

GO-index课题数据预处理技术路线图

下载 SRA 数据

进入 <http://www.ncbi.nlm.nih.gov/books/NBK158899/> 页面，点击进入 Aspera 的下载界面，下载 Aspera 软件。

Overview

When to use a command line utility rather than the SRA website.

For multiple simultaneous downloads of SRA data, or for high-volume downloads, we recommend using command line utilities such as wget, FTP, or Aspera's 'ascp' utility. As with web-based downloads, the best speed is achieved with Aspera's FASP implementation. ascp is bundled with the [Aspera Connect](#) plugin.

Downloading SRA data using the SRA Toolkit.

进入 <http://www.ncbi.nlm.nih.gov/>，点击进入 Download 页面。

Welcome to NCBI

The National Center for Biotechnology Information advances science and health by providing access to biomedical and genomic information.

[About the NCBI](#) | [Mission](#) | [Organization](#) | [NCBI News](#) | [Blog](#)

Submit

Deposit data or manuscripts into NCBI databases



Download

Transfer NCBI data to your computer



Learn

Find help documents, attend a class or watch a tutorial



Develop

Use NCBI APIs and code libraries to build applications

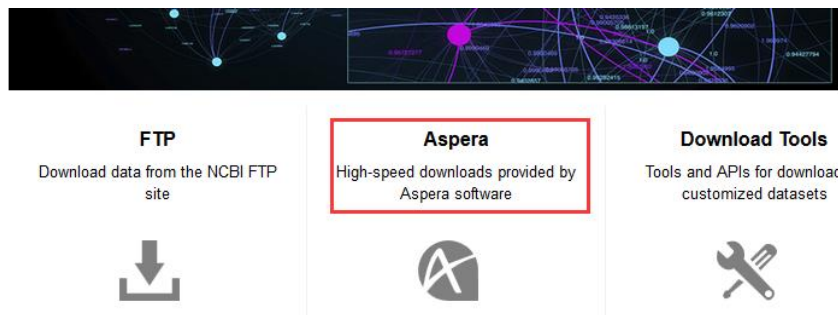
Analyze

Identify an NCBI tool for your data analysis task

Research

Explore NCBI research and collaborative projects

进入 <http://www.ncbi.nlm.nih.gov/home/download.shtml>, 点击进入 Aspera。



依次展开

- hapmap
- mmdb
- ncbi-asn1
- pathogen
- pub
- pubchem
- pubmed
- refseq
- repository
- sequin
- sky-cgh
- snp
- **sra**
- tech-reports
- toolbox
- tpa
- variation

- snp
 - sra
 - Submissions
 - development
 - doc
 - examples
 - metadata_table
 - ngs
 - ngs-tools
 - refseq
 - reports
 - sadb
 - sdk
 - **sra-instant**
 - srafuse
 - utilities
 - wgs
 - wgs_aux
- tech-reports

- sra
 - Submissions
 - development
 - doc
 - examples
 - metadata_table
 - ngs
 - ngs-tools
 - refseq
 - reports
 - sadb
 - sdk
 - sra-instant
 - analysis
 - reads
 - ByExp
 - ByRun
 - **BySample**
 - ByStudy
 - srafuse
 - utilities
 - wgs
 - wgs_aux
- tech-reports

- ✚ **reads** 文件夹下有 4 个不同的文件夹，实际上是 4 种不同的检索方式，展开后都有对应的编号，只要是同一个数据，按照不同的方式有不同的编号，链接的都是同一个数据，最终都是 **SRR** 开头的数据。例如对于编号为 **SRX1331783** 的数据，我们按照 NCBI 的 SRA 类别检索后，得到以下信息：

Runs: 1 run, 28.5M spots, 7.2G bases, [3.4Gb](#)

Run	# of Spots	# of Bases	Size	Published
SRR2601030	28,495,361	7.2G	3.4Gb	2015-10-13

- ✚ 如果通过 **SRX** 编号查找，就进入 **ByExp**，按照下图依次点开，找到后直接点击，如果安装了 **Aspera**，就可以自动唤起，然后下载。

The screenshot shows the NCBI SRA browser interface. On the left, a tree view shows the navigation path: **sra-instant** > **analysis** > **reads** > **ByExp** (highlighted with a red box) > **sra** > **DRX** > **ERX** > **SRX** (highlighted with a red box). A red arrow points to the **SRX** folder with the text "可以点击翻页!". Below the **SRX** folder, a list of SRX IDs is shown: **SRX000**, **SRX001**, **SRX002**, and **SRX1331780** through **SRX1331787**. A red box highlights **SRR2601030.sra** in the list. On the right, a list of SRX IDs is shown: **SRX1331780**, **SRX1331781**, **SRX1331782**, **SRX1331783**, **SRR2601030**, and **SRR2601030.sra** (highlighted with a red box). A red box also highlights the **SRX** folder in the left tree view.

