方法细节

Metatranscriptome获取

在2020年1月，我们从IMG/M检索了总共5,150个公开可用的预组装metatranscriptomes。这些metatranscriptomes主要使用MEGAHIT（Li等人，2016年）进行组装（有关不同样本使用的组装器信息，请参见表S4，并在可能的情况下，参考原始发表样本的研究）。

初步过滤过程

为了方便，我们在表S6 - 发现管道搜索和过滤阈值中总结了初步和次级过滤过程的最终工具和截止值。我们从IMG/M门户获取的contigs的初始标准是丢弃长度小于1,000 nt或编码rRNA基因的序列（剩余的contigs通过mmseqs easy-linclust在99%序列同一性下进行去重）（Steinegger和Söding，2017年）。为了过滤掉高度不可能代表RNA病毒的序列，我们将获得的metatranscriptome contigs与来自1,831个宏基因组的DNA序列汇编进行了比较，这些宏基因组与5,510个metatranscriptomes共享“Study_ID”元数据属性。我们选择与5,510个metatranscriptomes在GOLD（Mukherjee等人，2021年）门户中的“Study_ID”共享的metagenomes，因为这些DNA装配将覆盖与分析的metatranscriptomes相似的栖息地范围。使用多种序列搜索工具（具体来说，MMseqs2（核酸-核酸（搜索类型3））（Hauser等人，2016年；Steinegger和Söding，2017年），DIAMOND（翻译核苷酸与IMG源DNA宏基因组预测ORFs（diamond blastx））（Buchfink等人，2015年），和NCBI BLAST（核酸-核酸-blastn））（Altschul等人，1997年；Camacho等人，2009年；Johnson等人，2008年）），我们迭代地识别并排除了与“DNA组”中的序列产生可靠匹配的metatranscriptomic contigs。这个过程是基于这样的假设：RNA病毒不会出现在DNA装配中，后者将由细胞生物、基于DNA的移动元素和整合逆转录病毒组成。迭代搜索是这样进行的：每次迭代逐渐增加搜索灵敏度（例如，通过减少单词长度（BLASTn）和更高的灵敏度值（MMseqs2“-sensitivity”）），同时丢弃所有与“DNA组”中的序列产生可靠匹配的metatranscriptomes集合中的序列，然后使用过滤后的输出进行下一次迭代。这个过程总共重复了五次，尽管我们应该注意，最初的迭代主要是探索性的（用于粗调程序）。

次级过滤过程

为了进一步过滤contig集合，我们补充了来自参考数据库的5,954个RNA病毒序列，并使用公共数据库（NCBI NT/NR和IMG/VR）作为DNA集合进行了额外的迭代过滤过程。为了防止排除真正的RNA病毒序列，我们屏蔽了与后续迭代中匹配参考RNA病毒的公共数据库条目。所有丢弃的contigs被聚合并补充了手动识别的DNA编码contigs，创建了一个“假阳性”数据库，用于通过排除与“假阳性”集合产生可接受匹配的序列来进一步过滤metatranscriptome数据集。收集丢弃的匹配以进一步细化工作集的过程重复了三次。

估计中间集合中的DNA残留物

为了评估过滤过程中工作集中的DNA序列残留物，我们定期分析随机contig子集，通过（1）计算RdRP到逆转录酶域的比例作为RNA病毒到DNA编码contigs的代理；（2）手动检查最频繁非RNA病毒相关域的存在。值得注意的是，在这次性能评估中，几个特定的域频繁出现，手动检查显示这些是已知重复的域。大多数情况下，这些contigs完全填充了与这些重复域的匹配，并且这些域在公共数据库中有细胞匹配，其对齐值仅略低于我们的报告或接受标准。因此，我们决定如果这些contigs完全编码多个重复域，就丢弃它们，因为这些contigs将没有足够的编码空间来编码可识别的RdRP。在下面的RdRP识别步骤（在下面的“RdRP识别”部分描述）之后，大约130个逆转录酶通过了各种过滤过程并被手动移除。在所有评估中使用的是MMseqs2、PFamA数据库（Mistry等人，2021年）和来自Wolf等人（2018年）的RdRP和RT集合。

RdRP识别

以前发布的RdRP和逆转录酶（Wolf等人，2018年，2020年）的多序列比对被格式化为特定于工具的主题数据库，并用作查询，搜索由通过上述过滤过程的contigs的6帧端到端翻译组成的序列数据库，使用PSI-BLAST、hmmsearch、DIAMOND和MMseqs2。为了估计所需的搜索截止值，我们用可能产生假匹配的非RdRP序列补充了查询集（称为“真阴性”集），构建方法如下：（1）使用大量RdRP作为查询，对PDB70数据库（2019年）进行hhsearch（来自HH-Suite），收集所有比特分数R 20且未来自RNA病毒的匹配，且至少与2个RdRP对齐；（2）获取与这些匹配的PDB条目聚类（通过ftp://resources.rcsb.org/sequence/clusters/bc-70.out）；（3）获取与这些PDB ID相关的Pfam条目，以及链接到Pfam条目的序列；（4）将高度相似的序列合并为单个代表（MMseqs2最小覆盖率：100%，最小同一性：90%）。能够产生与“真阴性”集中任何序列对齐的主题RdRP配置文件被丢弃。否则，RdRP配置文件搜索的接受标准为：配置文件覆盖率R 50%，E值% 1e-10，得分R 70。然后微调这些严格的参数，以代表非RdRP序列能够达到的最佳可能值。随后，可靠的RdRP匹配被修剪到大约核心域，我们操作性地定义为A-D基序（见下面的“RdRP催化基序A-D的识别”）。提取的RdRP核心序列被预聚类（CD-HIT，覆盖率R 75%，同一性R 90%）（Fu等人，2012年），通过all vs. all（DIAMOND BLASTp）运行，格式化为与MCL一起使用（mcxload（–stream-mirror –stream-neg-log10 -stream-tf "ceil(200)"）），聚类（MCL，膨胀值在3.6和2.8之间），对齐（MUSCLE），并格式化为如上所述的配置文件数据库（Altschul等人，1997年；Buchfink等人，2015年；Edgar，2021年；Enright等人，2002年；Steinegger和Söding，2017年）。这个过程重复了两次。随后，具有潜在RdRPs的contigs被用来从整个metatranscriptomic集合中恢复额外的contigs，这些contigs与初始搜索长度标准高度相似但较短（有关详细信息，请参见下面的“全面识别”）。在结果集合中，覆盖RdRP配置文件R 75%或具有可识别的A-D基序的序列被认为是足够完整的，可以用于下游系统发育分析。

