LISA数据分析研讨会的7个TUTORIAL文件的详细翻译分析：

注释1：我们在除了翻译之外，还会对这里面内容的每一个地方进行注释和分析，

特别的提出一些想法和意见。然后我们特意保留了一些原本就有进行使用的函数的英文名，以求阅读方便。

自行添加的思考和注释内容用~&\*符号联合标记

注释2：我们知道TUTORIAL\_ANSWERS当中的七个TUTORIAL\_COMPLETE文件当中有源代码，

但是我们也还是根据自己的知识以及结合了引力波的物理学实质做了更为细致的分析和思考。

有进行修改和分析整理的地方我会用额外的注释标注。

（包括对于我们这整个研究计划和系统当中其他代码和文件）

对于这七个TUTORIAL文件内容的整体解析和思考：

教程0:LAT研讨会的介绍材料

在本入门教程中，我们将对一些Python主题进行基本概述，

这些主题对于理解和使用在LISA Analysis Tools环境中开发的代码至关重要。

您还将初步了解教程的结构。

这个模块应该在研讨会之前完成。它还将会确保环境和包的安装能够成功

·任务:在整个教程中，您将被分配“任务”。这些将指导教程方法，并将教程自然地分成几块。

·问题:在较大的任务中会给出附加问题。这些工作有时涉及写代码，

有时侯则不涉及。其中很多都是一些设计来让你和你的小组讨论的问题。

·结构:根据手头的任务，我们将提供有助于完成任务的具体指导

（注：这里原文写错了单词，应该是instruction,少了一个s）、

少量的对于完成任务有用的代码和文档。为了使这些笔记本成功运行，

用户必须完成大部分或全部实际代码编写工作。

注意:在许多情况下，特定代码库中的教程也可以让您接近正确答案。

大多数情况下，用户必须浏览这些教程才能找到有帮助的答案。

任务1:构建并初始化一个类对象。

类是构建LISA分析工具及其相关代码的基本对象。

理解和使用类将最大化您对LISA分析工具的理解和能力。

构建一个类对象有两个方法:\_\_init\_\_和get\_wave。

在\_\_init\_\_中，存储频率，幅度和初始相位。

在get\_wave中输入一个时间数组作为参数。

然后使用存储在self和时间数组中的频率、幅度和初始相位返回正弦函数结果给用户。

有用的文档:

Python类对象

通过提供必要的参数实例化类。为时间数组构建numpy数组并测试您的类输出。

任务2:\*args和\*kwargs

使用lisatools的另一个重要主题是\*args和\*kwargs的使用。

这些都是灵活传递参数(\*args)和关键字参数(\*\*kwargs)的方法。

在这里，我们将以几种方式进行实验。

有用的文档:

例子文件（examples）

我们将有效地复制这些例子。

首先开始于设计一个函数，使它接受灵活数量的参数并将它们打印出来。

这个函数的名字将会是func(\*args)。

在这种情况下，args将是一个可迭代对象，因此您可以使用for循环遍历所有的参数。

在\*args函数被定义之前向函数中添加一些参数，并观察其变化情况。

现在对func(\*\*kwargs)函数做同样的事情。

这将产生一个字典（这里的字典是一个电脑语言的术语），然后所以现我们在for循环中输出它。

\*\*kwargs在使函数变得灵活的场景当中非常有用，因为您可以让它字段所需的特定kwargs，

并在\*\*kwargs字典中存储任何其他内容。添加关键字参数到函数签名中，

看看它是如何改变打印结果的。

您还可以有效地执行反向操作。

您可以在具有\*args的函数调用中输入一个可迭代项，它将把可迭代项扩展到函数签名中的args。

只使用2个参数进行函数调用。

然后，创建一个包含两个条目目的列表。将其输入到函数调用中，并在函数中打印输出。

同样类型的东西工作与任何输入字典称为像\*\*dict。做和上面对于kwargs一样的事情。

任务3：类继承

类继承会变得更复杂一些。

对于研讨会，特别了解这一点并不必要，

但它可以真正帮助您理解LISA分析工具和其他为未来开发和维护而设计的抽象代码库。

从上面复制你的类。向\_\_init\_\_函数添加一个关键字参数，

该参数为正弦函数(一个float值)从零获取垂直偏差。

将其存储到self。在get\_wave函数的内部，将这个值添加到正弦返回值的末尾。

注意：请确保任何的为了A而进行输入的输入值与这个附加的偏差因素没有太多数量级。

否则，您可能会遇到机器精度方面的问题。

（机器精度无法很好地解决qaq,只能对于机器计算的底层做进一步的改进与探索才行）

现在，我们将创建一个类，它将继承上面的类，并在计算过程中调整其输出。

向此类中添加两个方法：

在\_\_init\_\_中，让它包含一个表示乘法因子（浮点值）的单一参数。

把它放在自己的地方

。然后，在函数签名中，通过添加可调数量的参数和关键字参数来确保它是灵活的。

将这些参数和关键字参数传递给具有以下类型的父类：

超级(ChildClass,自我). \_\_init\_\_ (\* args, \* \* kwargs)

\_\_在函数上的调用\_\_方法允许用户初始化类并将其作为函数或可调用传递。

换句话说：c=类（）->y=c(x)，而不是必须命名一个特定的方法，如get\_wave。

有用的文档：\*\_\_调用\_\_（这是一个可点击的链接）

在\_\_调用\_\_中，取一个参数，即时间数组（numpy数组），并将其放入get\_wave方法中。

取get\_wave的输出，然后乘以存储的乘法因子。请将其返回给用户。

现在，请初始化并测试您的类。

一个pdb宣言:

(迈克尔的意见)

python调试器（pdb）是一个令人难以置信的有用工具。

对我来说，它与Numpy和Matplotlib一样有用。

它从开发过程中节省了很多时间，老实说，这是不可估量的。

PDB允许您在代码的任何地方使用大多数单行python命令。

换句话说，您可以停止Python代码并运行函数，打印任何存储的变量，存储新变量，绘制内容等。

我注意到很多人不知道这个内在的python库，或者不使用它。

我的开发代码文件的开头通常是：

import numpy as np

if \_\_name\_\_ == "\_\_main\_\_":

