尊敬的王老师：

展信佳！一周的学习时光又圆满结束了，在此向您报告本周动向与收获，请多指教。

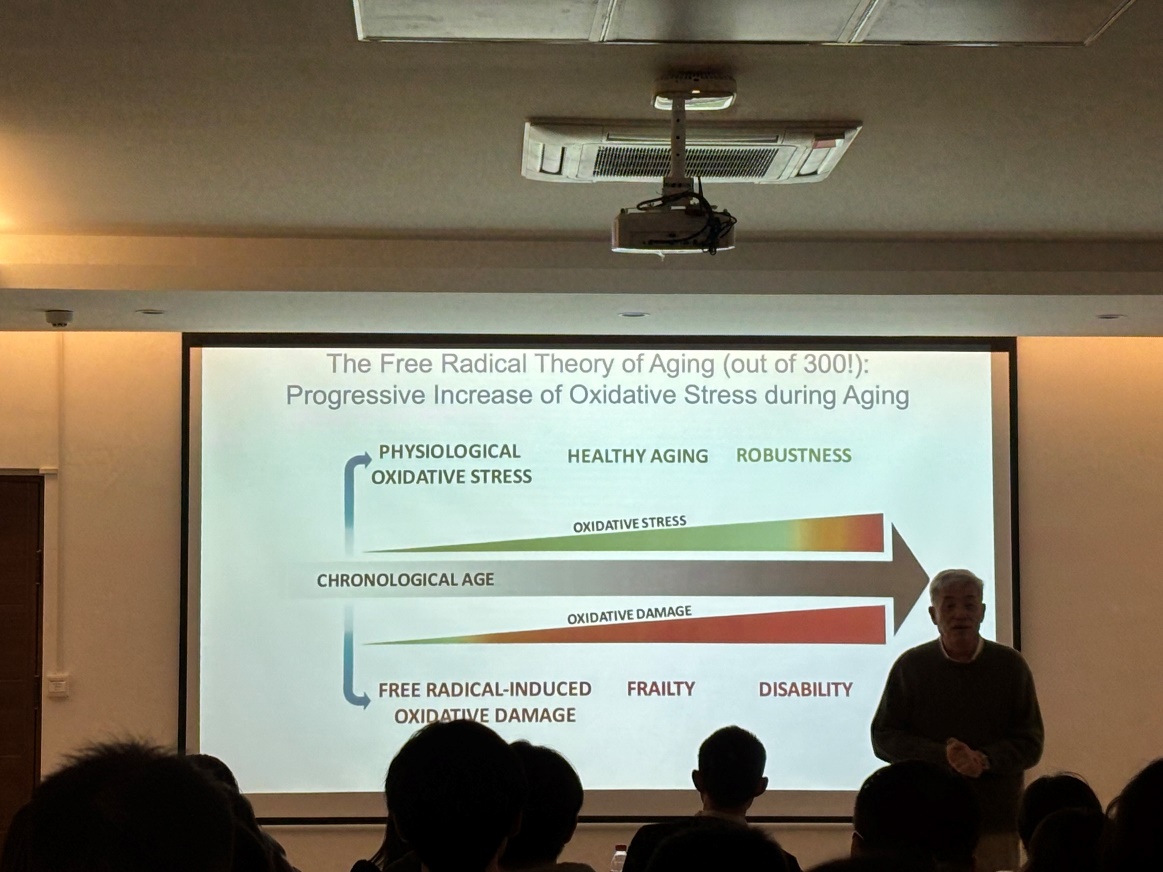
一、西湖大学教授讲座

1.付向东教授

与一位同学打车前往，雨夜堵车迟到7分钟，所幸付教授正好刚开始讲座。主要框架如下：

* Path to RNA Biology
* Switch from RNA Biology to Neuroscience
* Why neurodegenerative diseases undergo age-dependent onset?
* What determines the lifespan

针对前两点，教授主要介绍人生经历，1977年恢复高考后考入武汉大学，参与中美生物化学联合招生项目后研究RNA多年，后转向研究神经科学。其中最令我印象深刻的是他提起多次的名言：“You will be the best scientist in your own lab.”这或许代表生物学家需具备的自信，或许象征科研者的乐观态度，抑或诙谐的自勉，总之据教授所言，这句话激励着他走好每一步。