RdRP催化基序A-D的识别

通过半手动划分先前发布的RdRP MSAs（如“RdRP识别”部分所述），构建了一个自定义基序库（可在项目Zenodo存档中找到，参见数据和代码可用性）。为了识别沿单个RdRP序列的基序，执行了与上述全长RdRP域类似的迭代搜索。

校正潜在的移码

有1,656个contigs在多个帧上具有清晰的RdRP域签名，通常由<20个核苷酸分隔（n=1,118）。为了避免将这些签名作为简单的不完整遗漏，我们以两种方式解决这些问题：（1）如果任何一个签名覆盖了主体RdRP配置文件的R75%，或者编码所需的催化基序A-C，那么将使用该签名；或者（2）通过将两个签名连接成一个单一的氨基酸序列。 通过上述RdRP识别步骤（描述在“RdRP识别”部分）后，大约有130个逆转录酶通过了各种过滤过程并被手动移除。在所有评估中使用的是MMseqs2、PFamA数据库（Mistry等人，2021年）和来自Wolf等人（2018年）的RdRP和RT集合。

RdRP识别

以前发布的RdRP和逆转录酶（Wolf等人，2018年，2020年）的多序列比对被格式化为特定于工具的主题数据库，并用作查询，搜索由通过上述过滤过程的contigs的6帧端到端翻译组成的序列数据库，使用PSI-BLAST、hmmsearch、DIAMOND和MMseqs2。为了估计所需的搜索截止值，我们用可能产生假匹配的非RdRP序列补充了查询集（称为“真阴性”集），构建方法如下：（1）使用大量RdRP作为查询，对PDB70数据库（2019年）进行hhsearch（来自HH-Suite），收集所有比特分数R 20且未来自RNA病毒的匹配，且至少与2个RdRP对齐；（2）获取与这些匹配的PDB条目聚类（通过ftp://resources.rcsb.org/sequence/clusters/bc-70.out）；（3）获取与这些PDB ID相关的Pfam条目，以及链接到Pfam条目的序列；（4）将高度相似的序列合并为单个代表（MMseqs2最小覆盖率：100%，最小同一性：90%）。能够产生与“真阴性”集中任何序列对齐的主题RdRP配置文件被丢弃。否则，RdRP配置文件搜索的接受标准为：配置文件覆盖率R 50%，E值% 1e-10，得分R 70。然后微调这些严格的参数，以代表非RdRP序列能够达到的最佳可能值。随后，可靠的RdRP匹配被修剪到大约核心域，我们操作性地定义为A-D基序（见下面的“RdRP催化基序A-D的识别”）。提取的RdRP核心序列被预聚类（CD-HIT，覆盖率R 75%，同一性R 90%）（Fu等人，2012年），通过all vs. all（DIAMOND BLASTp）运行，格式化为与MCL一起使用（mcxload（–stream-mirror –stream-neg-log10 -stream-tf "ceil(200)"）），聚类（MCL，膨胀值在3.6和2.8之间），对齐（MUSCLE），并格式化为如上所述的配

置文件数据库（Altschul等人，1997年；Buchfink等人，2015年；Edgar，2021年；Enright等人，2002年；Steinegger和Söding，2017年）。这个过程重复了两次。随后，具有潜在RdRPs的contigs被用来从整个metatranscriptomic集合中恢复额外的contigs，这些contigs与初始搜索长度标准高度相似但较短（有关详细信息，请参见下面的"全面识别"）。在结果集合中，覆盖RdRP配置文件R 75%或具有可识别的A-D基序的序列被认为是足够完整的，可以用于下游系统发育分析。

RdRP催化基序A-D的识别

通过半手动划分先前发布的RdRP MSAs（如"RdRP识别"部分所述），构建了一个自定义基序库（可在项目Zenodo存档中找到，参见数据和代码可用性）。为了识别沿单个RdRP序列的基序，执行了与上述全长RdRP域类似的迭代搜索。

校正潜在的移码

有1,656个contigs在多个帧上具有清晰的RdRP域签名，通常由<20个核苷酸分隔（n=1,118）。为了避免将这些签名作为简单的不完整遗漏，我们以两种方式解决这些问题：（1）如果任何一个签名覆盖了主体RdRP配置文件的R75%，或者编码所需的催化基序A-C，那么将使用该签名；或者（2）通过将两个签名连接成一个单一的氨基酸序列。