#调试器在最后捕获和分析任何输出。

breakpoint()

```

我经常这样做，我现在在写这个文章时称之为“主动调试”。

主动调试意味着运行代码，直到它达到我将要开发的点。

然后我将在在调试器中一行一行地进行，然后依次测试每一行。

一旦它工作起来了，它就会被粘贴到代码中。

这有时候（注意仅仅只是有时）有点乏味，但我强烈建议尝试一下。

在我看来，这是一种非常有效的进行代码开发的方法。

我们不打算在这里练习它，因为它在笔记本上工作得不太好。它在终端上更有用。

（这里的notebook应该指的是jupyter notebook的notebook而不是我们传统意义上的纸质笔记本）

在终端中，这是一种几乎类似于ipython的体验，你可以在代码中放置断点的任何地方停止。

您可以使用它来停止在您设计的函数内部，或者在其他代码的python代码内部。

换句话说，您可以打开‘$PATH’中一串的python代码，并在其中放置一个断点。

然后，您可以运行调用这个包的代码。它将在您的断点处停止。它将在您的断点处停止。

你可以逐步浏览并打印任何你需要的东西。

在过去，这确实帮助我学习了更好的编码技术，并理解了别人设计的复杂代码。

下面是pdb的具体文档：

\* [pdb](https://docs.python.org/3/library/pdb.html)

对于TUTORIAL0的总结:

TUTORIAL1:

LAT研讨会教程1:

灵敏度曲线，信噪比，内部产品以及可能性

问题1：

你认为灵敏度曲线模型的差异会如何影响LISA观测结果?

考虑低频率或高频率的高噪声会对我们不同的源类造成什么影响。

问题2：

我们已经要求你计算天空平均灵敏度曲线。这是什么意思？

你能在不看文档的情况下确定总的天空平均系数是什么吗？

现在我们要把敏感度信息放入敏感度矩阵中。

建立一个双信道（没有跨信道）的灵敏度矩阵，其中两个信道的灵敏度都由LISASens敏感函数管理。

有用的文档：

敏感的矩阵

我们将在这里添加一个额外有用的信息：一个包括在灵敏度分析之内的银河前景噪声。

这在执行LISA数据分析时很重要。

默认的前台功能为：它只有一个参数:以秒为单位的观察时间。

使用灵敏度函数和随机函数的文档来确定如何通过lisatools中的抽象设置传递此信息。

有用的文档:

LISASens

FittedHyperbolicTangentGalacticForeground

用矩阵的对数方法把它画出来。

比较有前景混淆噪声和没有前景混淆噪声的噪声曲线。

然后比较两个不同观测时间值的前景噪声。

有用的文档：

随机\_更新

问题

为什么前景中的混乱会随着时间的推移而变化？这一更广泛的含义是什么

在我们的分析？

任务2:构建一个DataResArray来存放和绘制数据信息。

（这个词由三个部分组成，个人认为应该指的是数据资源箭头？

应该代表的是用来指向数据的一种指针箭头文件）

我们要建立一个正弦信号给我们来做一些计算。我们会把它储存在一个DataResArray中。

建立一个有4个参数的正弦信号:

a(幅度)，

fo(频率)，

fo（频率的导数（关于时间的））

φo（初始相位）

公式如下：（latex格式，附带图片）

$$ h\_1(t) = A\sin{\left(2\pi (f\_0 t + \frac{1}{2}\dot{f}\_0t^2) + \phi\_0\right)} $$

$$ h\_2(t) = A\cos{\left(2\pi (f\_0 t + \frac{1}{2}\dot{f}\_0t^2) + \phi\_0\right)} $$

选择你的参数，并且为LISA使用一个合理的时间步长和观测时间，最后绘制$h\_1$和$h\_2$

（下面是一个例子）：

现在我们将把两个通道（h1，h2）添加到DataResArray中。

有用的文档：

DataResidualArray

用数据残差阵列绘制函数绘制信号。

有用的文档：

loglog

问题3

当您从上面输入时域信号时，DataResidualArraay类在引擎盖下面进行转换。

看看上面的输出，它做了什么转换？

让我们使用numpy(np.fft.rfft)来获取输入时域数据集的FFT

（Fast Frourier Transfrom,快速傅里叶变换），

并将其添加到上面的图中。

我们所做的FFT和存储在数据剩余阵列中的频域谱之间的因素差异是什么？

这个词从何而来？为什么会有它？（这是一个很容易被忘记的小细节）

任务3：合并到一个分析容器中。

现在，我们将使用我们的数据和敏感度信息，并将其合并到一个分析容器对象中。