第三点中，教授谈到了自由基理论，我在高中课本接触过故而印象很深。随年龄增长，氧化应激逐渐加重并影响身体健康与衰老，而自由基导致的氧化损伤愈发严重，导致虚弱与残疾，影响寿命。

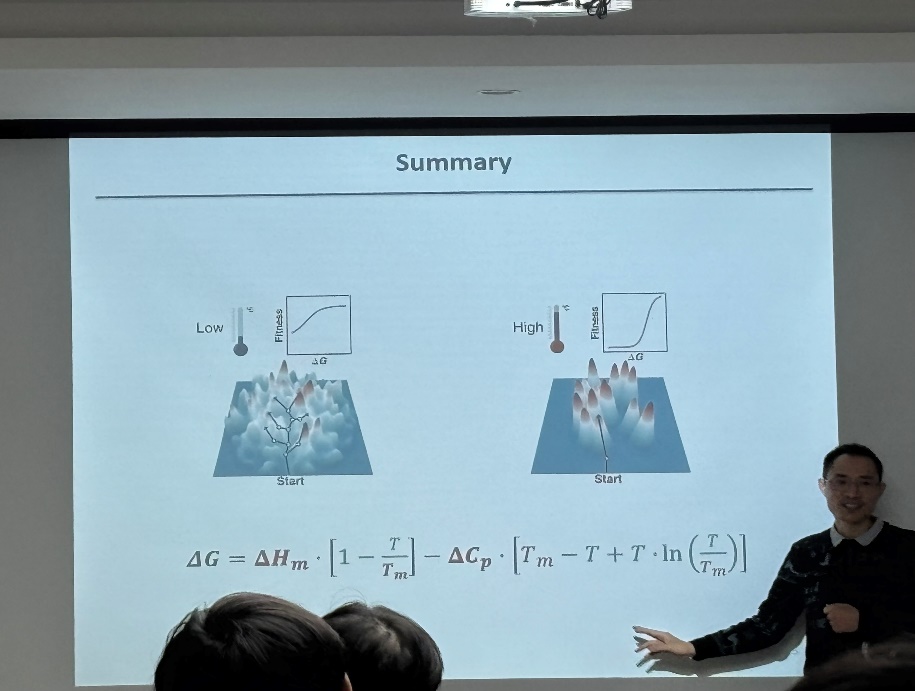
第四点中，教授谈论了生命的奥义，并以格陵兰鲨为例（心脏每分钟跳5次，而寿命达500岁），抛出“生命是否在于静止”这一趣问，但限于时间他并未细谈，比较遗憾。

最后教授简要介绍了西湖大学，其科研层次的高度众所周知。

2.郑甲教授

郑教授分享了他所在团队的最新研究成果。他从生命起源引入，指出地球上的温度在短期和长期尺度上都发生了波动，但这些变化影响产生新表型的速度尚不清楚。一方面，高温可能会加速表型进化，因为它会加速大多数生物过程；另一方面，它可能会减慢表型进化，因为蛋白质通常在高温下不太稳定，因此进化性较差。在这里，为了通过实验验证这些假设，教授在大肠杆菌中进化出绿色荧光蛋白，在不同温度下朝着黄色荧光的新表型发展。

经过一系列实验（我承认我听不太懂）最终验证出低温促成进化。



3.何丹阳教授

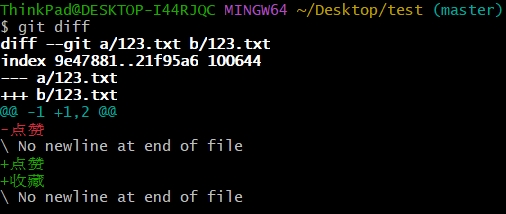
我能分辨出，何教授的分享与微信公众号上关于她研究方向的介绍是相契合的，故在此摘抄：研究调节小胶质细胞介导的突触重塑和脑内稳态的重要功能基因模块，从而阐释多个自闭症风险基因在小胶质细胞中的作用，直接证明了小胶质细胞功能障碍会导致神经发育缺陷与精神疾病的发生。因此得出，免疫信号通过调节神经元突触的发育和成熟来影响神经环路的功能。