contig集合的增强与已发布的基因组

为了评估我们在新预测的病毒基因组的数量和多样性方面的发现的新颖性，并为了避免排除可能在环境metatranscriptomes中代表性不足的已建立的病毒谱系，我们聚合并编制了一个名为"参考集"的"先前发布"的病毒基因组集合。这些包括在NCBI的NT数据库中识别的携带RdRP的序列（NCBI资源协调员，2018年），以及在公共数据库中未索引（在撰写本文时）的序列，这些序列在几个大规模和值得注意的RNA病毒调查和转录组图谱中被识别。我们添加这些补充序列的标准要求它们来自同行评审的出版物，并且所有底层序列完全公开可用，没有限制。NCBI NT序列是通过类似于上述描述的RdRP扫描过程（见RdRP识别）识别的。先前发布的集合是由Callanan等人（2020年）描述的Leviviricetes的广泛集合，"Yangshan-assemblage"和其他由Wolf等人（2020年）描述的集合，以及Lauber等人（2019年）描述的拟Plastroviruses组，以及在基因海洋图谱（Carradec等人，2018年；Salazar等人，2019年）中识别的几个RdRPs。在它们的聚合之后，这些序列经历了与本文中识别的metatranscriptomic序列类似的程序（即长度过滤、聚类和RdRP核心域提取）。最终的序列集合被标记为"已知"（即非新颖），并在本工作生成的数据中注明（例如，图1中的分支颜色）。处理过的"补充序列集合"被合并到主序列集合中（即本文中识别的那些），并且组合集合（称为"VR1507"）被用于所有下游分析（系统发育重建、域分析等）。

在metatranscriptomes中全面识别RNA病毒contigs

由于metatranscriptome组装通常会产生不完整的基因组，这些基因组不会满足从头RdRP检测的标准（见上文），我们使用"VR1507"contig集合（见上文），从未经聚类、未经过滤（长度、DNA相似性、RdRP存在）的"bulk-set"metatranscriptomic contigs中进行了二次"扫描"，以寻找额外的RNA病毒contigs。为此，我们使用"VR1507"作为诱饵，在"bulk-set"中寻找高度相似的contigs，使用非敏感的mmseqs搜索（mmseqs search –search-type 3 –min-aln-len 120 –min-seq-id 0.66 -s 1 -c 0.85 –cov-mode 1），然后严格过滤恢复的匹配（E值<1e-9，同一性>95%，目标覆盖率R 95%）。这些标准被选为质量保证措施，以便恢复的contigs主要包含在"VR1507"contig对应物中（这个大型扩展数据集可在项目的Zenodo存储库中找到，参见数据和代码可用性）。这个包裹标准被添加以避免捕获跨越"VR1507"查询的嵌合或其他不确定的核酸区域。过滤后的bulk contig集合与"VR1507"合并，由2,658,344个contigs组成（称为"Add1507"）。为了确保这个过程在避免捕获假阳性方面足够严格，我们验证了如果在不包含RNA病毒的DNA metagenome上执行它，不会捕获任何contigs。为此，我们使用了最近发布的高质量牛（Rumen）DNA metagenome（即长读，HiFi组装）（Bickhart等人，2022年），选择它因为它没有包含在用于发现管道的初步和次级过滤步骤中使用的DNA序列集中，使其成为一个可靠的基准。在这个搜索中，没有contig通过了我们95%同一性的对齐阈值（一个contig产生了72% ID的短对齐）。

系统发育重建

我们通过在包含完整或近乎完整RdRP的序列子集上执行初步的MMseqs2聚类运行（见表S6，工作表"聚类信息），选择了一组多样化的代表性RdRP进行系统发育分析。这些代表被称为RCR90，并经历了多次聚类（MMseqs2，序列同一性阈值为0.5）、对齐（MUSCLE5）（Edgar，2021年）和配置文件-配置文件比较（HHsearch）（Steinegger等人，2019年），如下面所述。"置换"的RdRP（具有转置的基序C，遵循C-A-B-D配置）被识别并"去置换"（即，包含基序C的环从序列中切下并重新插入到基序B下游）。一旦所有识别到的具有转置基序的序列被带入规范的A-B-C-D配置，以下程序被用来产生一个包含所有RCR90集合的多序列比对：

1. 使用MMseqs2
2. 进行聚类，序列同一性阈值为0.3；使用MUSCLE5对聚类结果中的4,514个聚类进行对齐；使用HHSEARCH对聚类对齐进行配置文件-配置文件比较，生成一个4,514x4,514的距离矩阵（距离估计为dAB = -ln(SAB/min(SAA, SBB))，其中SAB是配置文件A和B的HHSEARCH得分）；使用R函数hclust从距离矩阵生成最大连通树；
3. 将树在深度阈值1.5处剪切，产生1,360个子树；
4. 使用每个子树作为指南，使用HHALIGN对相应配置文件进行层次对齐，产生1,360个对齐；
5. 从这些对齐中提取1,360个共识序列（排除超过2/3的间隔字符的位点），并使用MUSCLE5进行对齐；
6. 将共识序列的每个位置扩展到原始对齐的相应列，产生一个包含77,510个RdRp的对齐（原始RdRp序列被减少到与其局部共识相匹配的一组位置）；
7. 从这个对齐中移除超过90%的间隔字符的位点；使用HHALIGN将这个对齐与十个RT（五个II组内含子序列和五个非LTR逆转录转座子序列）的对齐进行比对。

   RdRp和RT的对齐用于使用FastTree（V.2.1.4 SSE3，Price等人，2010）程序重建近似最大似然树（WAG进化模型，伽马分布的位点速率），并在RT和RdRp之间进行根化。