这个对象通过在不同的数据和灵敏度矩阵下正常工作来帮助简化分析。

用数据和灵敏度信息绘制这一新的设置。稍后我们将用这个对象进行计算。

有用的文档：

1.分析容器

2.分析容器.loglog

任务4:计算内积。

现在，我们将向数据中注入一个信号，并计算另一个信号与该数据的内积。

我们将以data\_res\_arr作为数据输入。

使用与上面选择的注入略有不同的参数集生成一个新信号。

问题：

使用注入数据的新信号的内积值是多少？用数字计算用这个方程求内积.

$$ \langle h\_1 | h\_2 \rangle = 4 \text{Re}

\int\_{f\_\text{min}}^{f\_\text{max}}\frac{\tilde{h}\_1(f)^\*\tilde{h}\_2(f)}{S\_n(f)}df \ \ .$$

现在，使用在分析容器类当中的内积计算方法计算内积。

这些数字可能并不完全匹配。在引擎盖下，内积函数在lisatools中使用'np.trapz'函数。

如果你做一个和，你可能会得到一个非常小的，可以忽略不计的差异。

你在上面选择的参数的附近”玩“一下（这里的玩翻译自play，更多的是指在参数附近进行运行和研究）。

试着去理解内积是如何根据你所选取的参数去改变的（即它们与注入参数的距离有多远）。

任务5：计算信噪比。

有两个信噪比（信噪比）是LISA感兴趣的。一个是最佳的模板的SNR（Popt）。

这是模板在给定噪声比谱（h为模板）下可能实现的最大信噪比：

$$ \rho\_\text{opt} = \sqrt{\langle h | h \rangle}\ \ . $$

换句话说，如果注入数据与模板完全等价，它将实现这个信噪比。

另一个感兴趣的信噪比是检测到的信噪比(Paet)。它由以下公式给出：

在上方，你计算了，现在使用numpy计算最优信噪比和检测信噪比。

要使用' AnalysisContainer '对象计算这些值，请使用[template\_snr]

问题：

更改模板的参数会如何影响最佳信噪比和检测到的信噪比？

任务6：计算似然（或者说可能性Likelihood这个词有多种解释和考虑方式）

LISA(以及引力波的其他领域)中使用的似然是高斯似然，

它用残差数据与自身的噪声加权内积取代了典型项:(d-hi|a-h)让我们来分解一下。

1，我们说残差是什么意思？这是数据和模板之间的误差：

计算数据和模板的残差，并将其存储在一个新的DataResidualArray中。

2计算残差与自身的内积。提示：你可以把它放在一个新的地方分析容器并直接计算。

3.构建完整的可能性。高斯似然由以下这个公式给出：

我们通常使用似然（可能性）的自然对数：

用这种方法，我们也可以把内积分解成通常的部分，因为内积是线性的操作：

你应该会看到一些熟悉的术语出现:

在MCMC分析中，(aia)项通常在整个采样过程中保持不变。

您可以使用模板\_似然法根据数据计算模板的可能性

分析容器类中的方法。

问题：

改变这些参数会如何影响这个可能性呢？这与最优和检测到信噪比值？

在模板\_似然值中有两个有趣的关键字参数

方法： include\_psd\_info和phase\_maximize。可以找到相位最大化的参数

对于似然值和信噪比的计算，我们能够发现 phase\_maximize。

请阅读文档并添加这些文档的关键字参数，你能弄清楚它们在做什么吗?

任务7：添加一个信号模型。

现在，我们将在似然类中添加一个信号模型。

在实例化分析容器时，您可以将其设置为关键字参数。或者你也可以把它设置为事后的一个属性。

但首先，我们需要把正弦信号变成一个函数。

构建一个函数，其需要我们的四个参数(A、fo、fo和φo)，并返回一个列表[h1,h2]。

将时间数组添加为参数传递给生成器。所以它应该有一个这样的签名:

sinusoidal\_waveform(A:float，f0：float，fdot0:float,phi0:float,t\_arr：np.ndarray)

现在，通过更新signal\_gen属性，将信号模型添加到我们的似然类中。

用一组新的参数计算似然值。记得添加t作为最后一个参数的你的输入。

帮助文件：

AnalysisContainer.calculate\_signal\_likelihood

额外联系

现在重复教程的各个部分，调整信号或灵敏度模型。

观察它是如何改变内积、信噪比和可能性的。

有一个对于这些元素的好理解对LISA的数据分析至关重要！

1 LAT研讨会教程2：EMRLs 和 LISA响应+TDI

在第二个教程中，我们将会介绍

·生成EMRI波形，