4.感悟

路漫漫其修远兮！几位教授分享的内容非常专业，以至于我很难抓住他们的思维动向，看来生物专业知识以及实验思维的不足确实是有较大影响的。但同时也有不少收获，比如拔高了生物思维高度、认识到自己的知识缺陷、在问答环节见识了与学长学姐的差距、了解了西湖大学的招生政策等等。

二、*Bioinformatics Data Skills*阅读

1.GIT基操

当文件未在暂存区时被改动，使用git diff命令可以查看改动内容，绿色为增加的信息，红色为删减的信息。比如我在本有“点赞”字样的txt文件中增添“收藏”字样会出现下图信息。

但如果将文件放入暂存区后再命令就无事发生了，因为当前文件与暂存文件是一致的。

使用git log命令可以查看commit的历史。

使用git mv命令可以对文件进行移动或重命名等操作。比如：

git mv 123.txt 1234.txt

这样就可以把文件名改成1234.txt。

使用git rm命令可以删除指定文件。有个小发现：只有将修改后文件commit后才能进行git mv和git rm，否则识别不了。

随着分析的进行存在大量的文件的时候，继续使用git status显示大量非追踪文件是一种负担。此时可在工作区创建名为.gitignore的文件，来忽视不想显示的文件。以下是作者建议忽略的文件：原始数据。这些数据特别大，进行clone的时候会非常缓慢，推荐采用别的方式共享；部分中间结果。某些中间结果数据量也非常大而某些重新生成非常容易，这些文件建议忽略；编辑器生成的临时文件。例如vim和emac编辑器的临时文件example.txt~和#example.txt#；程序生成的临时文件。

针对某些无论如何都应该忽略的文件，可以在目录下设定一个全局的忽视文件~/.gitignore\_global，所有工程的git配置都可以通过以下命令共享：

git config --global core.excludefile ~/.gitignore\_global

为了实现版本控制的组织性与纪律性，git的每一次commit的版本和版本描述最好是有阶段性意义的。如果我们无意中将一份修改一半的文件放入暂存区，可以采用git reset命令来移出。

2.Github使用

GitHub是一个面向开源及私有软件项目的托管平台，只支持Git作为唯一的版本库格式进行托管，可以用于代码的存储与协作工作。使用流程为：

创建一个共享的库，你和同事均有权限访问

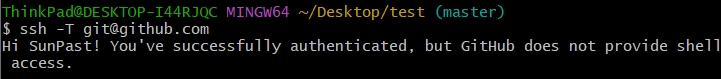
你将工程push到github

你的partner clone github库的工程到本地

你的partner修改本地的工程并commit，通过push提交到github库

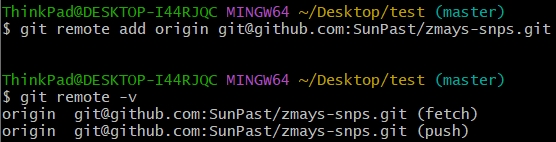
你将github库上的修改pull到本地

这个过程反复进行，即可共同修改库的文件。

注册github账号后，创建一个远程库。在git中输入ssh-keygen -t rsa以本地创建密钥，再输入cat ~/.ssh/id\_rsa.pub即可查看公钥。在github网站的setting中点击SSH and GPG keys，在New SSH Key中粘贴刚刚获取的公钥即可认证身份。最后在git中输入ssh -T [git@github.com](mailto:git@github.com)以验证是否成功，如图即可。

在git中输入如下命令可连接github库：

git remote add origin git@github.com:SunPast/zmays-snps.git

再输入git remote -v可查看本地已知的库，如图所示

想要删除库，使用git remote rm <库名字>命令。

使用git push origin master命令可将本地内容上传至github库中。

作为同事，则可以使用git clone来克隆github库至新文件夹zmays-snp-partner，比如：

git clone [git@github.com:SunPast/zmays-snps.git](mailto:git@github.com:SunPast/zmays-snps.git) zmays-snp-partner