- ✚ 按照 **SRR** 编号也是同理，都可以找到。

SRA 格式转换成 fastq 格式

PE 数据:

- `/home/emma/tools_download/sratoolkit.2.6.3-ubuntu64/bin/fastq-dump -A /path/to/*.sra --origfmt - -split-3 -O /path/to/outputdir/`

SE 数据:

- `/home/emma/tools_download/sratoolkit.2.6.3-ubuntu64/bin/fastq-dump -A /path/to/*.sra -origfmt -O /path/to/outputdir/`

参数解释:

- **-A** 下载的 sra 文件
- **--split-3** 如果下载的 SRA 文件中只有一个文件，那么这个参数就会被忽略。如果原文件中有两个文件，那么它就会把成对的文件按*_1.fastq, *_2.fastq 这样分开。如果还有出现了第三个文件，就意味着这个文件本身是未成配对的部分。可能是当初提交的时候因为事先过滤过了一下，所以有一部分数据被删除了。
- **--origfmt** 如果不加该参数，sra 转换成 fq 以后的序列 ID 带有 SRA 的标识，如 SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=72。加 **-origfmt** 参数后，得到的序列 ID 是提交时的序列 ID，如 HWI-ST758:138:C394CACXX:6:1101:1216:2026。
- **-O** 输出路径

生成标准的 fastq 格式和 fasta 格式

- （对于 PE 而言）标准的 fastq 格式，序列 ID 后面需要数值 1 和 2 表示互为 read pair。Trinity 拼接处理 PE 时需要这个

信息。Fastq 标准格式参见 https://en.wikipedia.org/wiki/FASTQ_format 的 **Illumina sequence identifiers**。

✚ 因为 blast 目前不能直接处理 fastq，需要将 fastq 转成 fasta。

✚ `perl /path/to/modified.pl /path/to/*.fastq /path/to/modified.fq /path/to/modified.fa`

Fastq 第一次 QC

✚ 使用 FastQC 做评估（业内常用 QC 工具）。

✚ `/home/emma/tools_download/sratoolkit.2.6.3-ubuntu64/bin/Fastqc/fastqc -f fastq /path/to/*_1.fq
/path/to/*_2.fq -o /path/to/outputdir`

✚ 由于 RNA-seq 自身特性，即便是经过预处理（去接头和去低质量）FastQC 的很多模块都会出现 Warn 甚至 Fail（参见 <https://www.biostars.org/p/160440/>）。

- **Per tile sequence quality**（参见 <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/12%20Per%20Tile%20Sequence%20Quality.html>，和 <https://www.biostars.org/p/170625/>）。HiSeq 测序仪物理分区。横坐标是 cycle 数目，纵坐标是 tile 的标识。这个模块 warn 或者 fail 是测序仪器的问题，没有补救方法。权衡利弊，或者要把整个 tile 的数据删除。
- **Per base sequence content**（<https://sequencing.qcfail.com/articles/positional-sequence-bias-in-random-primed-libraries/>）。RNAseq 的前 12-13bp 几乎都会出现波动，这是由于反转录时的 random priming 并非真的那么 random（参见 <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2896536/>），无法消除。同样原因会导致 Kmer Content 模块的前 10 几 bp 的波动，也无法消除。