   类群的分类归属

   通过将VR1507序列集映射到分析时最新的ICTV数据（2021年7月20日发布的Virus Metadata Repository (VMR)文件，对应于MSL36，可在https://talk.ictvonline.org/taxonomy/vmr/m/vmr-file-repository/13175找到）进行MEGA-BLAST（E值<1e-30，查询覆盖率R 95%，目标覆盖率R 95%，对齐长度>200，同一性R 98%，(Alignment_length)/Query_length > 0.95）来识别具有现有分类信息的树叶。总共映射了2,765个contigs，并将ICTV分类信息克隆到VR1507查询中，基于最高得分。对于VR1507 contigs的其余部分，我们使用NCBI的NR数据库进行了类似的程序（这增加了额外的6,878个映射contigs，尽管其中相当一部分缺乏分类信息或匹配已废除的分类单元）。建立树内部节点（即类群）的分类归属的程序依赖于上述参考树叶的分类归属，以及两个原则：

   1. 所有从参考叶的最后一个共同祖先派生的序列，分配给分类单元T，也属于分类单元T；从更深的树节点派生的序列，不属于分类单元T，因此，应分配给同一等级的新分类单元（分类群）；
   2. 树类群分裂为给定等级的分类单元的深度，由同一等级的现有分类单元定义，并且是局部依赖的（例如，不同门的家族的特征深度可能不同）；

   应用这些原则的前提是现有分类学与树不矛盾，即分配给分类单元的参考序列形成单系群，在同一等级内不重叠且不嵌套（例如，家族类群不能嵌套在另一个家族内）。对参考叶的分类归属的检查显示，这个假设虽然通常满足，但在多个地方被违反。这需要首先解决冲突关系。为此，对给定等级的所有分类单元（即分别对门、类等）应用了以下程序：

   1. 修剪树，只包含定义该等级的叶；从修剪后的树中导出叶权重（w）；
   2. 对于树中的每个分类单元T，计算该分类单元中叶的总权重（WT = Swi跨越分配给T的叶）；
   3. 对于树中的任何树类群C，计算该类群中叶的总权重（WC = Swi跨越属于C的叶）；

4. 对于类群C和分类单元T的每个组合，计算类群-分类单元权重（WCT = Swi跨越属于C且分配给T的叶）；然后可以计算类似精确度和召回率的措施（PCT = WCT / WC和RCT = WCT / WT），并组合成质量指数QCT = PCT * RCT。
5. 对于树中的每个分类单元T，确定CT = argmax QCT作为T的"本地"位置（类群，其中T的最大权重集中在最小侵入其他分类单元的地方）；属于CT类群但未分配给T的叶，以及分配给T但不属于CT类群的叶，分别标记为"侵入"或"外层"；
6. 检查并解决所有与树不兼容的分类学分配。在大多数情况下，解决冲突的最不可知方法是剥离相应叶的分类标签。在一个案例中，Lenarviricota的Timlovirales秩序中的大多数家族被发现嵌套在一个非常深分支的Blumeviridae家族内。为了本工作的目的，我们在没有冲突家族分配的最大Timlovirales类群上保留了Blumeviridae标签，并从Timlovirales的其余部分移除了Blumeviridae标签。在其他几个案例中，当小家族完全嵌套在较大的家族中时（例如，一个单独的叶被分类为Sunviridae，嵌套在一个大型的Paramyxoviridae类群内），为了后续分析的目的，移除了嵌套家族的标签，并在事后恢复。一旦所有叶的分类标签与树兼容，就对每个分类等级的未标记叶进行以下程序，单独分配新的分类标签：
7. 为树中的所有节点分配深度，定义为从该节点到叶的最长节点到叶路径；
8. 在77,510个叶的完整树中，确定每个分类单元的最后一个共同祖先节点；记录分类单元的深度，定义为LCA节点的深度加上进入树边的长度；从分类单元LCA派生的未标记叶被分配给该分类单元；
9. 隔离所有现有分类单元之外的类群；对于每个这样的类群，确定所有现有姐妹分类单元的深度；如果一个类群只有一个姐妹分类单元，将搜索最接近的亲属扩展到根部，直到至少识别到另一个相关分类单元；计算阈值深度为相关分类单元集合的平均值；
10. 在阈值深度处解剖现有分类单元之外的类群；将每个结果（子）类群分配给给定等级的新分类单元；
11. 具有单个现有分类单元作为姐妹的新分类单元被标记为与该分类单元相关联。

新的分类单元被赋予名称，表示等级（即用p、c、o、f和g分别表示门、类、目、科和属），后面跟着新分类单元的序数，可选地，用标签终止于与先前描述的分类单元相关联的分类单元（例如，f.0127.base-Noda是RdRP树中位于Nodaviridae基部的第127个新家族）。

为了评估深度系统发育重建的稳健性，执行了以下程序：

1. 收集了201个至少有20个RCR90序列的家族列表；
2. 对每个家族和RT集合随机抽取一个代表；
3. 从主对齐中提取202个样本的子对齐；
4. 使用IQ-Tree程序（Nguyen等人，2015）重建系统发育树，自动选择最佳拟合模型，分析100个独立样本：

首先，为已知的五个门识别每个门的最高质量指数（QI，如上文"类群分类归属"部分所述）的类群；使用质量指数值作为门单系性的度量。记录涉及破坏各自门单系性的家族（注意，一个叶既可以是其自身门的异常值，也可以是另一个门的侵入者）。其次，将子树折叠到门级别；排除15个（100个中的）具有并系门的树（例如，Pisuviricota的最高质量类群嵌套在Kitrinoviricota的最高质量类群中）。使用IQTree程序构建剩余85个（大部分）单系门的扩展多数规则共识树；将分支支持值乘以0.85（这些树在整体样本中的比例）。

将单个contigs分配给RCR90聚类：一旦上述RCR90巨型树的新区域被主要分类等级（门/属）完全填充，我们继续将来自较大的VR1507集合（见上文 - contig集合）的contigs进行分类。contig分类是通过以下4个级别逐步进行的：级别A。是用于创建树的contigs编码RdRPs。级别B。包括与级别A的RdRPs具有异常高氨基酸同一性的contigs（通过最佳BLASTp匹配，同一性R9 0%，查询覆盖率R75%，E值<1e-3）。级别C。由与级别{A, B}的contigs相同的RvANI90聚类中的contigs组成，级别D。由与级别{A - C}的contigs具有高核苷酸相似性的contigs组成（通过最佳dc-MEGABLAST命中，