使用git pull origin master命令则将远程的改变同步到本地。

假如有两名工作者在同一文件的同一地方添加了信息，并进行push，后push的一方会收到报错信息，表明文件存在冲突，需要手动处理。

可以通过以下4个步骤来解决冲突问题：

（1）使用git status定位起冲突的文件

（2）手动处理冲突部分内容，其中冲突部分被显著标出

（3）修改完冲突文件后进行git add

（4）通过git status命令确认冲突完全解决

解决冲突之后须尽快push文件到github库，以便进行其他协作者接下来的修改。

3. 操作已提交的commit

（1）如果不小心损坏了文件内容，可以使用git checkout恢复文件。作者以不小心覆盖README文件为例，可采取如下命令恢复：

git checkout -- README.md

若想恢复到特定的commit版本，可先采取如下命令查看最近的版本：

git log --pretty=oneline --abbrev-commit -n 3

再根据版本号hash值（可以使用前7位缩写），将文件恢复到到特定的版本：

git checkout db64487 -- README.md

须注意，如果git checkout后不加文件参数，整个branch都会恢复到前一个状态。

（2）使用git stash可保存临时改变。git stash命令会将上一个commit之后修改的内容另行存储，并将工程恢复到上一个版本的状态。如果想要git stash存储的更改生效的话，则使用git stash pop命令。

（3）使用git diff对比不同版本的文件内容。前文中，作者介绍git diff命令可用于比较当前文件与暂存区文件的区别。事实上，git diff可以用于比较任意两个版本文件的区别。以下是作者的两个例子。

比较当前README.md文件与某一个版本的区别：

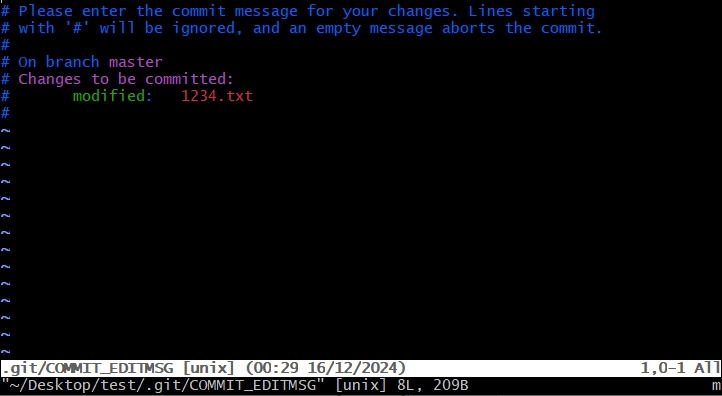
git diff db64487 -- README.md

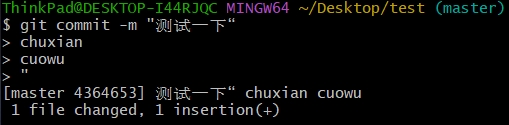
比较某两个版本README.md文件的区别：

git diff db64487 cb17f02 -- README.md

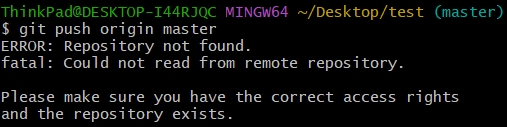
（4）如果提交了一个commit，但是描述的信息是错误的，可使用git commit –amend命令修改commit信息。输入此命令后会弹出一个编辑器，在里面修正信息并保存即可。

三、对上周笔记的补充

1.git commit后不加-m会导致下图情况出现，按任何键都无法退出，着实困扰。但现在有解决方法了：按esc键后输入“:q”并回车即可安全退出。

2.输入git commit -m“内容”后另起一行出现>符号且无法输入命令。一番搜索后知晓原因：git把下引号识别为输入的内容，必须再来个”才能结束识别。全过程如图。

四、问题求助

1.其实在git push命令这一步我就遇见了无法解决的问题。在输入git push origin master后出现报错，如图。

在网上查了很久，有非常多的解决方案，包括重新添加公钥、检查url等等，都无法解决，着实苦恼。不知道老师有什么经验可以传授一下，帮我了结这个心结，谢谢！

2.关于git学习，章节最后一个主题是branch。但是据作者所言，如果到目前为止对 Git 感到不知所措（作者称自己第一次学习时确实如此），可以继续往后阅读，过段时间再学习本节。就目前而言，我对于git知识的掌握貌似有种半桶水晃荡的感觉。老师认为我有必要先学习这块内容让git部分结束吗？还是说先跳过，等日后补充了更多知识再学呢？我担心过段时间会对git有所生疏而更学不明白。

感谢老师能读完我的报告，非常期待您的指点。

丁加

2024年12月15日