- Per sequence GC content (参见 <https://www.biostars.org/p/103348/>)
- Sequence Duplication Levels (参见 <https://sequencing.qcfail.com/articles/libraries-can-contain-technical-duplication/>)。RNAseq 的 duplication 问题很难处理。有的研究不去 duplication, 直接后续分析, 有些研究去 duplication。去 duplication 又分两种。**FastUniq** 是把 PE 两端完全相同的 reads 去掉, **SOAPfilter** 看 reads 片段 (前 50 or 75bp?) 是否相同。SOAPfilter 主要是考虑到 Illumina 的测序特点, 越往后测序错误会越多, 原本是 duplication 的 reads 由于测序错误而被看成不是 duplication。我认为先测试一下, 用三种数据 (SOAPfilter 去 duplication 后数据, FastUniq 去 duplication 后数据, 没有去 duplication 数据) 进行 trinity 拼接, 看哪个效果好。
- Overrepresented sequences 对我们来说比较有价值。因为从 SRA 上下载数据无法知道其 adapter (及 index) 信息, overrepresented 的序列可能是接头序列。如果在该模块中出现且不是 adapter 序列, 应对其进行 blast 确认其来源是污染还是样本本身的序列 (特别是 RNAseq 某些高表达的序列)。

Fasta 与 univec 比对, 找出接头序列

- ✚ 安装 blast+ (sudo apt-get install blast+)
- ✚ 下载 univec 数据 (<http://www.ncbi.nlm.nih.gov/tools/vecscreen/univec/>)
- ✚ 将 univec 数据库格式化: makeblastdb -in UniVec -dbtype nucl -out UniVec_formatted -parse_seqids
- ✚ makeblastdb (我的理解是对其建索引) 参数解释 (<http://blog.csdn.net/likelet/article/details/7567426>):
 - -in 输入
 - -dbtype 要格式化的数据库是核酸序列还是蛋白序列
 - -out 输出名称

- **-parse_seqid** 推荐加上，现在有啥原因还没搞清楚

✚ 将*_1.fa 、*_2.fa 分别和 univec 进行比对: `blastn -query *_1.fa -out *_1.blast -db UniVec -outfmt "6 qseqid qlen sallseqid slen qstart qend sstrand evalue length pident" -evalue 1e-10 -num_threads 2 -max_target_seqs 1`

✚ blastn 参数解释 (<http://www.ncbi.nlm.nih.gov/books/NBK279675/>):

- **-query** 自己的输入文件 (*_1.fa 或*_2.fa)
- **-out** 输出
- **-db** 作为比对的数据库 (univec)
- **-outfmt** 自己定义输出格式，数字 6 表示输出格式以 tab 键分隔；qseqid 是 query (也就是我们的 read) 的 ID，qlen 是 query 的序列长度，sallseqid 是 (数据库中序列，在这里是 univec) 与 query 比对上的 ID (由于后面参数 -max_target_seqs 1, 这里 sallseqid 最后也只会出现 1 个 seqid)，slen 是比对上的 seq 的长度，qstart 是 alignment 部分在 query 序列上的起始位置，qend 是 alignment 部分在 query 序列上的终止位置 (通过这两个位置可以知道比对在测序序列的 3' 端还是 5' 端)，sstrand 是比对上 seq 的正链还是反链，evalue 是该 alignment 的 Evalue 值，length 是指 alignment 部分的长度，pident 是 univec 中 seq 和测序序列 read 之间 (alignment 部分) 相似度。
- **-evalue** (将数据库大小、query 长度等考虑在内) 偶然出现这样比对的可能性；Evalue 值越小，表示可能性越小。
- **-num_threads** 线程数目，在硬件容许的情况下，数值越大速度越快
- **-max_target_seqs** 设置显示比对上的序列数目

✚ 得到 blast 结果以后进行排序和统计，得到最可能 adapter 序列。

- 使用 awk、sort、uniq 命令进行组合: `awk '{print $3, $4, $5, $6, $7, $8}' *_1.blast | grep "NGB" | sort -k 1 | uniq -c | sort -n -r | head -50 | less -S`
- \$3 是 univec 中的序列 seqID
- \$4 是 seq 的长度 (如果比对上的 seq 本身很长，而 alignment 部分只有很短，那这 seq 是 adapter 序列的可能性比较

小)

- \$5 是 alignment 在 read 的起始位置, \$6 是 alignment 在 read 的终止位置, 可以判断接头污染在 read 的 3' 端还是 5' 端 (根据 hiseq2000PE 的测序原理接头污染应该是在 3' 端)
- \$7 是 Evalue, 命令行中已经设置了 evalue 为 1e-10, evalue 大于这个 1e-10 的 alignment 结果不会输出
- \$8 是 alignment 的长度, alignment 长度太短没有意义
- grep "NGB" 是因为在 seqID 中带有"NGB"字眼的 seq 是商业测序用的接头
- sort -k 1 | uniq -c | sort -n -r | head -50 | less -S 排序统计最后显示前 50 个比对上 seq (排名越靠前越有可能是该 SRA 数据使用的 adapter)