同一性R90%，查询覆盖率R75% OR Nident R 900nt和E值<1e-3）。基于上述级别中ICTV标记的RdRPs的分布，我们估计通过这种方式分类的大多数contigs将在属级别大致共享相同的分类等级。值得注意的是，对于级别C。，我们设计了一个自定义测量单位RvANI，这是标准平均核苷酸同一性（ANI）聚类的扩展，旨在适应metatranscriptomic装配的片段化特性，从而避免由于相关序列的相对较低配对覆盖率而导致的新颖性高估。简而言之，RvANI的计算如下：最初，使用mmseqs计算contig集合中的所有成对序列比对，然后用于传统的ANI和比对分数（AF）计算，其中：

ANI = (%ID * 比对长度) / (contig m的长度 * contig n的长度)
AF = Min（contig m的比对覆盖率；contig n的比对覆盖率）

给定所有ANI和AF对（对于原核生物，95-96%的ANI通常被接受为物种边界，对于某些病毒也有类似的粒度定义），聚类被定义为在核苷酸相似性图中修剪为ANI R90%和AF R90%的连通分量。RvANI通过重新插入特定的成对比对到修剪过的核苷酸相似性图中来纠正metatranscriptomes中的不均匀基因组覆盖率，即使它们的AF低于所需截止值，只要底层的成对比对满足这些标准：%ID R 99，比对长度R 150 [bp]，并且比对发生在每个contig的末端，即比对覆盖了每个contig的5'或3'末端。随后，我们定义RvANI90聚类为使用R-igraph包处理的核苷酸相似性图中的不同连通分量（Csardi和Nepusz，2006）。

识别可靠的CRISPR间隔序列

将非冗余RNA病毒序列与IMG数据库中细菌和古菌CRISPR间隔序列（Chen等人，2021）进行比较，以（i）确定可能感染原核宿主的病毒，以及（ii）可能预测这些病毒的特定宿主分类群。首先，使用blastn v2.9.0（选项"-dust no -word_size 7"）将非冗余RNA病毒序列与IMG数据库中细菌和古菌全基因组预测的1,568,535个CRISPR间隔序列进行比较。为了最小化由于低复杂度和/或重复序列导致的假阳性命中数量，如果CRISPR间隔序列（i）编码在包含2个或更少间隔的预测CRISPR阵列中，（ii）长度为20bp或更短，或（iii）包含低复杂度或重复序列（由dustmasker（v1.0.0）（Morgulis等人，2006）（选项"-window 20 -level 10"）或etandem（v6.6.0.0）（Rice等人，2000）（选项"-minrepeat 4 -maxrepeat 15 -threshold 2"））检测到的直接重复R 4bp，或（iv）包含在dustmasker或etandem检测到的重复序列中，则从分析中排除这些CRISPR间隔序列。为了将RNA病毒与CRISPR间隔序列联系起来，只考虑在整个间隔长度上有0或1个错配的blastn命中。进一步检查间隔和阵列，以检查（i）阵列中的间隔是否长度一致，以及（ii）预测宿主基因组中是否发现Cas和/或RT基因，如果是，这些基因是否与命中的CRISPR阵列相邻。为了将CRISPR链接搜索扩展到可用草图基因组的细菌和古菌之外，我们接下来使用相同的方法将非冗余RNA病毒序列与IMG数据库中预测的53,372,161个CRISPR间隔序列进行比较。使用与上述基因组衍生的CRISPR阵列相同的方法过滤掉虚假间隔序列（见上文），仅保留RNA病毒和CRISPR间隔序列来自同一生态系统（如GOLD数据库定义）的命中。由于CRISPR阵列通常组装在没有其他基因的短contigs上，我们使用阵列的重复序列将它们与预测宿主联系起来。具有至少1个命中的metagenome衍生的CRISPR阵列的重复序列与IMG细菌和古菌基因组进行blastn（v2.9.0）比较（选项"-perc_identity 90 -dust no -word_size 7"）。然后检查这些命中在预测宿主基因组中的位置，以确定CRISPR间隔阵列、Cas基因和RT基因的存在。当单个RNA病毒序列或间隔序列与多个宿主基因组潜在相关联时，根据以下标准进行优先排序：（i）间隔阵列位于编码RT的CRISPR阵列旁边，（ii）在基因组的其他位置识别到编码RT的CRISPR阵列，（iii）间隔阵列位于编码Type III CRISPR阵列旁边，（iv）在基因组的其他位置识别到Type III CRISPR阵列，（v）在基因组中识别到另一种类型的CRISPR阵列，以及（vi）在基因组中无法识别Cas基因。进一步研究了Mushroom Spring中Roseiflexus sp. RS-1编码的CRISPR阵列的间隔内容。首先，使用专用工具Crass v1.0.1（Skennerton等人，2013）以默认参数（Skennerton等人，2013）特别组装了来自Mushroom Spring微生物垫的17个metagenomes的CRISPR阵列。接下来，识别基于与Roseiflexus sp. RS-1（表S3）已知CRISPR阵列相对应的重复的阵列，并收集和过滤相应的间隔，如前所述。使用blastn（v2.9.0）将IMG/VR v3数据库（Roux等人，2021）中的RNA病毒序列和DNA病毒序列与这个Roseiflexus sp. RS-1间隔阵列数据库进行比较（选项"-dust no -word_size 7"）。基于对R 1 RS-1间隔的命中（整个间隔长度的% 1错配）首先识别感染Roseiflexus sp. RS-1的候选RNA噬菌体。对于这些选定的噬菌体，收集整个间隔长度上有多达4个错配的命中，以检测更遥远的病毒-间隔命中。基于以下3个标准识别Roseiflexus sp. RS-1 clade genPartiti.0019病毒的候选衣壳片段：间隔匹配RNA靶向CRISPR阵列，没有相应的DNA序