·使用我们上一课中的工具来估计EMRI的信噪比，

·检查EMRI运动轨迹的变化，

·添加LISA响应函数，

·并计算EMRIs+响应+轨迹的可能性。

任务1：为一个典型的EMRI生成一个EMRI波形

在探测器框架中生成一个相对论性的Schwarzschild(施瓦西黑洞的那个施瓦西)偏心EMRI波形

(使用generateemriwaveform而不是FastSchwarzschildEccentricFlux)。

选择一组合理的参数。有用的文件：

GenerateEMRIWaveform和

FastSchwarzschildEccentricFlux

问题

为什么EMRI的波形是这样的？

这显然是一个非常开放的问题。

试着解释为什么波形在紧凑的物体轨道上有这样的结构。

任务2：添加LISA响应函数

当在时域中实际构建LISA响应函数时，数据的边缘可能会很复杂。

这是由于仪器响应和延时干涉(TDI)变换中包含的各种延迟。

简而言之，当前时间和过去时间的波形求和要求我们在观测开始后稍微开始计算:

如果你不能深入到过去足够的时间(即。延迟导致时间小于0。延迟导致时间小于0.0)。

现在，我们将从fastlisasresponse使用ResponseWrapper包装在LISA响应里面的EMRI波形

有用的文档:

ResponseWrapper

pyResponseTDI

EqualArmlengthOrbits

问题

通过足够长的观测时间，当添加响应时，您将看到波形的新包络结构被加入。为什么会这样呢？

如果你想在其他类型的波形上面使用RespanseWrapper，上面的cell就是你如何做的过程。

如果您想要一起使用generateemriwaveform和ResponseWrapper，

您可以使用LISA分析工具:emritdiwaveform中的股票模型。

使用这个类，您不必设置任何选项。

在这种情况下，它将设置所有默认值。在使用新类的情况下生成上面相同的波形。

文档:\*

EMRITDIWaveform

任务3：计算信噪比、内积和似然值

现在，我们将结合教程1和教程2中的知识来计算EMRI的信噪比、内积和可能性。

让我们首先只为A和E通道生成数据（现在不要担心T）。

请注意，确保在使用EMRITDI波形时，使用关键字参数response\_kwargs=dict（tdi\_chan=“AE”）

来初始化以删除T通道。

用数据填充DataResidualArray。

如果你愿意，就画出来。

现在让我们建立敏感矩阵。

请记住，我们现在已经包含了响应函数。

因此，我们不希望使用LISASens。相反，我们将使用TDI灵敏度函数:A1TDISens和E1TDISens.

现在将它们都打包到分析容器中，并计算注入数据的信噪比。

1.4任务4：使用可能性确定一维的标准差的宽度

在一维空间中，可以通过求解logC(θ)=-1来确定低似然宽度。

使用一个循环，绘制出与您决定更改的一个参数之间的可能性。

确保可能性都是~1的，如此接近真实值。

为了确保这可以作为一个例子工作的，

建议通过设置一个模式waveform\_kwargs=dict(mode=[（2,2,0）]（您可以使用任何模式）。

使用任何你喜欢的方法来决定1个标准差的宽度你可以使用plotting方法来将其可视化。

我们只是估计它现在的值而已

任务5：分析GR的修改或环境影响

注释：GR为广义相对论的简写

现在我们将着眼于few的一个非常有用的方面:通过向轨迹中添加条款来测试超越GR或环境影响。

在这里，我们将使用一个修改过的轨迹，并且我们已经把它放在一起了。

它的修改是pdot =pdot \*(1+mod)和edot =edot \*(1+mod)。

它在下面被导入（ModifiedPnTrajectory）。

它是史瓦西偏心极限中的一个基本的PN轨迹。

这是有效地从few教程（添加GR修改部分）。

对于振幅模块，使用Interp2DAmplitude,并且对于总和使用InterpolateModeSumHint。

提示:您创建了一个继承SchwarzschildEccentricWaveformBase的类对象，

并用适当的模块初始化它。请参阅教程。它就是这样做的。

文件教程：

SchwarzschildEccentricWaveformBase

Make waveform Tutorial

创建要启动的波形发生器类。看看few的教程。它有一个精确的例子来说明如何做到这一点。

现在使用EMRItdiwaveform来创建该波形的完整TDI版本。

提示：为EMRITDIWaveform使用emri\_waveform\_args关键字参数