Trimmomatic 去接头、去低质量数据

✚ Trimmomatic manual 参见

http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf

✚ PE 数据:

- java -jar /home/emma/tools_download/Trimmomatic/trimmomatic-0.36.jar PE -threads 2 *_1.fq *_2.fq forward_paired.fq forward_unpaired.fq reverse_paired.fq reverse_unpaired.fq ILLUMINACLIP:/home/emma/tools_download/Trimmomatic/adapters/HalioTisTuberculata.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:20 MINLEN:40

✚ SE 数据:

- java -jar /home/emma/tools_download/Trimmomatic/trimmomatic-0.36.jar SE raw.fq clean.fq ILLUMINACLIP:TruSeq3-SE:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:40

✚ 输入：PE 情况下是两个输入文件，SE 情况下是一个输入文件。

✚ 输出：PE 情况下的输出是 4 个文件（两个 **pair reads** 文件，两个 **single read** 文件）；SE 情况下一个输出文件。

✚ Trimmomatic 需要自己准备接头文件（如

`/home/emma/tools_download/Trimmomatic/adapters/HaliotisTuberculata.fa`）。除了正常比对的方法根据接头序列

以外，Trimmomatic 还针对 PE 测序的特点进行接头序列判断。（1）PE 测序可能出现空载，即两个接头之间没有 **insert**；

（2）**insert** 长度太短；这两种情况都导致测序“测通”了，这时同一个 **insert** 片段的两个 **read** 会出现互补片段。

Trimmomatic 利用这个特点判断接头。

`/home/emma/tools_download/Trimmomatic/adapters/HaliotisTuberculata.fa` 文件内容见附录。

✚ 参数解释：

- PE/SE：表示调用 PE 模式还是 SE 模式
- ILLUMINACLIP: This step is used to find and remove Illumina adapters (Trimmomatic 称自己是针对 illumina 的数据特点)。ILLUMINACLIP:<fastaWithAdaptersEtc>:<seed mismatches>:<palindrome clip threshold>:<simple clip threshold>，其中<fastaWithAdaptersEtc>是前面所说的 adapter 文件，<seed mismatches>是匹配时允许的错配碱基数目，<palindrome clip threshold>和<simple clip threshold>是得分阈值，超过该阈值就判断为 adapters。因为 Trimmomatic 有两种判断 adapter 的方式，所以有两个阈值。上面给出的数值是经验值。
- LEADING, TRAILING, SLIDINGWINDOW 是为了去除低质量数据，其中 LEADING 是 5' 端质量值低于 3 的碱基，TRAILING 是 3' 端质量值低于 3 的碱基。Illumina 官方认为碱基质量值小于 3 的碱基就是很不可靠的碱基。SLIDINGWINDOW 是通过窗口的模式来判断质量是否可靠，（SLIDINGWINDOW 是不是可以用来减轻 Per tile sequence quality 的 quality 问题），SLIDINGWINDOW:<windowSize>:<requiredQuality>，命令行中 4 和 20 是经验值。

- MINLEN 设置输出结果最短的 read 长度，小于此阈值的 read 不输出。

经过去除接头、去接头得到的 reads 进行第二次 fastqc

✚ 命令参见第一次 fastqc。

Emma_RNAseq获取CleanData

Trimmomatic 接头文件:

>PrefixPE/1

AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT

>PrefixPE/2

GATCGGAAGAGCACACGTCTGAACTCCAGTCACAGTCAACAATCTCGTATGCCGTCTTCTGCTTG

>PCR_Primer1

AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT

>PCR_Primer1_rc

AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT

>PCR_Primer2

GATCGGAAGAGCACACGTCTGAACTCCAGTCACAGTCAACAATCTCGTATGCCGTCTTCTGCTTG

>PCR_Primer2_rc

CAAGCAGAAGACGGCATACGAGATTGTTGACTGTGACTGGAGTTCAGACGTGTGCTCTTCCGATC

>FlowCell1

TTTTTTTTTTAATGATACGGCGACCACCGAGATCTACAC

>FlowCell2

TTTTTTTTTTCAAGCAGAAGACGGCATACGA