列，以及在metatranscriptome时间序列中与R 1 RdRP contig的高覆盖率相关。首先，使用类似于Crass组装间隔的blastn比较（blastn，选项“-dust no -word_size 7”，允许% 1错配）来识别候选衣壳编码contigs，即在同一metatranscriptomes中，排除所有编码RdRP或CRISPR阵列的contigs（n=3,958）。接下来，将具有R 1间隔匹配的候选者与Mushroom Spring DNA metagenomes的所有contigs进行比较（blastn（v2.9.0），选项“-task megablast -max_target_seqs 500 -perc_identity 90”），并排除所有具有匹配DNA contig（R 90%同一性）的候选者（n=3,650）。最后，使用bbmap.sh（v.38.90）获取所有genPartiti.0019 RdRP contigs和所有候选衣壳片段的覆盖率（选项“vslow minid=0 indelfilter=2 inslenfilter=3 dellenfilter=3”），并保留在42个Mushroom Spring metatranscriptomes中具有R 0.9 Pearson相关性的候选者作为可能的衣壳片段（n=88）。为了评估这些衣壳片段的基因内容，使用Prodigal（v2.6.3）（Hyatt等人，2010）（选项“-p meta”）进行de novo cds预测，并使用标准blast-mcl流程（blastp（v2.9.0），默认选项，基于得分R 50选择命中，MCL聚类（v.14-137）膨胀值2）进行聚类。对于三个最大的蛋白质聚类，使用MAFFT v7.407（Katoh和Standley，2013）构建序列比对，并将其用作hhsearch的输入，针对病毒聚焦的uniprot公共数据库（uniprot_sprot_vir70），以及由已知 分子病毒和picobirnaviruses的衣壳蛋白制成的自定义数据库（可在项目的Zenodo存储库中找到，参见数据和代码可用性“Partiti_Picob_CP.tar.gz”和PC1_PROMALS3D_new.hhr）。

栖息地分布和相对丰度估计

为了可视化目的，从IMG和GOLD数据库获取了每个metatranscriptome的位置、生态和分类信息。具体来说，从GOLD获取GPS坐标和生态系统分类，并将生态系统信息进一步分为自定义类别（表S4）。为了大致估计每个metatranscriptome中存在的宿主多样性，查询了IMG注释流程（Clum等人，2021）预测的所有contigs的分类信息，即细菌、古菌、真核生物和病毒。然后，将真核生物相关contigs的数量与原核生物相关contigs的数量的比率用作“原核生物主导”与“真核生物主导”数据集的代理。具体来说，将真核生物相关与原核生物相关contigs的比率≤0.3或≥0.7的数据集视为“原核生物主导”或“真核生物主导”，其他数据集视为“混合”。使用matplotlib v3.3.4和basemap v1.2.2 for python 3.8.5（Hunter，2007）绘制地图。对于读取映射，建立了一个去重的RNA病毒序列集（95% ANI超过95% AF，使用CheckV anicalc.py和aniclust.py脚本建立；Roux等人，2021），此后称为“NR-mapping”数据集。然后，将来自3,998个metatranscriptomes（表S4）的质量修剪读取（sensu；Clum等人，2021）映射到这个数据集。首先，使用blastn v2.9.0+（E值≤0.01）将每个metatranscriptome的contigs与NR-mapping数据集进行比较。所有累积blast命中的平均核苷酸同一性≥90%且覆盖最短序列的≥80%的contigs被视为潜在RNA病毒。从现有的IMG读取映射信息中提取所有映射到被识别为潜在RNA病毒的contigs的读取以及所有未映射的读取，并使用bbmap v38.81（Bushnell，2014）在NR-mapping数据集上进行de novo映射，选项为：“vslow minid=0 indelfilter=2 inslenfilter=3 dellenfilter=3”。这一步是为了通过排除所有映射到非病毒metatranscriptome contigs的读取来减少计算时间和假阳性映射的风险。然后使用FilterBam（https://github.com/nextgenusfs/augustus/tree/master/auxprogs/filterBam）过滤生成的bam文件，仅保留同一性≥50%和覆盖率≥50%的映射，使用bedtools v2.30.0（Quinlan，2014）的genomecov计算每个样本中每个contig的平均覆盖深度。然后计算每个分类群的相对比例，即分类群成员的累积覆盖率除以该数据集中所有预测RNA病毒contigs的总累积覆盖率。

遗传密码分配和ORF调用

目前，为多样化的宏基因组数据设计的ORF识别软件仅限于标准遗传密码（11）或线粒体遗传密码（4）（当预测的ORF异常短时选择）。为了识别可能使用替代遗传密码的类群，我们提取了RdRp核心足迹，并在其中扫描了标准终止密码子。首先，将所有RdRP编码contigs分为两个子集：“标准”和“非标准”，如果在RdRP核心的狭窄坐标内出现任何规范终止密码子。然后，对“标准”集合使用Prodigal（v2.6.3）的metagenomic模式（“anonymous”）进行metaprodigal CDS预测（默认参数）（Hyatt等人，2010）。在“非标准”子集中，终止密码子使用模式在contigs之间进行聚合，并与每个树叶相关联，分为“线粒体”（使用UGA作为感码子）和“原生生物”（其他模式）。计算内部树节点的模式流行率（后代叶中的相对频率）；高流行率的类群被记录并研究。为了实际目的，使用第一个使整个RdRP核心能够翻译的遗传密码对“非标准”子集进行ORF预测。对于没有任何可用遗传密码使RdRP核心不间断翻译的情况，分配了一般的“非标准”值，并使用线粒体遗传密码（4）进行