。另外，只使用A通道和E通道。提供response\_kwargs关键字参数来指定它。

构建注入波形并将其放置在DataResidualArray中。

然后，用灵敏度矩阵、数据和我们之前所做的信号发生器填充一个AnalysisContainer。

提示：修改用参数将作为最后一个参数添加。

教程3：用Eryn固定维度的MCMC

在第三个教程中，我们将学习如何通过Eryn使用MCMC进行数据分析调查。

在本教程中，我们将坚持使用简单的例子和简单的信号(就像在教程1中一样)。

在以后的教程中，我们将使用真正的引力波信号。

任务1：构建你自己的基本MCMC

为了更好地理解MCMC和Eryn，我们将首先为一个简单的问题构建我们自己的MCMC算法。

我们将使用以（μ=0）为中心的单维高斯似然和单位标准差（sigma=1）和统一先验来实现这一点。

现在画出这条链。你会注意到开始需要一个“燃烧”阶段。

所以计算后验时，必须去掉这个。有很多方法可以计算多少燃烧是合适的，但我们不会在这里讨论。

绘制由MCMC生成的后验分布，并将其与上面的真实似然图进行比较。

在构建直方图时，记住设置density=True。

如果我们后退一步，我们就会意识到MCMC实际上只是从分布中提取样本的一种方法。

当发行版像本示例那样简单时，它通常已经可用，而不需要运行MCMC（就像在scipy中一样）。

然而，在我们的工作领域中，似然分布往往非常难以处理，这就是为什么MCMC对我们如此有用。

为了说明这一点，我们将从Numpy（np.random.randn）中可用的相同分布中抽取样本）。

将你的输出链与注入的高斯分布进行比较。它们应该匹配。确

1.3任务3:并行退火

现在，我们将添加平行回火。

为了检验附加回火的效果，我们将研究一个具有两个具有不同权重的高斯峰的一维分布。

我们为这个练习提供了对数似然函数。你只需设置x的限制来适当地规范化分布。

并行回退火的工作原理是通过抑制对数似然值，与对数之前：1/T\*logL+logp相比。

这种效应降低了峰值，使高温链更容易穿过似然表面的低似然部分。

这有助于正确地采样具有多个后验模式的分布。

我们将从取样开始，而不缓和，看看效果如何。然后我们将添加回火来看到改进。

从一个非退火的取样器开始。运行它，并根据注入的可能性绘制直方图。

现在这里的情节应该非常匹配。你能描述一下所发生的变化吗？

1.4任务6:添加引力波 !

现在，我们将以我们在第一个教程中使用的简单正弦符号的形式添加引力波信号。

我们将从波形函数开始。设置DataResidualArray（注入任何您想要的任何参数）、

SensitivityMatrix和AnalysisContainer。

记住，我们对这个信号没有响应，所以灵敏度曲线应该是LISASens。

我们将为您添加波形。

生成注入数据，将其输入到数据剩余阵列，加载灵敏度矩阵，并将所有内容存储在一个分析容器中，

包括正弦波形发生器。

计算注入量的信噪比。(AnalysisContainer.calculate\_signal\_snr)

为这三个参数构建先验分布。

生成开始状态并运行采样器。

如果您想，您可以检查链和/或似然值，但您可能需要运行更长的时间才能获得任何合理的结果。

1.4.1问题

我们可以做什么样的事情来加快这个计算速度？将其转换为波形环境。

1.5任务4：计算尺寸

利用Eryn的热力学积分计算单峰高斯分布的尺寸。用10个步行器和50个温度开始。

您可以稍后使用这些数字并观察它们对测量的影响。在运行run\_mcmc时使用burn kwarg。

重要提示：对于集成采样器的tempering\_kwargs关键字参数，必须添加到字典：stot\_适应=burn。这将适应燃烧期间的温度，然后在记录样品时保持它们固定。

有用的文档：

Thermodynamics\_intedration\_log\_evidence

初始化采样器。运行MCMC。

1)计算在每个温度链中所有样本和步行器的平均可能性。

2)从采样器后端获得逆温度(betas)，并确保它们在整个运行过程中是相同的。

3)计算迹象。

有用的文件：

thermodynamic\_integration\_log\_evidence

Backend.get\_log\_like

Backend.get\_betas

1.6任务5：模型选择

对于尺寸的热力学积分对高斯脉冲和柯西脉冲进行模型比较。

这是脉冲函数和可能性。我们将假设每个脉冲都有一个振幅和平均参数。

我们把标准差设为1。

我们需要为每个模型运行单独的采样器对象。

这两个模型的先验是相同的。先验振幅应跨越注入值。平均先验应该跨越时间域。

