('c2 类的查准率为 %f,查全率为 %f,F 测度为 %f\n\n',c2_p,c2_r,c2_F);
    accuracy(k,:) = (C(1,1) + C(2,2))/sum(sum(C))
    all_precision(k,:) = (c1_p + c2_p)/2
    all_recall(k,:) = (c1_r + c2_r)/2
end
acc_ave = mean(accuracy)
```

```
rec_ave = mean(all_recall)
pre_ave = mean(all_precision)
```

五折交叉验证结果为 $\text{acc_ave} = 0.6385$, $\text{rec_ave} = 0.6385$, $\text{pre_ave} = 0.6422$ 。

3.6.4 基于深度学习分类

CNN 是一种具有局部连接、权值共享等特点的深层前馈神经网络,是深度学习的代表算法之一,擅长处理图像等相关机器学习问题,比如图像分类、目标检测、图像分割等各种视觉是目前应用最广泛的模型之一。

一个 CNN 主要由以下 5 层组成:数据输入层、卷积计算层、ReLU 激励层、池化层和全连接层。

MATLAB 中使用一些自带的内置函数即可完成 CNN 分类, `cnn.m` 代码如下:

```
% %
% 样本数:(24 + 24) * 150 = 7200;特征大小:1 * 3(第一列:lzc;第二列:小波熵;第三列:分形维数)
clc;
clear;
close all;
tic
% % 加载数据和标签
% NC - 健康 P - 患者
all_datas_NC = csvread('D:\Desktop\书稿校对\材料\3.6 数据集及代码\计算特征结果\
nonlinearNC.csv',0,0)
all_datas_P = csvread('D:\Desktop\书稿校对\材料\3.6 数据集及代码\计算特征结果\
nonlinearP.csv',0,0)
all_samples = [all_datas_NC(:,1:3);all_datas_P(:,1:3)]
% 输入格式需要转置,行代表特征
all_samples = all_samples'
all_labels = [all_datas_NC(:,4);all_datas_P(:,4)]
all_labels = all_labels'
% %
K = 5; % K-fold CV
accuracy = zeros(K,1)
indices = crossvalind('Kfold',all_labels,K); % 生成交叉验证的索引
layers = [...
    imageInputLayer([3 1 1]); % 输入层,要正确输入 height,
    % 图像的宽度和通道数量
    batchNormalizationLayer(); % 批量归一化
    convolution2dLayer([3,1],16,'Padding','same'); % 卷积层
    batchNormalizationLayer();
    reluLayer() % ReLU 激活函数
    % maxPooling2dLayer(2,'Stride',2); % 池化层
    fullyConnectedLayer(16); % 全连接层
    fullyConnectedLayer(2); % 全连接层
    softmaxLayer(); % SoftMax 层
    classificationLayer(),...
];
options = trainingOptions('sgdm',... % 也可以用 adam、rmsprop 等方法
    'MaxEpochs',5,... % 最大迭代次数
    'Plots','training-progress');
for k = 1:K
    cv_test_idx = find(indices == k); % 交叉验证中第 k 折测试数据的索引
    cv_train_idx = find(indices ~= k);
    train_data = all_samples(:,cv_train_idx)
```

```

train_data = reshape(train_data, [3,1,1,5760])
test_data = all_samples(:,cv_test_idx)
test_data = reshape(test_data, [3,1,1,1440])
train_label = categorical(all_labels(:,cv_train_idx))
test_label = all_labels(:,cv_test_idx)
net_cnn = trainNetwork(train_data,train_label, layers,options);
testLabel = classify(net_cnn,test_data);
test_label = test_label'
testLabel = double(testLabel)
precision = sum(testLabel == test_label)/numel(testLabel);
disp(['测试集分类准确率为',num2str(precision*100),'%'])
accuracy(k,:) = precision
end
acc_ave = mean(accuracy)

```

结果如图 3-81 所示。

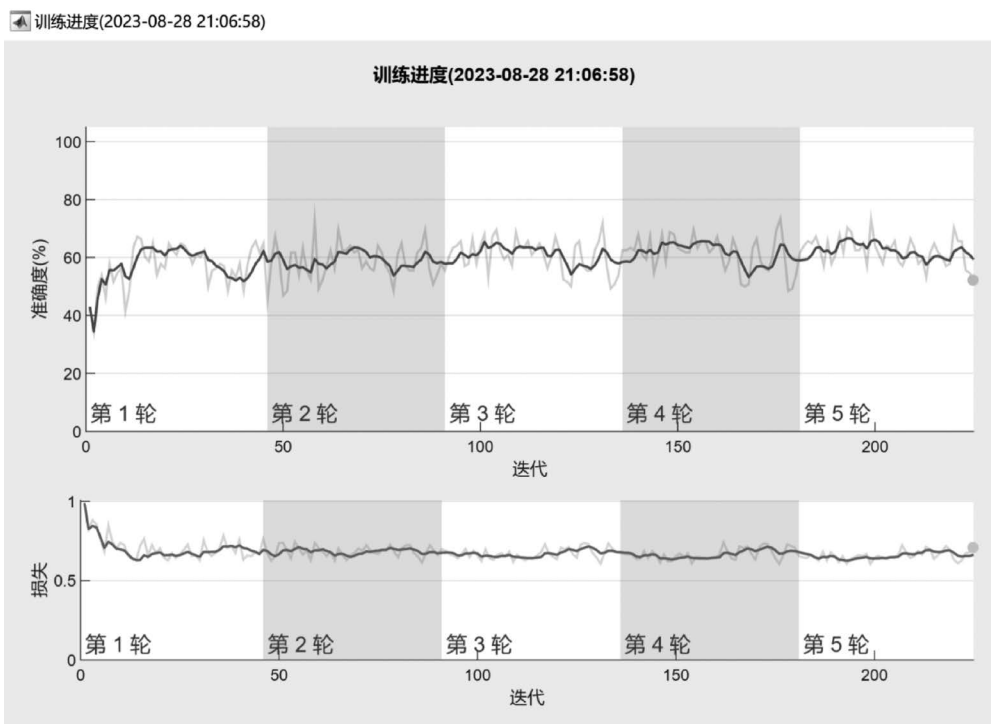


图 3-81 训练进度

参考文献

- [1] 葛哲学,陈仲生. MATLAB 时频分析技术及其应用[M]. 北京: 人民邮电出版社, 2006.
- [2] Li Y, Tong S, Liu D, et al. Abnormal EEG complexity in patients with schizophrenia and depression [J]. Clinical Neurophysiology, 2008, 119(6): 1232-1241.
- [3] Méndez M A, Zuluaga P, Hornero R, et al. Complexity analysis of spontaneous brain activity: Effects of depression and antidepressant treatment[J]. Journal of Psychopharmacology, 2012, 26(5): 636-643.
- [4] Akar S A, Kara S, Agambayev S, et al. Nonlinear analysis of EEGs of patients with major depression

- during different emotional states[J]. Computers in Biology and Medicine, 2015, 67: 49-60.
- [5] 张胜, 乔世妮, 王蔚. 抑郁症患者脑电复杂度的小波熵分析[J]. 计算机工程与应用, 2012, 48(4): 143-145.
- [6] 盖淑萍, 刘欣阳, 刘军涛, 等. 抑郁症静息脑电的小波包节点功率谱熵分析[J]. 传感器与微系统, 2017, 36(3): 6-9.
- [7] Ahmadlou M, Adeli H, Adeli A. Fractality analysis of frontal brain in major depressive disorder[J]. International Journal of Psychophysiology, 2012, 85(2): 206-211.