预测。为了排除这些预测的RNA病毒通过tRNAs的活性重编码的可能性，使用tRNAscanME2（Chan等人，2021）对VR1507集合进行了单次传递，使用“global”标志（用于非特定域的tRNA预测）。在任何预测使用替代遗传密码的病毒contigs上都没有识别到tRNAs，这表明这些很可能是适应宿主而不是病毒-宿主军备竞赛的元素，如在某些dsDNA噬菌体中所见（Ivanova等人，2014）。

RBS识别和量化

使用VR1507作为输入，按照Schulz等人（2020）的描述执行RBS量化。简要地说，如上所述运行Prodigal（v2.6.3）（Hyatt等人，2010；Schulz等人，2020），然后从Prodigal的GFF输出文件中获取“rbs_motif”字段，并将不同的50 UTR序列分类为“SD”（对于类似于AGGAGG的基序，即规范的Shine-Dalgarno）、“None”和“Other”（有关详细信息，请参见数据和代码可用性，“RBS_Motif2Type.tsv”）。然后，对于每个contig，定义“%SD”为所有“SD”ORF与所有具有真实起始（即不被contigs边缘截断，字段“start_type”不同于“Edge”）的ORF之间的比率。

域注释

为了对RdRP包含contigs编码的蛋白质进行初步域注释，我们使用hmmsearch（来自HMMER V3.3.2套件）（Finn等人，2011；Wheeler和Eddy，2013）将这些蛋白质与从多个蛋白质剖面数据库收集的隐藏马尔可夫模型（HMMs）进行匹配，最大E值为0.001（PFam 34，COG 2020发布，CDD v.3.19，CATH/Gene3D v4.3，RNAVirDB2020，ECOD 2020.07.17发布，SCOPe v.1.75）（Andreeva等人，2014，2020；Cheng等人，2015；Galperin等人，2021；Lu等人，2020；Mistry等人，2021；Sillitoe等人，2021；Wolf等人，2020）。我们用具有溶菌功能的自定义剖面集合（称为“LysDB” - 可在项目的Zenodo存储库中找到，参见数据和代码可用性）补充了这组HMMs。LysDB是从（1）手动审查的公共数据库剖面条目构建的，我们可以将它们链接到与病毒诱导的细胞裂解或病毒从宿主细胞退出相关的GO术语，以及（2）“Sgl”蛋白质的自定义剖面，Chamakura等人实验证明这些蛋白质可以诱导细胞裂解（Chamakura等人，2020）。此外，我们使用InterProScan（v.5.52-86.0）扫描蛋白质序列，使用MobiDBLite（v2.0），Phobius（v.1.01），PRINTS（v.42.0），TMHMM（v.2.0c）（Attwood等人，2012；Jones等人，2014；Käll等人，2004；Käll等人，2007；Krogh等人，2001；Potenza等人，2015）。由于用于初始注释的公共蛋白质剖面数据库可能包含代表跨越多个功能域的多蛋白体的HMMs，我们开发并采用了一种程序来识别这些剖面，以便在后续注释过程中将其屏蔽。为此，我们首先使用hmmemit命令将HMMER剖面转换为多序列比对，然后将这些比对用作HH-Suite的all-versus-all剖面比较的输入。接下来，通过标记涵盖至少两个其他不重叠剖面的剖面来识别假定的多蛋白体剖面（“get_polyproteins.ipynb”脚本，请参见数据和代码可用性）。提取多蛋白体域之间的不匹配区域，创建一组保守但未知的域，称为“InterDomains”。此外，手动使用HHpred检查了超过1000个匹配状态（定义为少于50%间隙的列）的剖面。识别出的一些多蛋白体剖面被拆分为其组成域。随后，将所有hmmsearch结果聚合，并根据其分类级别（未整理的剖面或未知功能（例如“DUF”）的剖面被降低优先级）和相对对齐统计数据对剖面匹配进行优先排序。为了提高域剖 面的功能性注释质量并为未注释的剖面分配功能，我们识别了相似剖面的聚类（称为“clans”）。首先，从原始数据库中提取至少有一个命中的剖面，将其重新格式化为HH-Suite的HHMs（如上所述），并用于额外的all-versus-all步骤。然后，这个剖面比较的输出被用作基于Leiden算法（“get_clan_membership.ipynb”脚本，参见数据和代码可用性）的基于图的聚类过程的输入，该过程识别clans为高度相似域的社区。然后，clan成员身份被用来通过从功能注释的剖面转移注释来提高功能注释的覆盖率。简而言之，这个过程遵循基于共识的标签分配。例如，一个有12个剖面标记为“RdRP”和2个“未分类”剖面的clan被设置为“RdRP”clan，这2个未分类成员被重新分类为“RdRP”。冲突的情况要么保持未解决，要么选择最低的分母。例如，一个有4个“未分类”剖面，还有12个成员剖面标记为“超家族2解旋酶”和另外10个成员剖面标记为“超家族1解旋酶”的clan，被设置为“解旋酶-不确定”，并将这个标签扩展到那4个“未分类”成员。所有后续通过预定义截止值（E值% e-7，得分R 9，对齐长度R 8[AA]）的剖面匹配被用来生成一个新的自定义剖面数据库，这个过程类似于用于RdRPs的过程（见上文）。只有包含R 10个序列且共享相同功能分类的聚类被用来生成HMMs。然后，这个剖面集合被RNAVirDB2020数据库的大部分剖面以及来自其他数据库的几十个精选剖面补充（这个最终的剖面数据库称为“NVPC”，可通过项目的Zenodo存储库获取，参见数据和代码可用性）。最后，我们使用新的剖面数据库查询了330k contig集合的六帧翻译（如上所