为每个采样器初始化先验，为每个采样器初始化起点，并使用我们上面使用的设置运行两个采样器。

对这两个模型重复上面的计算。然后找到贝叶斯因子。

问题

如果我们改变了噪音，这会对我们的研究结果有什么影响呢？

教程4:MBHBs和mcmc

在第四个教程中，我们将研究MBHB波形。

我们将研究如何生成波形，添加LISA响应，计算可能性，然后我们将运行一个使用MBHBs的MCMC

任务1：生成IMRPhenomHM的波形

我们将从生成MBHBs的IMRPhenomHM波形开始。选择合理的参数，建立一个波形，

并将其绘制在特征应变表示的LISA灵敏度曲线(LISASens)上。

提示：设置f\_ref =0.0。您可以在波形生成后访问作为类属性的信息。这可能会更新。

有用的文档：

PhenomHMAmpPhase

任务5：加入速度的异构化

这将需要很长时间才能完成一次采样运行。一种加速波形的技术称为异调谐。包装使用参数子集的异质化可能性。您也可以使用Eryn转换函数。

有用的文档:

HeterodynedLikelihood

教程5:RJMCMC与Eryn

在第五篇教程中，我们将了解在底层模型发生变化时如何运行Eryn。

这意味着我们将使用可逆跳跃MCMC，也称为跨维MCMC。

在本教程中，只有2个任务，因为RJMCMC很复杂，并且将RJMCMC运行在一起需要花费时间和精力。

任务1：有多少个高斯脉冲？

我们将研究的第一个RJ问题是确定在一组有噪声的数据中存在多少个二维高斯脉冲。

这是一个你可以在更高级的Eryn教程中找到的例子。

为了在我们所安排的时间内保持这个简单性，我们将使用一个基于先验分布的建议（这是默认的）。

如果您在集成采样器的设置中设置了rj\_moves=True，它将自动从前面生成RJ提案。

但是，在使用多个分支（模型类型）时必须小心，因为默认情况下，

此建议总是建议将采样器中的每个模型更改1个模型计数，即添加或删除。

但是，在使用多个分支（模型类型）时必须小心，因为默认情况下，

此建议总是建议将采样器中的每个模型更改1个模型计数，即添加或删除。

有用的文档:

EnsembleSampler

State

uniform\_dist

ProbDistContainer

GaussianMove

DistributionGenerateRJ

我们将为您做初始设置。这里我们将设置二维网格的x和y维数。

现在，我们将为每组脉冲绘制初始参数。每个脉冲都有一个振幅、x和y值。

我们还填充了高斯函数和似然值，以避免任何问题。接下来的几个cells给出了注入数据的一个概念。

现在我们完成了绝大部分的初始设置。我们现在需要建立我们的抽样运行。

让我们从选择设置和构建基本需求开始：

`ndims`, `nleaves\_max`, `nleaves\_min`, `branch\_names`, `ntemps`, `nwalkers`

现在，我们将实例化集成采样器。tempering\_kwargs=字典（模板=目录）。

对于退火，请设置

在准备好采样器后，我们只需要生成我们的起点。这里也有很多选择。在RJMCMC中，由于模型的不确定性，它更加复杂。对于这个简化的例子，我们首先从前面抽取每个步行者1高斯分布。随着时间的推移，它会增加更多。你的目标是制作一个形状字典（ntemps，nleaves\_max，ndim）。您可以采样该和弦数组中的每个源，因为我们将用inds字典指导每个步行器只有一个叶子。您可以对该弦数组中的每个源进行采样，因为我们将通过inds字典指导每个步行者只有一个叶子。ind应该是一个形状的布尔数组（ntemps、nleaves\_max、nleaves\_max）。然后对此进行索引，并为每个步行器设置一个值为True。将这些字典填充到一个状态对象中。

现在运行采样器。我们建议好好燃烧。

数据中峰值的个数

问题:

这是你所期望的吗？RJMCMC是否会告诉我们“正确”的答案是什么？

你认为噪音是如何影响这个后验图的？

如果我们增加了噪音，我们预计会发生什么类型的变化，为什么会发生变化？

现在我们来看看恢复脉冲的参数，看看我们对注入脉冲的定位如何。

为了做到这一点，我们将把所有叶片计数中的所有样本变平，found，

i.e.we将取冷链中的每一个来源，并绘制所有这些覆盖物。提示：在这里充分使用inds数组。

我们建议使用熊猫和链消费者来做，但你可以做什么你想做什么。

问题：

我们做得怎么样?它成功了吗?会有改进吗?

任务2：使用RJMCMC选择模型