述）。（图S3 - 注释管道）。随后，我们使用GGGenomes（https://github.com/thackl/gggenomes）为每个400+个已识别家族（新颖和已建立）的z4-20个代表性contigs生成了基因组图，然后手动检查以识别新的域以及不常见的域融合和分割。

宏基因组装配的质量控制和可靠性

宏基因组装配容易产生各种类型的伪影，可能导致装配中出现不代表原始生物样本中任何现有核酸分子的明显contigs（Arroyo Mu¨ hr等人，2020）。臭名昭著的是，嵌合体（由至少两个不同的核酸分子错误装配而成的contigs）可能是新颖性声明的重大挫折，并且很难识别和分离出真实的遗传实体。我们通过实施几项严格的程序来解决这个问题，以避免可能源于潜在嵌合体contigs分析的任何误解：

1. 首先，本研究中的声明不基于单例。相反，我们只报告基于进化保守的序列组（来自多个装配的两个或更多可比对contigs，理想情况下）或在粗略系统发育水平（家族级别及以上）保守的特征的观察。嵌合体在多个装配中重现的可能性似乎可以忽略不计。
2. 其次，当观察到意外现象时，例如基因组重排、基因裂解和基因融合，我们在读取水平上手动检查每个案例，即追踪原始测序运行并通过上述"栖息地分布和相对丰度估计"部分描述的过程将原始Illumina短读取映射到有问题的contigs，并检查沿装配contigs的读取分布，确保contigs（不仅仅是RdRP编码区域）得到了良好的覆盖。在某些部分显示出异常低覆盖率或偏斜的GC%含量的contigs被认为是不可靠的，并被丢弃。
3. 我们观察并从我们通过聚合已发布来源构建的集合中移除了几十个contigs，这些contigs很可能是嵌合体（主要是部分levivirus，部分rRNA）。受到这一观察的启发，我们使用SILVA rRNA数据库（BLASTn针对SILVA SSU和LSU Ref NR99，默认参数）（Quast等人，2013）对整个VR1507 contig集合进行了搜索，并手动检查了"域注释"部分识别的40个编码核糖体蛋白的contigs，以匹配公共数据库中的核糖体蛋白剖面（例如，核糖体蛋白L3 PF00297.24）。总的来说，我们标记了75个这种类型的潜在嵌合体（其中23个来自先前发布的来源，请参见表S6，工作表"rRNA_summary"了解详细信息）。只有这些可疑contigs的RdRPs被用于下游分析，而其余部分被忽略。
4. 我们执行的DNA减法大大降低了由RNA病毒序列和DNA编码序列（无论是rRNA还是mRNA）组成的嵌合体的丰度。显然，然而，这个过程无法消除由不同RNA病毒基因组的部分组成的嵌合体。因为这样的嵌合体很难与真正的重组病毒基因组区分开来，我们采用了一种启发式方法，使用域注释来检测具有重复全长RdRP足迹的contigs。这些被认为是嵌合体，因为RNA病毒通常编码一个（全长）RdRP。我们发现了一个这样的案例，ND_250651，一个由levivirus和cystovirus组成的嵌合体。

与最近发布的RNA病毒发现工作的定量比较

从https://datacommons.cyverse.org/browse/iplant/home/shared/iVirus/ZayedWainainaDominguez-Huerta_RNAevolution_Dec2021下载了Tara项目的44,779个RdRPs。Serratus项目的RdRPs由296,623个独特的PalmDB序列表示，这些序列从https://github.com/rcedgar/palmdb repository下载。Serratus序列代表了一个紧密定义的RdRP核心（仅包含A、B和C基序），其平均长度为107（与RCR90集合的453 aa相比）。值得注意的是，我们的研究、Tara项目和Serratus项目各自定义了RdRP的哪些区域可用于MSA和随后的系统发育分析。因此，我们将比较限制在研究之间共享且定义最低的RdRP区域，即由palmDB定义的区域。我们通过使用本研究的所有329,202个独特的RdRP序列和Tara项目的44,779个RdRPs进行BLASTP搜索（e-value 0.0001）来实现这一点，使用最佳命中来修剪查询（具体来说，对于长度为K的查询和长度为L的主体以及主体的命中足迹p1..p2和查询的命中足迹q1..q2，查询被修剪为max(1,p1-q1-1).min(K,p2+L-q2)以考虑主体的缺失部分）。没有与PalmDB显著命中的查询保持未修剪。将完整的序列集合合并在一起，并使用MMseqs2进行聚类，序列同一性阈值为0.9和0.5（–min-seq-id 0.5/0.9 -c 0.333 -e 0.1 –cov-mode 1 –cluster-mode 2）。所有序列被分类为四个类别：i）"已知"（来自GenBank和其他已发布的来源，参见本研究的"contig集合增强与已发布的基因组"），ii）"RVMT"（来自当前数据集的RNA病毒MetaTranscriptomes），iii）Serratus和iv）

Tara。检查聚类以确定每个集合的成员的存在，聚类集合交集列在表S8中。 精确阈值，包括期望值（E值），用于所有分析，这些分析源于序列搜索或比对过程（例如域预测，CRISPR间隔匹配等），在相关主要文本或方法细节中提供，并在表S6（工作表“过滤阈值”用于DNA过滤过程的E值，工作表“聚类信息”用于聚类阈值和相关量化）中提供。