在此任务中，您将使用RJMCMC来进行直接的模型比较。这将是在教程3中使用的比较高斯脉冲和柯西脉冲的相同的模型比较。在教程3中，我们使用固定维的MCMC和热力学积分来估计每个模型的证据。然后我们比较它们，得到贝叶斯因子。使用RJMCMC将在模型选择指数上产生一个后验分布，而不是为每个模型提供直接证据。步行者中强调一种模式而不是另一种模式的比例代表了优势比。步行者中强调一种模式而不是另一种模式的比例代表了优势比。

当一个模型比另一个模型更受欢迎时，这就不是严格正确的。然而，当模型的有利度接近时，这种直接计算是可以的。

为简单起见，我们考虑信号的扩散为1。因此，我们只拟合和比较脉冲的振幅和平均值。

再次，我们将首先提供高斯脉冲、柯西脉冲和似然函数的函数。

我们将注入高斯脉冲并与柯西脉冲进行比较。

现在我们将把这两个模型的先验放在一起。这两种模型的先验都是相同的。

之前的振幅应跨越注入值。平均先验应该跨越时间域。

现在，我们将生产与我们之前建立的相同类型的高斯移动。

我们需要确保为每个模型提供一个协方差！因此，它必须是一个包含两个模型条目的字典。

但条目也可以是一样的。

实例化EnsembleSampler。

生成起点。注意：这一点非常重要。这个测试只在以下情况下有效：\*我们有两个模型，\*每个模型的最大叶为1，最小叶为0，\*并且起点必须都有高斯或柯西脉冲，不是两者都有，也不是两者都没有。

在这个设置中，每个提案都将建议切换模型，并从之前的模型中提取模型。

生成起点并使用indeds数组告诉采样器每个步行者只有一个模型实例。

使用老化器运行采样器。

问题：

如果我们改变了噪音，这会对我们的研究结果有什么影响呢？

当一个模型非常受青睐，从而使分数为1时，会发生什么？

这是对优势比的正确估计吗？

在这种情况下，如果初始优势比为1，我们还能做些什么来实际估计优势比呢？

教程6:银河双星&RJMCMC

在第六个教程中，我们将研究银河系的双星波形。

然后我们将在固定维的MCMC和RJMCMC中使用它们。

我们使用RJMCMC对数据中的源的数量进行模型选择。

任务1：使用GBGPU构建和绘制一个半乳糖二元波形

我们将从用GBGPU生成银河双星波形开始。选取合理的参数，建立一个波形，并将其绘制成与特征应变表示中的LISA A通道TDI灵敏度曲线（A1TDISens）相对应。您可以在波形生成后访问作为类属性的信息。这可能会在将来进行更新。

有用的文档：

GBGPU

任务2:在单个GB源上运行MCMC

使用选定的GB源运行一个固定维的MCMC运行。

现在修复天空位置以简化计算问题（这对于使用GBs的RJ的下一节尤为重要）。

因此，您将采样超过8个参数中的6个。

讨论或考虑这些参数的合理先验，以及您将如何确定它。

为了简单起见，我们建议在这个示例设置中使用紧密（但不太紧密）的绑定的均匀分布。

在GBGPU类上有一个更快的get\_l1方法。

但是，使用完整的分析容器设置可能更容易。这将使RJ部分更直接，但实际上不是

任务3：带有GBs的RJ

我们的最后一个任务将是在几个接近的银河双星上运行RJMCMC。这里的关键组成部分是“全局”似然函数。工作是构建一个函数，从Eryn和可调长度的模板数组，在可能性计算之前求和成一个全局模板。这可能有点乏味，但对于理解这个过程非常重要。

这个问题还有另一个细微差别必须处理，才能让这一切发挥作用。在一个二进制的固定维情况下，默认的拉伸建议对每个维的尺度是不变的，因为在提出建议时没有混合维信息，¥=X；+z（X₄-X；）。我们在教程5中使用的默认高斯移动需要对协方差矩阵的反演。如果我们对上面使用的参数（A、fo、j等）进行采样，参数之间的尺度差异会导致矩阵反演和乘法的数值问题

有各种各样的方法来处理这个问题。以下是两种可能性：

1)你可以对数尺度和减少每个参数，使它们都是1阶。这将涉及改变先验，并确保在可能性函数中包含此转换。您可以使用erryn.utils。如果转换容器，可以执行此转换。

2)您可以创建自己的提案，其中假设一个对角线协方差，并自己生成信息。我为这个例子选择了这条路线。提示：Eryn教程就有一个例子。

如果你真的想喜欢：

在描述的设置中，当前具有ind=True的每个源将一起移动。这意味着，如果一个给定的步行器有5个源，所有5个源的参数将同时发生变化。这可能会损害这些举措的接受度。实际上，您可能希望使用吉布斯抽样来同时抽样一个或几个来源。您可以使用gibbs\_采样设置kwarg来实现这一点。移动。

有用的文档:

MHMove

TransformContainer

Movement

如果您可以运行采样器并确认可能性正在工作，则认为这已完成。

分配给教程的时间和正确运行RJ设置所需的总体设置需要更多的运行时才能获得合理的结果。

所以，你可以画出结果，但它会随着采样器运行时间的延长而变得更准确。